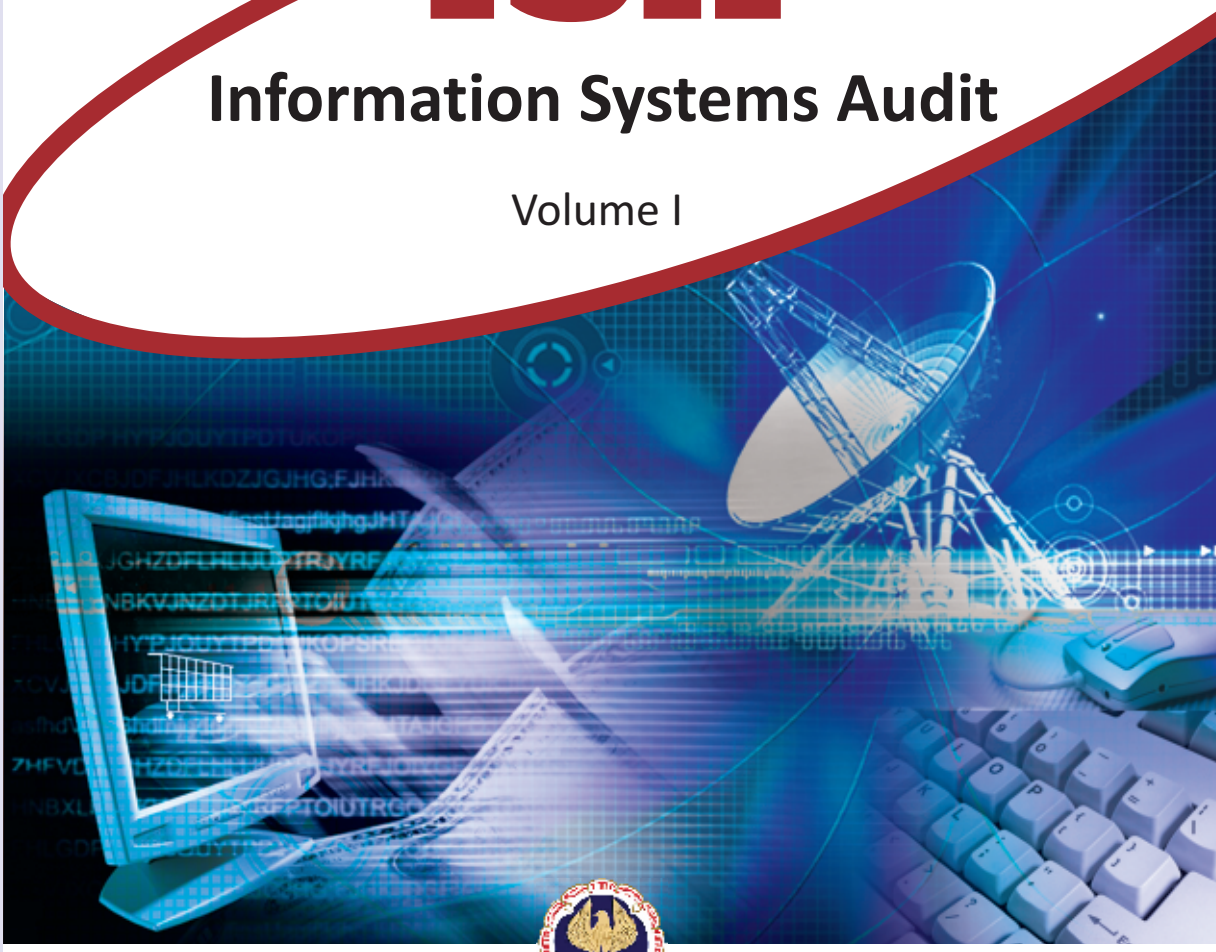
Background Material

ISA

Information Systems Audit

Volume I



Committee on Information Technology
The Institute of Chartered Accountants of India
(Set up by an Act of Parliament)
New Delhi

Background Material

On

Information Systems Audit

Volume I

The Institute of Chartered Accountants of India
(Set up by an Act of Parliament)
New Delhi

Revised Edition: February, 2010

Committee/ Department : Committee on Information Technology

E-mail : cit@icai.org

Website : www.icai.org, http://www.cit.icai.org

Price : Rs. 500/- (For Vol-I and Vol-II, Including CD)

ISBN : 978-81-8841-335-9

Published by : The Publication Department on behalf of the Institute of Chartered Accountants of India, ICAI Bhawan, Post Box No. 7100, Indraprastha Marg, New Delhi - 110 002.

Printed by : Sahitya Bhawan Publications, Hospital Road, Agra 282 003. February / 2010 / 1,000 Copies (Revised)

FOREWORD

Information Technology is revolutionizing the way businesses operate and offer goods and services. Business Process Outsourcing (BPO) is fast becoming the order of the day and is now migrating into Knowledge Process off shoring; Internet has expanded our horizons with the free flow of vast amount of information. Networks are increasingly connecting offices and diverse businesses. The world is truly transforming into a Global Village. All these developments are Information Technology driven.

The increasing use of Information Technology is not without the attached risks and threats. Hacking is the order of the day. Viruses/ Worms are commonplace. Denial of service attacks have happened. The ever increasing globalization is shrinking barriers amongst nations across the world. Developments in outsourcing and off shoring are based on sophisticated and complex Information System Infrastructures. All these have resulted in a growing need for assurance services on Information Systems in India.

The Committee on Information Technology (CIT) of the Institute has been established to identify the emerging professional opportunities in the Information Technology sector for members and prepare them to face the threats and challenges ahead. Since its inception, the Committee has proactively considered the modern day requirements and initiated steps to suitably equip the members in terms of knowledge and skills to face the challenges ahead.

Post Qualification Course on Information Systems Audit is the first initiatives of the Committee to enable members to offer value added services of IS Audit, which are in increasing demand.

It gives me immense pleasure to so see this revised ISA Background Material for the Post Qualification Course on IS Audit, to enable members to develop understanding of the intricacies of Information Systems Audit in a simple and lucid manner.

I appreciate the efforts put in by CA. K. Raghu, Chairman, IT Committee and other members of the Committee and also the faculty members for bringing out the revised background material.

I am sure that this course will equip you to practice in this emerging field and enhance the range of services that can be provided by you. I wish you all the very best in pursuing the ISA Course.

January 11th, 2010
New Delhi

CA. Uttam Prakash Agarwal
President

PREFACE

Today there is seamless integration of business processes, internal controls, accounting systems and Information Technology. Our members need to provide increased assurance and other value added services to clients in this scenario.

The Committee on Information Technology (CIT) of ICAI established in the year 2000 to identify the emerging professional opportunities in the Information Technology sector for members and prepare them to face the challenges ahead had started the course on Information Systems Audit (ISA) to suitably equip members to provide assurance related services in the field of Systems & Process Assurance and a course on Computer Accounting & Auditing Techniques (CAAT) to provide hands-on training on use of computers, CAAT Resources CD to provide exposure to use of General Audit Software/ Computer Assisted Audit Techniques. Additionally, the Committee has published the Technical Guide on Information Systems Audit to provide a guide to conduct of IS Audit and also organizes practical workshops/ Conferences/ Seminars to provide technical updates to members, apart from other developments.

The Committee on Information Technology is pleased to present this thoroughly revised, upgraded and enhanced Background Materials for the ISA Course keeping in view the required revisions to the course and modules in tune with developments in the field.

I am very thankful to CA. Uttam Prakash Agarwal, President and CA. Amarjit Chopra, Vice President, for the guidance and support in coming out with this revised material. I would like to record my deep appreciation for the guidance and support of the Members of the Committee on Information Technology in coming out with the revised materials. I am also very thankful to all the team members involved in conceptualizing the revision, content/ material development, content review and content editing/consolidation for this commendable job. I also acknowledge the significant contribution made by the Ms. Indu Arora, Additional Director of Studies.

I am confident that the revised Background Materials for the ISA Course will be of significant assistance to the members providing Information Assurance Services that is currently in increasing demand.

January 11th, 2010
New Delhi

CA K. Raghu
Chairman
Committee on Information Technology

Module 1: Information Technology Infrastructure and Communication/ Networking Technologies

Chapter 1: Introduction to Computer Hardware and Software

Types of computers - Hardware architecture of the computer - Various Input/Output (I/O) devices - ASCII and EBCDIC codes - Hardware monitoring procedures - Data and capacity management - Hardware acquisition plan - Definition of systems and application software - Various systems software and its brief description -Operating systems and its functions

Introduction to Database Management Systems - Introduction - Database and Database Management Systems (DBMS) - DBMS architecture - DBMS models - Database Languages - SQL - Roles and duties of a Database Administrator (DBA) and Data Administrator (DA)

Chapter 2 : Introduction to Computer Networks

Basics of communication - Simplex, Half-Duplex, and Full-Duplex Communications, Asynchronous & Synchronous Communication, Multiplexing, Switching techniques

Modem, Network Categories- LAN, WAN & MAN, Network Topology, Media used in communication, Factors that influence the use of media, Factors that degrade a signal.

Chapter 3: Introduction to OSI model

Various layers of OSI model - Application layer, Presentation layer, Session, Transport, Network layer, Datalink layer, Physical layer. Networking devices- Introduction to network management -IEEE LAN standards

Chapter 4: TCP/IP and Internet

A brief history of Internet & TCP/IP - Internet Administration - Generic Top-Level Domains (gTLDs)- TCP/IP Protocol Architecture -The architecture of TCP/IP suite -IP Addressing Scheme - The Domain Name System – Ports -Comparison between OSI model and TCP/IP protocol suite - Internet Services -Client/Server (C/S) Software Architectures--An Overview - Intrusion Detection Systems (IDS)

Chapter 5: Introduction to Firewalls

Characteristics of a Firewall -Types of Firewalls - Common implementation structures of a firewall - Limitations of Firewalls - Costs involved with Firewalls - General Controls associated with Firewalls - Phases in firewall lifecycle

Chapter 6: Cryptography

What is Cryptography? - Brief History of Cryptography - Why Cryptography? - The goals of cryptographic systems - Symmetric Key and Asymmetric Key Algorithms - How public key encryption method works - RSA : An Example for Public-Key Encryption - Digital Signatures - Comparison between Symmetric and Asymmetric Key Encryption Algorithms - Digital Envelopes - Digital Certificates - Cryptanalysis and their ways

Module 2: Protection of Information Assets

Chapter 1: Securing Physical Access

Introduction, IS Assets: Objects of Physical Access Controls, Physical Access, Threats and Exposures, Sources of Physical Access Threats, Physical Access Control Techniques, Administrative Controls, Technical Controls, Auditing Physical Access, Environmental Access Controls, Introduction, IS Assets: Objects of Environmental Controls, Environmental Threats and Exposures, Techniques of Environmental Control, Administrative Controls, Technical Controls, Integration and Fine Tuning of Environmental Controls, Audit and Evaluation of Environmental Controls, Audit of technical controls, Documentation of findings

Chapter 2: Logical Access Controls

Introduction, Objectives of Logical Access Controls, Paths of Logical Access, Logical Access Exposures, Technical Exposures, Malicious Code, Logical Access Controls Identification and Authentication, Authentication Techniques, Biometric Security, Access Controls in Operating Systems, Database Controls, Database Roles and Permissions, Views , Stored Procedures, Triggers, Database Restrictions, Audit Trail, Audit of Access Controls, Audit Procedures - Special Considerations, Identification of logical access paths, Audit Test Procedures, Systems Configuration, Logical Access mechanisms, User account management and password management, Privileged logons and special user accounts, Access to file directories and application logic and system instruction sets, Bypass Security Procedures, Appendix: Access Controls Checklist

Chapter 3: Network Security Controls

Introduction, Network Characteristics, Threats and Vulnerabilities, Information Gathering, Communication Subsystem Vulnerabilities, Protocol Flaws, Impersonation, Message Confidentiality Threats, Message Integrity Threats, Web Site Defacement, Denial of Service, Distributed Denial of Service, Threats from Cookies, Scripts and Active or Mobile Code, Network Security Controls, Architecture, Cryptography/ Encryption, Content Integrity, Strong Authentication, Remote Access Security, Firewalls, Intrusion Detection Systems, Auditing Network Security, Penetration Testing, Penetration Testing Scope, Penetration Testing Strategies, Types of Penetration Testing, Risks associated with Penetration Testing, Network Infrastructure Auditing Checklist, Network Server, Router, Firewalls, Network Administration and Security Auditing Checklist, Process, Authentication, Public Key Infrastructure (PKI), Access Control, Cryptography, Network Information Security, Information Security Administration, Microcomputer/PC Security, Audit Trails

Chapter 4: Application Controls

Introduction, Components of Application Controls, Application Boundary Controls, Input Controls, Source Document Design, Data entry screen design, Data code controls, Batch Controls, Data Input Validation Controls, Input Authentication Controls, Edit Controls , Data Input Error Handling and Reporting, Instruction Input Controls, Instruction input methods, Reporting Instruction Input Errors , Processing Controls, Data processing controls , Data file Controls, Output Controls, Existence Controls in Application Systems, Audit of Application Controls, Review of application controls

Chapter 5: Information Assets & Their Protection

Introduction, Information Classification, Classification of Information Assets, Data Privacy and Data Protection, Classification of Users, Naming Conventions, Access Control Models, Information Security Policy, Tools to Implement Policy: Standards, Guidelines, and Procedures, Components of a security policy, Program Policy, Components of Program Policy, Issue-Specific Policy, Components of Issue-Specific Policy, Areas Appropriate for Issue-specific Policies, Examples of Issue-Specific Policies, Network Policies, Data Privacy Policies, Data Integrity Policies, System Administration Policies, Usage Policies, Physical Security Policies, System-Specific Policy, Policy Implementation, Policy Documentation, Policy Visibility , System-Specify Policy Implementation, Interdependencies, Awareness, Training and Education, Cost Considerations, Audit of IS Security Policy

Module 3: Systems development life cycle & Application Systems

Chapter 1: Business Application Development Framework

Business Application Development Framework, Characteristics of System, Business, Application Development involves, Project Initiation, Need for Structured Systems Development Methodology, Risks associated with SDLC, Advantages for IS Audit of Structured Methodology, Overview of Phases in Structured Methodology of SDLC, Phase-Feasibility Study, Identification of problem, Identification of objective, Delineation of scope, Feasibility Study, Phase – Requirements Analysis, Understanding Requirements, Study of history, structure and culture, Study of Information flows, Eliciting user requirements, Structured Analysis, Context and Data Flow Diagrams (DFD), Entity-Relationship diagram, Data dictionaries, Decision Table / Decision Tree /Structured English, Decision Tree, Structured English (Psuedocode), State Transition diagram, System charts / program flow charts, Interface in form of data entry screens and dialogue boxes, Report layouts, Software Acquisition, Roles involved in SDLC, Steering committee, Project manager, Systems analyst, Module leader/Team leader, Programmers, Database Administrator (DBA), Quality assurance, Testers, Domain specialist, Technology specialist, Documentation specialist, IS auditor

Chapter 2: Phases in Software Development

Learning Goals, System Design Phase, Systems Design, Architectural design, Design of data / Information flow, Design of database, Design of user interface, Physical Design, Development Phase: Programming Methods, Techniques And Languages, Programming Methods & Techniques, Programming Language, Windows Platform, Unix / Linux based Platform, Coding style, Software Testing Phase, Objectives of testing, Levels of testing, Types of unit tests, Static analysis tests, Dynamic analysis tests, Integration / Interface testing: Final Acceptance Testing, Implementation of Software, Direct implementation / Abrupt change-over, Parallel implementation, Phased implementation, Pilot implementation, Activities during Implementation Stage, Post Implementation Review, Corrective maintenance, Adaptive maintenance, Perfective maintenance, Preventive maintenance, Umbrella Activities

Chapter 3: Alternative Methodologies of Software Development

Waterfall Model, Spiral Model, Data Oriented Systems Development, Process Oriented Approach, Object Oriented Systems Development, Prototyping, Rapid Application Development - RAD, Reengineering, Software reengineering consists of six activities, Inventory analysis, Document restructuring, Reverse engineering, Structured Analysis, Web-based Application Development, Informational, Download, Customization, Interaction, User Input, Transaction oriented, Service Oriented, Portal, Database Access, Data Warehousing, Risks associated with Web Based Applications, Agile Development, Information Systems Maintenance Practices, Change control, Continuous update of systems documentation, Program migration process, Testing program changes, Library control software, Executable and source code integrity, Program code comparison, Source code comparison, Object code comparison, Emergency changes, Configuration Management.

Chapter 4: Project Management Tools and Techniques

Budgets an Schedules, Software size estimation, Gantt Charts, Schedule, Gantt Chart for above schedule, Program Evaluation Review Technique (PERT), PERT terminology, Activity, Event, Predecessor activity, Successor activity, Slack, Maximum Total duration of this project = days, Dummy, Time estimate, Critical Path Method (CPM), System Development Tools and Productivity Aids, Code generators, Computer Aided Software Engineering (CASE), Classification of CASE tools, Upper CASE, Middle CASE, Lower CASE, Integrated CASE environments, CASE database (Repository), Advantages and limitations in using CASE, Benefits of using CASE, Disadvantages of CASE

Chapter 5: Specialised Systems

Artificial Intelligence (AI), AI applications, Cognitive Science, Expert Systems, Learning Systems, Fuzzy logic, Neural networks, Intelligent agents, Robotics, Virtual reality, Auditor's Role, Expert Systems, Components of expert systems, User interface, Interface engine, Knowledge base, Advantages of expert systems, Limitations of expert systems, Applications of expert systems, Applications of expert systems in IS Audit, Risk Analysis, Evaluation of Internal Control, Audit Program planning, Technical Advice, Data Warehouse, Features of Data Warehouse, Preparation of Data Warehouse, Consolidation, Drill-down, Slicing and dicing, Auditor's Role, Data Mining, Decision Support Systems (DSS), DSS frameworks, Design and Development, Implementation and use, Assessment and evaluation, DSS trends, Point of Sale Systems (POS), Automatic Teller Machines (ATM), Auditor's

Role, EDI, E-Commerce, ERP Systems, Electronic Data Interchange (EDI Systems), How does the EDI system function, Communication Software, Translation Software, EDI standard, Communication handler, EDI Interface, EDI Translator, Applications Interface, Application System, EDI standards, Features of ANSI ASCX, Features of UN/ EDIFACT, UN/XML, Web Based EDI, EDI Risks and Controls, Auditor's Role in Auditing EDI, Electronic Commerce (E-Commerce), The Advantages of the E Commerce, Types of E Commerce Models, Enterprise Resource Planning Systems (ERP Systems), Auditor's Role

Chapter 6: Auditing the System Development Process

IS Auditor's Role in Systems Development, Acquisition and Maintenance, IS Auditor's Role in Reviewing Developmental Phases of SDLC, Feasibility study, Requirement definition, Software acquisition process, Detailed design and programming phases, Testing phase, Implementation phase, Post-implementation review, System change procedures and program migration process, IS Auditor's Role in Project Management, Systems Development Project - Audit Checklist, Corporate Policies and Practices, User Requirements, Feasibility Analysis, Systems Design, Systems Specifications, Systems Development, Implementation, Post-Implementation

Module 4: Business Continuity Planning

Chapter 1: Business Continuity & Disaster Recovery Plan

Disasters and other disruptive events

Chapter 2: Documenting a Business Continuity Plan

Pre requisites in developing a Business Continuity Plan, Steps in developing a Business Continuity Plan (Phase I – Project Management and Initiation, Phase II – Business Impact Analysis / Risk Assessment, Phase III – Recovery strategies, Data communications, Voice communications, Fault tolerant, implementation strategies, Phase IV - Plan design and development, Phase V –Testing, maintenance, awareness and training)

Chapter 3: The Business Continuity Plan Audit

Priorities, Strategies, Responsibilities and Tasks, Plan Maintenance, Review of insurance coverage

Module 5 : Information Systems Organisation & Management

Chapter 1 – Governance

Enterprise Governance Definition - The enterprise governance framework - Best Practices in Enterprise Governance - Strategic Oversight -Enterprise risk management -The acquisition process - Board performance - Corporate Governance Definitions - Information Technology Governance - The Changing Role of the IT Department - Definition of IT Governance - Purpose of IT Governance - Some benefits of good IT governance - Who needs IT governance? – Best Practices in IT Governance - IT / IS Assurance Systems - IT Strategy Committee - The Balanced Score Card - Information Security Governance - Enterprise Architecture - Risk Management - E-Governance Definition- Users – Models – Benefits – Questions – Answers - Glossary of Terms

Chapter 2 - The Information System Management Process

The objectives of an organisation - The importance of management - The importance of managing the information systems department (ISD) - The process of The Deming Cycle - The Planning Function - The IS Steering Committee - The Master Plan of the Organisation - Long Range Plans - Short Range Plans – Policies – Standards – Guidelines – Procedures - The importance of leadership - The Acquisition of resources and Implementation of processes - Sequencing of policies, systems, processes, procedures and work instructions - The acquisition of IS resources - The Implementation of processes - Benchmarking processes - Financial Management processes -

IS Budgets and Variances - User Pays Scheme and Transfer Prices - User satisfaction survey processes - Capacity Management & Growth Planning processes - Goal Accomplishment processes / Indicators - Performance Measurement processes / Indicators - Quality Management processes Definition - ISO 9000:2000 Series - ISO 9126 Software Quality Model - The Software Capability Maturity Model (CMM) - Sourcing processes - HR processes - Documentation processes - Management Organisation Structures - Project and Line Management - The risks and controls of the various roles performed by personnel in the IS Department - Separation of Duties – Check – Act – Questions – Answers - Glossary of Terms

Chapter 3 – Auditing Information Systems Organisation & Management

Checklists / Audit Programmes - Suggestive Audit Checklist for auditing information systems organisation and management

Module 6: IS Audit Process

Chapter 1: IS Audit Process

Information Systems Audit Strategy, Fundamentals for Establishing an IS Audit Function, Audit Mission, Audit Charter, Structure and Reporting of the IS audit function, Staffing the IS Audit function, Internal and External Audit Control Framework, Quality Assessment and Peer Reviews, Engagement Letter, Skills and Competence Requirements of an IS Auditor, Phases in Information Systems Audit, Audit Planning, Preliminary Review , Knowledge of the Business, Understanding the Technology, Understanding Internal Control Systems, Legal Considerations and Audit Standards, Risk and Materiality, IS Audit Program, IS Audit Methodology, Examining and Evaluating Information, Communicating the Audit Results i.e. Reporting, Follow Up, Documentation Requirements, Use of Sampling in Information Systems Audits

Chapter 2: Information Risk Management

Information Risk Management: the Process (Step 1: Identification of Information Assets, Conceptual / Intangible Assets, Physical / Tangible Assets, Step 2: Valuation of Information Assets, Step 3: Identifying the potential threats, Step 4: Information Risk Assessment, Vulnerability Assessment, Probability or likelihood assessment, Impact analysis, Step 5: Developing Strategies for Information Risk Management),

Understanding the Relationships Between IS Risks and Controls, Acceptable / Residual Risk, Controls Assessment, IT Control Objectives, Category of Controls, Information Systems Control Framework, Information Systems, Risks & Controls – implications for Financial Auditor.

Chapter: 3 –IS Audit Techniques & Computer Assisted Audit Techniques

IT Environment Impact on audit methodology- Auditing in a computerized information system environment-Audit of IT controls and security-IS Audit approach-Computer Assisted Audit techniques-Type of CAATs-Other computer assisted audit techniques-Continuous auditing approach

Chapter 4: Overview of Information Systems Audit Regulations and Standards

Audit Standards, The Auditing and Assurance Standards issued by ICAI, Professional ethics and Code of Conduct prescribed by ICAI, IS Audit Guidelines by ISACA, COBIT – IT Governance Model, Other Global Standards on IS Assurance and Audit (A: The information security standards BS7799 & ISO 27001, B : SAS 70 - Statement

on Auditing Standards (SAS) No. 70, Service Organizations (AICPA), C: SysTrust, D: IT Infrastructure Library (ITIL), ISO 20000)

Overview of Regulatory Developments Impacting Controls in a Computerized Environment (A: Information Technology Act, 2000 of Government of India, B. The UNCITRAL Code, C: Sarbanes - Oxley Act 2002 Internal Control & COSO Criminal Penalties and Protection SOX and IT Controls Amendments to Clause 49 of the SEBI Listing Agreement, D: Basel II Framework for Risk Management).

Volume – I

MODULE 1: Information Technology Infrastructure and Communication/ Networking Technologies

MODULE 2: Protection of Information Assets

MODULE 3: System Development Life Cycle & Application Systems

Volume – II

MODULE 4: Business Continuity Planning

MODULE 5: Information Systems Organisation & Management

MODULE 6: IS Audit Process

Module – I

Information Technology Infrastructure and Communication/ Networking Technologies

1 Introduction to Computer Hardware and Software

Learning Objectives

To understand

- The concept of the Computer and its types.
- Hardware architecture of the computer.
- Various Input / Output (I/O) devices.
- ASCII and EBCDIC codes.
- Hardware monitoring procedures.
- Data and capacity management.
- Hardware acquisition plan.
- Definition of systems and application software.
- Various systems software and its brief description.
- Operating systems and their functions.
- Concept of DBMS and its three level architecture.
- Data independence, data models.
- Role of Database Manager.

Introduction

One of the key competence requirements for the Information Systems Auditor is the detailed understanding and knowledge of how computers process information and how their various components perform critical roles in input, processing, storage and output of information. This basic understanding is essential to appreciate the role each component plays in the computing architecture and the potential vulnerabilities thereof.

A computer is an electronic device that performs high speed computation of data. Collins English Dictionary describes it as 'A device, usually electronic, that processes data according to a set of instructions.' The actions carried out by the computer are either arithmetic or logical in nature. Its principal characteristics are:

- It responds to a specific set of instructions in a well-defined manner.
- It executes a prerecorded list of instructions (a program).

- It quickly stores and retrieves large amounts of data. Computers perform complex and repetitive procedures quickly, precisely and reliably. Modern computers are electronic and digital. The actual machinery (wires, transistors, and circuits) is called hardware; the instructions and data are called software. All general-purpose computers have the following hardware components:
    - **Central processing unit (CPU):** It is the heart of the computer, a component that actually executes instructions organized in programs ("software") which tell it what to do.
    - **Memory (fast, expensive, short-term memory):** It enables the computer to store, at least temporarily, data, programs, and intermediate results.
    - **Mass storage device (slower, cheaper, long-term memory):** It allows the computer to permanently retain large amounts of data and programs between jobs. Common mass storage devices include disk drives and tape drives.
    - **Input device:** Usually a keyboard and mouse, the input device is the conduit through which data and instructions enter a computer.
    - **Output device:** A display screen, printer, or any other device that lets us see what the computer has accomplished.

In addition to these components, many others like bus, registers, accumulators etc. make it possible for the basic components to work together efficiently. For example, every computer requires a bus that transmits data from one part to another.

Classification of Computers

Computers are generally classified on the basis of various factors:

1. The operational principle of computers.
2. Purpose for which they are built (General or Special).
3. Their size and data processing power.

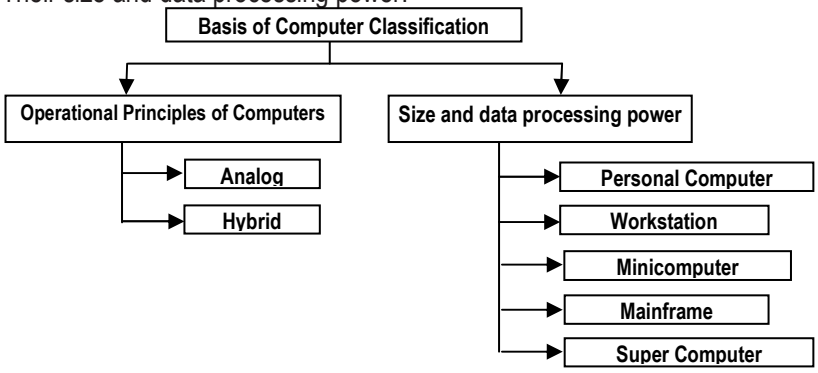


Fig. 1.1: Computer Classification

On the basis of operational principle of computers, they are categorized into **Analog** and **Hybrid** computers. Computers of yesteryears were analog in nature, but modern-day computers are founded on digital technology.

- **Analog Computers:** The analog computer is virtually extinct now. It is different from the digital computer in that it can perform numerous mathematical operations simultaneously. It is also unique in terms of operation, as it utilizes continuous variables for the purpose of mathematical computation. It utilizes mechanical, hydraulic, or electrical energy or operation.
- **Hybrid computers:** These computers, as the name suggests, are a combination of both Analog and Digital computers. Digital computers which work on the principle of binary digit system of "0" and "1" can give very precise results. But the problem is that they are too slow and incapable of large scale mathematical operation. In the hybrid types, the Digital counterparts convert the analog signals to perform Robotics and Process control.

On the basis of their size and data processing power, computers are classified as:

1. Personal computer:

A small, single-user computer based on a microprocessor. It can be defined as a small, relatively inexpensive computer designed for an individual user. Its price ranges anywhere from a few hundred pounds to over five thousand pounds. Because these are based on the microprocessor technology, manufacturers can put an entire CPU on one chip. Businesses use personal computers for word processing, accounting, desktop publishing, and running spreadsheet and database management applications. At home, personal computers are used mostly for playing games and surfing the Internet.

Nowadays, the world of personal computers is divided between Apple Macintoshes and PCs. The principal characteristics of personal computers are that they are single-user systems and are based on microprocessors. However, although these are designed as single-user systems, it is common to link them together to form a network.

Types of Personal Computers

Personal computers are generally classified by their size and chassis / case. The chassis or case is the metal frame that provides structural support for electronic components. Every computer system requires at least one chassis to house its circuit boards and wiring. The chassis also contains slots for expansion boards. If we want to insert more boards than there are slots, we will need an expansion chassis, which provides additional slots. There are two basic (flavors?) of chassis designs–desktop

models and tower models, and many variations on these two basic types. Then there are portable computers, which are small enough for people to carry. Portable computers include notebook and subnotebook computers, hand-held computers, palmtops, and PDAs.

- **Tower model:** The term refers to a computer in which the power supply, motherboard, and mass storage devices are stacked on top of each other in a cabinet. This is in contrast to desktop models, in which these are housed in a more compact box. The main advantage of tower models is that these provide for some usable space, which makes installation of additional storage devices easier.
- **Desktop model:** It is a computer designed to fit comfortably on top of a desk, typically with the monitor sitting on top of the computer. Desktop model computers are broad and low, whereas tower model computers are narrow and tall. Because of their shape, desktop model computers are generally limited to three internal mass storage devices. Desktop models that are designed to be very small are sometimes called **Slimline Models.**
- **Notebook computer:** Notebook computers are extremely lightweight computers, typically weighing less than 6 pounds and are small enough to fit easily in a briefcase. Aside from size, the principal difference between a notebook and a personal computer is the display screen. Notebook computers use a variety of techniques, known as flat-panel technologies, to produce a lightweight and non-bulky display screen of different qualities. Their computing power is nearly equivalent to that of personal computers. They have the same CPUs, memory capacity, and disk drives. However, all this power in a small package is expensive. Notebook computers cost about twice as much as equivalent regular-sized computers. They also come with battery packs, which enables us to run them without plugging them in. However, the batteries need to be recharged every few hours.
- **Laptop computer:** It is a small, portable computer that can easily sit in a user's lap. Nowadays, laptop computers are usually called notebook computers.
- **Subnotebook computer:** It is a portable computer that is lighter and smaller than a full-sized notebook computer. It has a smaller keyboard and screen, but is otherwise equivalent to a notebook computer.
- **Palmtop:** It is a small computer that literally fits in a user's palm. Compared to full-size computers, palmtops are severely limited, but good enough as phone books and calendars. Palmtops that use a pen rather than a keyboard for input are often called hand-held computers or PDAs. Because of their small size, most

palmtop computers do not have disk drives. However, many contain PCMCIA **(Personal Computer Memory Card International Association)** slots in which disk drives, modems, memory, and other devices can be inserted. Palmtops are also called PDAs, hand-held computers and pocket computers.

- **PDA :** Short for **Personal Digital Assistant**, it is a handheld device that combines computing, telephone/fax, and networking features. It is a portable computer that is small enough to be held in one's hand. The most popular hand-held computers are specifically designed to provide **PIM (Personal Information Manager)** functions, such as a calendar and address book.

Although extremely convenient to carry, handheld computers have not replaced notebook computers because they have small keyboards and screens. A typical PDA can function as a cellular phone, fax sender, and personal organizer. Unlike portable computers, most PDAs are pen-based, using a stylus rather than a keyboard for input. Some manufacturers are trying to solve the keyboard problem by providing an electronic pen. However, these pen-based devices rely on handwriting recognition technologies, which are still in their infancy. This means that they also need to incorporate handwriting recognition features. Some PDAs do react to voice input by using voice recognition technologies. PDAs are also called **palmtops**, **hand-held computers** and **pocket computers**.

2. Workstation:

A workstation is like a single - user personal computer, but with a more powerful microprocessor and, in general, a higher-quality monitor. In networking, workstation refers to any computer connected to a local-area network. It could be a workstation or a personal computer.

Fig. 1.2: Workstation

This computer is used for engineering applications (CAD/CAM), desktop publishing, software development, and other types of applications that require a moderate amount of computing power and relatively high quality graphics capabilities. Workstations generally come with a large, high-resolution graphics screen, built-in network support, and a graphical user interface. Most workstations also have a mass storage device like a disk drive, but a special type of workstation, called a **Diskless Workstation**, comes without a disk drive. The most common operating systems for workstations are UNIX and Windows NT. Like personal computers, most workstations are single-user computers. However, workstations are typically linked

together to form a local-area network, although they can also be used as stand-alone systems.

3. Minicomputer:

It is a midsize, multi-user computer capable of supporting up to hundreds of users simultaneously. In the past decade, the distinction between large minicomputers and small mainframes has blurred, almost like the distinction between small minicomputers and workstations. But in general, a minicomputer is a multiprocessing system capable of supporting up to 200 users simultaneously.

Fig. 1.3 : Minicomputer

4. Mainframe:

A mainframe is a very large and expensive computer capable of supporting hundreds, or even thousands, of users simultaneously. The chief difference between a super computer and a mainframe is that a supercomputer channels all its power to execute a few programs as fast as possible, whereas a mainframe uses its power to execute many programs concurrently. In some ways, mainframes are more powerful than supercomputers as they support more simultaneous programs. But supercomputers can execute a single program faster than a mainframe. The distinction between small mainframes and minicomputers is vague, depending really on how the manufacturer wants to market the machines.

Fig. 1.4 : Mainframe

5. Supercomputer:

Supercomputer is an extremely fast computer that can perform hundreds of millions of instructions per second. Supercomputers are very expensive and used only for specialized applications that require immense mathematical calculations (number crunching). For example - weather forecasting requires a supercomputer. Other uses of supercomputers are in the fields of scientific simulations,

Fig. 1.5 : Supercomputer

(animated) graphics, fluid dynamic calculations, nuclear energy research, electronic design, and analysis of geological data (e.g., in petrochemical prospecting).

Table 1 provides a comparison chart of these machines on various parameters

Parameter	PC	Workstation	Minicomputer	Mainframe	Supercomputer
Number of processors	Uni-processor	Uni-processor	Uni-processor	Multi-processor	Multi-processor
Computing power	Normal	Moderate	Middle range	Very high	Extremely high
Quantity of RAM	Normal	High	High	Very high	Very high
Number of users	Single user	Single user	Multi-user	Multi-user	Multi-user
Cost	Affordable	Affordable	High	Very high	Very high
Operating System	Open Source (Linux)	Proprietary but source code available for a price	Proprietary but source code available for a price	Proprietary	Proprietary
Used for	General purpose applications	General purpose applications	General purpose applications	Applications requiring very large programs and data that must be processed quickly	Applications requiring very large programs and data that must be processed quickly
Speed	In Million Instructions Per Second (MIPS)	In Million Instructions Per Second (MIPS)	In Million Instructions Per Second (MIPS)	In Floating Point Operations per second (FLOPS)	In Floating Point Operations per second (FLOPS)
Examples	Acer, Lenovo	PDP- 8	DEC VAX machine	IBM S/390	Cray Y-MP/C90

Table 1.1: Comparison of various machines on the basis of various parameters.

Components of a Computer

Computers perform the following four main functions:

- **Input :** Receive data through input devices for processing.
- **Processing :** Perform arithmetic and logic operations on the data received through input devices.

- **Output :** Present the result of computation to users through output device.
- **Storage :** Store information (both input and output data) on secondary storage devices.

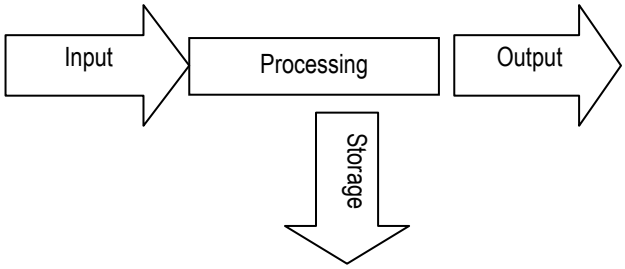


Fig. 1.6. Block Diagram of a Computer

Some components of a computer specialize in getting the inputs of data and instructions from the user, the primary memory that acts as the worktable for the computer to perform all temporary computations, the processor which actually performs the computations, the secondary storage which stores the information that is available for use in subsequent sessions and the various output devices that present the results of processing.

The basic hardware architecture of a personal computer or a desktop is:

a. Case/Chassis: It is a rectangular box which houses the motherboard, the power supply and the secondary storage device, the hard disk. These cases are of three main types: desktop, mini tower and full tower. Most of the modern PC cases are tower systems and stand upright and the smaller cases are often called mini towers. The size of the case is a deciding factor if additional disk drives or plug-in expansion cards, like the graphics cards, are to be fitted into a normal tower case. So there is need to check a case's capacity for future upgrading plans. The motherboard in a tower case is usually mounted vertically on one side, giving easy access to components.

Fig. 1.7: Case

b. Motherboard: It is the heart of the PC. It is a large circuit board in the middle of the case which houses the main components of a computer. It includes the BIOS (**Basic Input / Output System** that determines what the computer can do without accessing programs from a disk), the cache and the CPU.

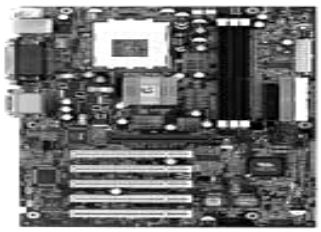


Fig. 1.9 : Motherboard

There are slots for plugging-in memory (RAM) and expansion slots for plugging in devices, such as graphics cards, TV Tuners, Ethernet LAN card, Universal serial Ports USB, and so on.

c. **CPU (Central Processing Unit):** The processor or CPU controls the computer and executes instructions and transports data around the computer system. It is a functional unit that interprets and carries out instructions. Based on the integrated circuit (silicon technology-Transistor) and their architecture configuration, they were called 386, 486, Pentium 2 and Pentium D. The latest chips are the Core 2 series which comprise the Duo, Quad and Extreme.

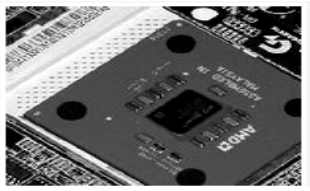


Fig. 1.10: Processor

d. **Graphics card:** Graphics cards are also known as video cards or video adapters and generate images onto the monitor. The display can be in a range of resolutions. The Common ones are 800 x 600, 1024 x 768, 1280 x 1024, 1680 x 1050 and higher, but there are other resolutions, too. The display can use a range of colors from 16 or 256 right up to 16-bit, 24-bit or 32-bit, which supports millions of colors. The greater the resolution and the more colors the card displays, the more memory it requires and the longer it takes to draw the image.

Fig. 1.11: Graphics Card

e. **Ports:** These are the plugs and sockets of the motherboard or the expansion cards on the motherboard that are visible on the chaises. They are the Serial and Parallel communication paths which were used to connect printers and scanners but have now been replaced by the Universal Serial Bus (USB). The USB can be used to connect any device to the computer including printers, scanners, a mouse, external hard drives, cameras, music interfaces and modems.

Fig: 1.12: USB Port

f. **Memory:** Here the computer does its sums, essentially an area where data can be stored, retrieved and manipulated. The types of memory are RAM and ROM.

Fig. 1.13: Memory

RAM or Random Access Memory: The RAM is volatile, which means that when we remove the power, their content is lost.

The two main types of memory are **SRAM (Static RAM)** and **DRAM (Dynamic RAM)**. SRAM retains its contents for as long as power is supplied to it whereas DRAM retains data for a few milliseconds even under power. The new types of RAM are the **DDR (Double Data Rate)** and **SDRAM (Synchronous DRAM)**.

DDR2 (Double Data Rate Two). Synchronous Dynamic Random Access Memory runs twice as fast as DDR and generally needs less power. The speed

and capacity of the RAM are important factors that determine the performance of a computer. For example, a personal computer running on Windows XP should have a minimum of 1GB RAM; Vista requires a minimum RAM memory of 2GB.

ROM or Read Only Memory: It is used to store set-up data which is executed at the initial startup or when the power is switched on. Data in ROM is permanent and non-volatile after the power is switched off.

Flash memory is both volatile and non-volatile; its contents can be changed and remembered even after switching off. This is popular in modems, cameras and, of course, USB flash drives. There are also Flash Drives known as **SSD (Solid State Drive/Disk)**, which use a type of memory rather than a mechanical disk drive. Accessing data from a Flash Drive is faster than from a hard disk.

Parts of the computer that constitute the storage function are:

i. **Hard Drive:** A collection of hard platters coated with magnetic material to which data can be written and read using a series of read/write heads. A drive has about eight platters which rotate at speeds of 5420 or 7200 rpm. The whole unit is sealed inside a case which prevents it from dust. There are read-write heads above the platters at a distance of 10 to 25 millionths of an inch. The storage capacity of a hard drive ranges from 10GB, 80GB and 120GB. Modern drives have a storage capacity of about 250 GB to 500 GB.

ii. **Optical Drive:** An Optical drive reads CDs and DVDs. The main difference between these drives is the speed at which they write to a disc. They are typically 16x, 32x and 64x, and vary depending on the media (CD or DVD).

Fig. 1.14: Optical Drive

Input / Output Devices

Input / **Output** or **I/O** refers to the communication between an information processing system (such as a computer), and the outside world, possibly a human, or another

information processing system. **Inputs** are the signals or data received by the system, and outputs are the signals or data sent from it. I/O devices are used by a person (or other system) to communicate with a computer.

Fig: 1.15: Power Supply

Power Supply Unit: It supplies power to all the parts inside the chaises and also the monitor. The power configuration rated at 250 W, 350 W or 600W allows further expansion and additions to the system.

Input Devices

An input device is any peripheral piece of computer hardware equipment that is used to provide data and control signals to an information processing system, such as a computer. An input device converts input data and instruction into a suitable binary form which can be accepted by the computer.

Commonly used input devices are Keyboard, Mouse, Trackball, Game Controllers, Scanners, Barcode Readers, Optical Character Readers, Digitizer, and Multi-media input devices.

1. Keyboard

A keyboard is an input device, partially modeled after the typewriter keyboard, which uses an arrangement of buttons or keys. A keyboard typically has characters engraved or printed on the keys and each press of a key typically corresponds to a single written symbol. While most keyboard keys produce letters, numbers or signs (characters), other keys or simultaneous key presses can produce actions or computer commands. When a key is pressed, it produces an electrical signal which is detected by an electric circuit called **Keyboard encoder,** which detects the key that has been pressed and sends a binary code to the computer. Normally, the keyboard is used to type text and numbers into a word processor, text editor or any other program. Nowadays most users use style keyboards, as shown in Fig. 1.16. Keyboards are also used for computer gaming, either with regular keyboards or keyboards with special gaming features, which can expedite frequently-used keystroke combinations.

Fig.1.16: Keyboard QWERTY

2. Mouse

A mouse is a pointing device that functions by detecting two-dimensional motion relative to its supporting surface.

Fig.1.17: Mouse

Physically, a mouse consists of an object held under one of the user's hands, with one or more buttons.

It sometimes features other elements, such as "wheels", which allow the user to perform various system-dependent operations. An extra button or feature can add more control or dimensional input. Each wheel of the mouse is connected to a shaft encoder which emits an electrical pulse for every incremental rotation of the wheel. When a user moves the mouse across a flat surface, the cursor on the CRT screen also moves in the direction of the mouse movement. By moving the mouse, the user can point to the menu on the screen, and the user can communicate his choice by clicking the mouse button on that position. The mouse's motion typically translates into the motion of a pointer on a display, which allows for the fine control of a **Graphical User Interface**.

3. Joystick

A joystick is an input device consisting of a stick that pivots on a base and reports its angle or direction to the device it controls. Joysticks are often used to control video games, and usually have one or more push-buttons, whose state can also be read by the computer. A popular variation of the joystick used on modern video game consoles is the analog stick. The joystick has been the principal input device in flight control in the cockpit of aircrafts, particularly the fast military jets, where centre stick or side-stick location may be employed. Joysticks are also used for controlling machines such as cranes, trucks, underwater unmanned vehicles and zero turning radius lawn mowers. Miniature finger-operated joysticks have been adopted as input devices for smaller electronic equipment, such as mobile phones.

Fig.1.18 : Joystick

4. Scanner

A scanner is a device that optically scans images, printed handwriting, or an object, and converts it into a digital image. Common examples are variations of the desktop (or flatbed) scanner where the document is placed on a glass window for scanning. Hand-held scanners, where the device is moved by hand, have evolved from text scanning "wands" to 3D scanners, which are used for industrial design, reverse engineering, test measurement, orthotics, gaming and other applications. Mechanically driven scan that move the document are typically used for large - format documents, where a flatbed design would be impractical.

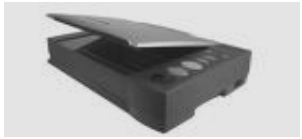


Fig. 1.19 : Scanner

5. Barcode Reader

A barcode reader is an electronic device for reading printed barcodes. It consists of a light source, a lens and a light sensor translating optical impulses into electrical ones. Nearly all barcode readers contain decoder circuitry, which analyzes the barcode's image data provided by the sensor and sends the barcode's content to the scanner's output port to the scanner's output port.

Fig. 1.20: Barcode Reader

6. Webcam

Webcams are video capture devices connected to computers or computer networks, often using a USB port or, if connected to networks, via Ethernet or Wi-Fi. Their most popular use is for video telephony, permitting a computer to act as a videophone or video conferencing station. Other popular uses, which include the recording of video files or even still-images, are accessible via numerous software programs, applications, and devices. They are known for their low manufacturing cost and flexibility.

Fig. 1.21 : Webcam

Output Devices

An output device is any piece of computer hardware equipment used to communicate the results of data processing carried out by an information processing system (such as a computer to the outside world). The output devices receive information from the computer and provide them to the user. The computer sends information to the output devices in binary coded forms. The output devices covert them into a form which can be used by users such as a printed form or a display on a screen.

Commonly used output devices are monitors (both CRT and LCD), printers, plotters, speakers and multi-media output devices like LCD Projectors.

1. Visual Display Terminal

Visual Display Terminals (VDTs), known as **Monitors**, are the essential link between computers and people that generates soft-copy output on the screen. VDTs translate complex computer languages into displayed text and pictures that humans can comprehend and use. Without monitors, computers would still be the province of high-tech laboratories and university researchers, performing mathematical computations with complex printouts as their only output. Most VDT units contain CRT for visual display. Other types of display units that are also available include **LED (Light Emitting Diodes)** display, LCD display and plasma screens.

- **CRT**: The VDT which contains CRT for visual display is also called CRT Terminal. It consists of a small memory known as **Buffer**. Each character entered through keyboard is stored in the buffer and displayed on the terminal screen.
- **LCD**: In LCDs, a liquid crystalline material is sandwiched between two glass or plastic plates. The front plate is transparent and conductive. A voltage is applied between a segment and back plate to create an electric field in the region under the segment. LCD simply changes the reflection of available light. As LCDs are lightweight, they are used mainly in portable computers.

2. Printers

A printer is a peripheral device which produces a hard copy (permanent human-readable text and/or graphics) of documents stored in electronic form, usually on paper or transparencies. Based on their technology, printers are classified as

- **Impact Printers**
- **Non-Impact Printers**

i. Impact Printers

An Impact printer is the oldest print technology of electromechanical mechanism that uses hammers or pins to strike against a ribbon or paper to print texts. The most common impact printers are:

- **Dot Matrix Printer**
- **Daisy Wheel**
- **Line Printers**

a. Dot Matrix Printers: These are impact printers that use a matrix or small pins to create precise dots. They operate by using electromagnetic and small wires to press an ink ribbon onto the page to form a series of dots to create characters. The more pins per inch, the higher the print resolution. Most dot-matrix printers have a maximum resolution around 240 dpi (dots per inch). Dot matrix printers vary in print resolution and overall quality with either 9 or 24-pin print-heads. As these move from left to right, these are slower, but are cheaper.

Fig.1.22 : Dot Matrix Printer

b. Daisy Wheel Printers: These printers have print-heads composed of metallic or plastic wheels cut into petals. Each petal has the form of a letter (in capital and lower-case), number, or

Fig. 1.23 : Daisy wheel Printer

punctuation mark on it. When the petal is struck against the printer ribbon, the resulting shape forces ink onto the paper. Daisywheel printers are loud and slow. They cannot print graphics, and cannot change fonts unless the print wheel is physically replaced. With the advent of laser printers, daisywheel printers are no longer used in modern computing environments.

c. Line Printers: The line printer is a form of high-speed impact printer in which one line of type is printed at a time. The mechanism uses a large spinning print drum or a looped print chain. As the drum chain are rotated over the paper's surface, electromechanical hammers behind the paper push the paper (along with a ribbon) onto the surface of the drum or chain, marking the paper with the shape of the character on the drum or chain.

Fig. 1.24: Line Printer

Because of the nature of the print mechanism, line printers are much faster than dot matrix or daisywheel printers; however, they tend to be quite loud, have limited multi-font capability, and have low print quality than more recent printing technologies.

ii. Non Impact Printers

Non impact printers becoming popular largely because of their improved print quality coupled with low cost. These printers can produce both text and graphics. Because nothing actually strikes the paper, these are much quieter than impact printers. Most non-impact printers produce dot-matrix patterns. The main types of non-impact printer are:

- **Thermal Printer**
- **Laser Printer**
- **Ink Jet Printer**

a. Thermal Printers: Thermal printers use a thermal ribbon which is soaked with a wax type ink which melts and is then transferred to the paper. Characters are formed by heated elements placed in contact with special heat sensitive paper, forming darkened dots when the elements reach a critical temperature. Thermal printers are widely used in battery powered equipment such as portable calculators. They have a low printing speed and poor quality of print.

Fig.1.25 : Thermal Printer

b. Laser Printers: Laser printers are page printers that rapidly produce high quality text and graphics on plain paper. An entire page is processed at a time. The printers use a laser beam to produce an image of the page containing text/graphics on a photosensitive drum which is coated with negatively charged photo-conductive material. Negatively charged ink powder caller **Toner** is used on a laser printer for printing. These printers produce low noise, work with high speed and the printing quality is high, but they are more expensive than ink jet or dot matrix printers and also large in size.

Fig. 1.26 : Laser Printer

c. Ink-Jet Printers: In Ink-Jet printers, characters are formed when an electrically charged or heated ink is sprayed in fine jets onto the paper. When voltage is applied to the element, it bends and creates a pressure wave to force out a drop of ink. Individual nozzles in the printing head produce high resolution (up to 400 dots per inch or 400 dpi) dot matrix characters. Ink jet printers have a low cast, are compact in size, and produce low noise.

Fig. 1.27: Ink Jet Printer

Their color printing quality is affordable but requires special paper. Usually the speed of ink jet printers is slower than that of laser printers.

3. Plotters

A plotter is a vector graphics printing device that prints graphical plots, and is connected to a computer. Vector graphics include geometrical primitives, such as points, lines, curves, and shapes or polygon(s), which are all based on mathematical equations, to represent images in computer graphics. There are two types of plotters:

- **Pen plotters**
- **Electrostatic plotters**

i. Pen Plotters: Pen plotters print by moving a pen across the surface of a piece of paper. But their printing is restricted to line art. Pen plotters can draw complex line art, including texts, but very slowly because of the mechanical movement of the pens. In fact, a pen plotter's speed depends on the type of pen used.

Fig. 1.28 : Pen Plotter

ii. **Electrostatic Plotters:** Electrostatic Plotters produce a raster image by charging the paper with high voltage. The voltage attracts the toner, which is then melted into the paper with heat. These type of plotters are fast, but the printing quality is poorer than that of pen plotters.

Fig. 1.29: Electrostatic Plotter

Character Encoding

To represent numeric, alphabetic, and special characters in a computer's internal storage and on magnetic media, some sort of coding system has to be used. In computers, the code is made up of fixed size groups of binary positions. Each binary position is assigned a specific value; for example, 8, 4, 2, or 1. In this way, every character is represented by a combination of bits that is different from other combinations.

Computer codes help one to represent characters for use by the machine. The two main computer codes that are widely used these days are:

i. **ASCII:** ASCII is the **American Standard Code for Information Interchange**, also known as ANSI X3.4. There are many variants of this standard, which allow different code pages for language encoding, but they all basically follow the same format. ASCII is quite elegant in the way it represents characters, and is very easy to write code to manipulate upper/lowercase and check for valid data ranges. ASCII is essentially a 7-bit code which allows the 8th most significant bit (MSB) to be used for error checking. However, most modern computer systems use ASCII values of 128 and above for extended character sets.
ii. **EBCDIC:** EBCDIC (**Extended Binary-Coded Decimal Interchange Code)** is a character encoding set used by IBM mainframes. It uses the full 8 bits available to it, so parity checking cannot be used on an 8 bit system. Also, EBCDIC has a wider range of control characters than ASCII.

Some Hardware Management Issues

1. Data and Capacity Management

One of the challenges to the right size of the capacity of the computing architecture arises from its need to handle varied business requirements. For example, the capacity requirements for a retail grocery store are very different from that of a stock broking firm or an architect's firm. Some of the key factors that influence the capacity of each of the key components, such as the input devices, RAM, CPU architecture & speed, HDD, output devices, networking capability and capacity are the volume of transactions, complexity of computations and mode of input and output delivery. For example, the choice of inappropriate and inadequate software can significantly reduce the operational efficiency of a business and its productivity, which can affect

the achievement of its objectives. In the present age, one of the main issues an organization has to face is the constant and never-ending growth of data and requirement for greater storage capacity along with the problem of data safety, security and integrity.

In modern day enterprises, which employ large scale database applications, multimedia applications, the requirements for disk storage run from gigabytes to terabytes. If a proper data and storage management mechanism is in place, problems of downtime, business loss on account of lost data and insufficient storage space can be avoided.

The key issues in data and capacity management include the following:

- How to effectively manage rapidly growing volume of data?
- How to leverage data and storage technology to support business needs?
- What is the best data and storage management framework for an enterprising business environment?
- How to optimize the performance of the data storage hardware and software to ensure high availability?
- What is the best way to achieve greater storage capacity?
- How effective is the current data backup and storage management system?

Like every other investment in an organization, IT investment too needs to be justified on the basis of Returns on Investment (ROI). The capacity of IT resources required, such as processor power, CPU clock speed, hard disk capacity and number of terminals have to be planned meticulously with business need in mind. Proper capacity planning has the following advantages:

i. It assures appropriate infrastructure capacity levels to support existing and future business functions.
ii. It reduces resource expenditures and IT operating costs.
iii. It improves application infrastructure availability.
iv. It enables prioritization and reallocation of existing applications, systems, and network components.
v. It projects future resource needs, protecting the infrastructure from slow-downs as capacity demands increase.
vi. It facilitates improved support of new and existing applications.

2. Selection Criteria for Hardware Acquisition

Organizations need to be clear on their Information System requirements and the hardware and software they need to procure. This calls for a set of selection criteria for both hardware and software acquisition, but only after one carries out a proper data and capacity planning.

Hardware acquisition is not merely limited to computers only but also to all equipment and peripherals associated with the entire computer system, such as printers, scanners, modems, routers, and CD-ROMs, to cite a few. Before providing specifications on what is needed to vendors in the form of an **Invitation to Tender (ITT)**, it is essential to prepare a checklist based on the following evaluation criteria.

3. Compatibility and Industry Standards

i. Is the hardware to be procured compatible with the existing one and does it take care of future applications?
ii. Have the workload and performance requirements been calculated and is the hardware suggested capable of fulfilling them?
iii. Are there any industry standards for the same, and do the hardware components comply with them?

Easy Operations

- Can the hardware be installed and maintained by locally available engineers?
- Can the hardware be serviced, maintained, and upgraded locally?
- Does the hardware need any special training for its operation or will the users be able to access/use it with minimal additional technological competence?

Support

- What type of technical support will be provided by the vendor?
- Are appropriate manuals for various operations available?
- If so, are they available in a variety of media?
- Can the manuals be understood by intended users?
- Does the vendor have a strong Research and Development Division with adequate staff?
- Will the vendor help in the smooth transition from the existing application to the new one?
- What is the quantum of training that the vendor will provide?
- What is the backup facility that the vendor will provide?

Cost

a. Is the cost quoted competitive and comprehensive?
b. Are all the requested components included in the purchase price?
c. Are there any hidden costs?
d. Will the vendor provide support at an affordable cost?

4. Hardware Maintenance

It is not uncommon for organizations to outsource the maintenance of computer hardware, which includes any or all of desktops, servers, networks, cabling, etc. One of the important criteria that need to be considered before finalizing the vendor is the issue of maintenance. The organization has to have a hardware maintenance program that takes into consideration the following:

i. Which company takes care of what IT resource? For example, computers may be serviced by one company and printers by another.
ii. How many times during a year does the vendor provide preventive maintenance and when?
iii. Was any problem reported in the past, and what corrective steps were suggested? This has to be documented.
iv. What is the cost of maintenance? Has, at any time during the year, the amount spent on maintenance exceeded the budgeted amount? If yes, the details have to be documented.
v. Apart from the preventive maintenance schedule, how many times during the year did the vendor come for servicing the equipment because of some snag failure?
vi. What is the MTBF (Mean-Time-Between-Failure) and MTTR (Mean-Time-To-Repair) value? Typically, MTBF value must be high and MTTR value must be low.

Hardware monitoring procedures help in the hardware maintenance program. The following are the typical reports that are generated:

i. **Hardware Error Reports:** These reports provide details of failures, if any, of hardware resources in an organization.
ii. **Availability Reports:** These reports provide details on the availability of the hardware, software and all associated systems. Being the life-line of the organization, the availability of IT division must be 24 x 365. If there is any break-down, this report provides details on the same.
iii. **Utilization Reports:** All IT resources must be optimally used. If there is excessive utilization of a resource, steps have to be taken to correct it. Details on utilization parameters are automatically provided by tools supplied with the operating system and hardware. By using these tools, one can check the utilization of CPU, memory, channel/bus, and hard disk.

These reports not only help in maintenance but also in effective planning for the future.

Software

Computer hardware performs various computations, based on the instruction input given by the users and other programs; such a set of instruction input is actually its software. The term software in its broad sense refers to the program that the machine executes.

More precisely, **Computer Software** (or simply software) is the collection of computer programs, procedures and documentation that perform different tasks on a computer system. Program software performs the function of the program it implements, either by directly providing instructions to the computer hardware or by serving as input to another piece of software. A system is a set of interacting or interdependent entities, real or abstract, forming an integrated whole.

Classification of Software

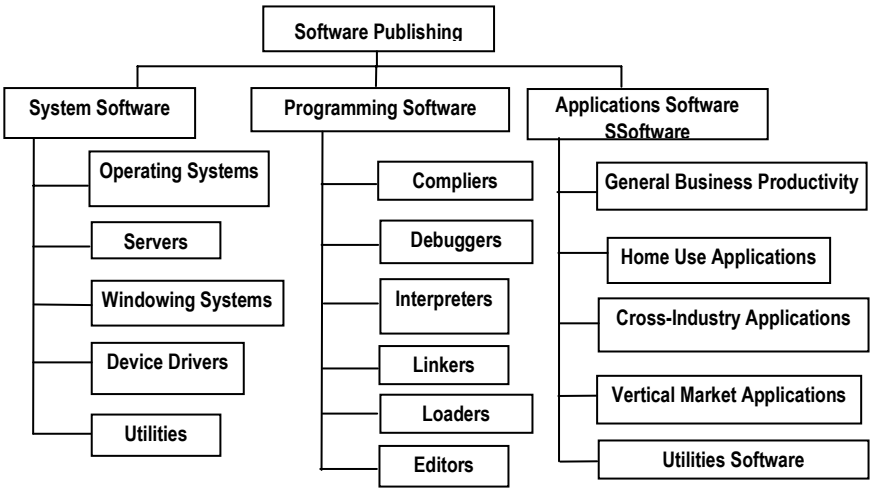


Fig. 1.30: Software Classification Chart

There are three broad kinds of software, as shown in Fig. 1.30.

1. System Software
2. Programming Software
3. Applications Software

1. System Software

It is the low-level software required to manage computer resources and support the production or execution of application programs but is not specific to any particular application. These refer to the set of programs that -

i. Enable a computer to work.
ii. Control the working of a computer.
iii. Provide facilities for programmers to develop applications.
iv. Enable various I/O devices to be connected ; and
v. Help in the efficient use of computer resources.

The various categories of systems software are:

a. Operating Systems: These include client and network operating systems which handle the interface to peripheral hardware, schedule tasks, allocate storage, and present a default interface to the user when no application program is running. (Includes all client and network operating systems.)

b. Servers: A server is any combination of hardware or software designed to provide services to clients. When used alone, the term typically refers to a computer which may run a server operating system, but is commonly used to refer to any software or dedicated hardware capable of providing services.

c. Windowing Systems: A **windowing system** (or **window system**) is the component of a graphical user interface (GUI), and more specifically of a desktop environment, which supports the implementation of window managers, and provides basic support for graphics hardware, pointing devices like mice, and keyboards. The mouse cursor is also generally drawn by the windowing system.

The term **windowing system** is sometimes used to refer to other elements of a graphical Interface, such as those belonging to window managers or even applications, but does not include windows themselves.

d. Device Drivers: These are a part of the system software which informs the computer how to use a particular peripheral device like a printer or CD-ROM. Normally, the device drivers are supplied along with the device and one has to install them in the machine.

e. Utilities: Utility software (also known as **Service program**, **Service routine**, or **Utility routine**) is a computer software designed to help manage and tune the computer hardware, operating system or application software by performing a single task or a small range of tasks. Some types of utility software are integrated into the major operating systems:

- Disk Storage Utilities including Disk defragmenters, Disk checkers, Disk cleaners, Disk partitioners, Backup, Disk compression, file managers and Archive.
- System Profilers
- Anti-virus
- Data Compression

- Cryptographic Utilities
- Network managers
- Registry cleaners
- Launcher Applications

2. Programming Software

Programming software usually provides tools to assist a programmer in writing computer programs and software by using different programming languages in a convenient way. The tools include:

a. Compiler: A compiler is a computer program (or a set of programs) that transforms the source code written in a computer language (the **source language**) into another computer language (the **target language**, often having a binary form known as the **object code**). The most common reason for transforming the source code is to create an executable program.

b. Debugger: A debugger is a utility that helps in identifying any problem occurring during execution. By using the debugger utility, one can pinpoint when and where a program terminates abnormally, indicate the value of the variables used, and in general provide information so that the programmer can locate bugs/

c. Interpreter: An interpreter is a computer program that executes instructions line by line, i.e performs instructions written in a programming language.

d. Linker: Linking is the process of combining various pieces of code and data together to form a single executable unit that can be loaded in memory. The process of linking is generally done during compilation of the program. After linking, the loading process takes place by using a loader software.

e. Loader: A loader is a program which loads the code and data of the executable object file into the main memory, goes to the very first instruction and then executes the program.

f. Editor: An editor program is a system software utility that allows the user to create and edit files. During the editing process, no special characters are added. On account of this, the editor utility is used to create the source code.

There are two kinds of editors: **Line editors** and **Screen editors**.

A Line editor edits files line by line whereas a Screen editor helps to load files and edit them by moving the cursor movement keys. Nowadays, users use only screen editors. Notepad utility in Windows, vi editor in UNIX, Edit utility in MS-DOS are typical examples of screen editors.

3. Applications Software

It refers to the set of software that performs a specific function directly for the end-user. There are varieties of application software for different needs of users.

a. General Business Productivity Applications

These refer to the software used for general business purposes, such as improving productivity. It includes office suite applications, such as word processors, spreadsheets, simple databases, graphics applications, project management software, computer bases training software, etc.

b. Home Use Applications

These refer to the software used in the home for entertainment reference or educational purposes that includes games, reference, home education, etc.

c. Cross-Industry Application Software

These refer to the software designed to perform and/or manage a specific business function or process that is not unique to a particular industry. It includes professional accounting software, human resource management, customer relations management, Geographic Information system software, Web page /site design software. etc.

d. Vertical Market application Software

These refer to the software that performs a wide range of business functions for a specific industry. These include manufacturing, retail healthcare, engineering, restaurants, etc.

e. Utilities Software

These refer to a small computer program that performs a specific task. Utilities differ from other applications software in terms of size, cost, and complexity. Examples include compression programs, antivirus, search engines, font, file viewers, voice recognition software, etc.

Some other common terminologies associated with software are:

Middleware

Middleware is computer software that connects software components or applications. It consists of a set of services that allows the running of multiple processes on one or more machines across a network. This technology evolved to provide for interoperability in support of the move to coherent distributed architectures, which are used most often to support and simplify complex, distributed applications. It includes web servers, application servers, and similar tools that support application development and delivery. Middleware sits "in the middle" between application software working on different operating systems. It is similar to the middle layer of a three-tier single system architecture, except that it is stretched across multiple systems or applications. Examples include database systems, telecommunications software, transaction monitors, and messaging-and-queuing software.

Proprietary Software

Proprietary Software (also called non-free software) is a computer software that is the legal property of one party. The terms of use for other parties are defined by contracts or licensing agreements. These terms may include various privileges to share, alter, dissemble, and use the software and its code. There are restrictions on using, copying and modifying this software imposed by its proprietor. Restrictions on use, modification and copying are achieved either by legal or technical means and sometimes by both. Technical means include releasing machine-readable binaries to users and withholding the human-readable source code. Legal means can involve software licensing, copyright, and patent law**.**

Shareware

The term **Shareware** refers to the proprietary software that is provided to users without payment on a trial basis and is often limited by any combination of functionality, availability or convenience. It is often offered as a download from an Internet website or as a compact disc included with a periodical such as a newspaper or magazine. The aim is to give buyers the opportunity to use the program and judge its usefulness before purchasing a license for the full version of the software.

Shareware is a method of marketing software based on the philosophy of “Try before We Buy”, and is usually offered as a trial version with only some features or as a full version, but only for the trial period. Once the trial period is over, the program may stop until a license is purchased. Shareware is often offered without support, updates, or help menus, which become available only with the purchase of a license. The words "free trial" or "trial version" are indicative of shareware.

Open Source

Open Source is an approach to the design, development, and distribution of software, offering practical accessibility to software's source code. In open source any software along with its source code is easily accessible, and one can customize it to specific requirements or add additional features to it, and make it available for distribution and further improvement. One of the best examples is Linux.

In practice, open source usually means that the application is free to users as well as developers. Furthermore, most open source software have communities that support each other and collaborate on development. Therefore, unlike freeware, there are future enhancements, and, unlike shareware, users are not dependent on a single organization.

Freeware

A software which permits users the following freedom is termed as a **Free Software** or **Freeware**.

i. The freedom to run the program, for any purpose
ii. The freedom to study how the program works, and adapt it to one's needs. Access to the source code is a precondition for this.
iii. The freedom to redistribute copies to others
iv. The freedom to improve the program, and release improvements to the public, so that the whole community benefits. Access to the source code is a precondition for this.

A program that is free provides the freedom to redistribute copies, either with or without modifications, either gratis or for a fee for distribution to anyone, anywhere without any charge or payment of royalty. Unlike Open source, freeware does not provide user the access to the source code.

The Selection Criteria for Software Acquisition

- When purchasing software, it is recommended that companies offering 30 day preview of software be used whenever and wherever possible so that one can use the software and evaluate its suitability.
- Compatibility of the software with the existing and proposed hardware is important.
- Ensure that the vendor is a reliable one.
- The software must be compatible with other programs that are being used.
- The installation and operating procedures of the software must be simple.
- The software must be easy to use.

A series of acquisition steps follow the finalization of software and hardware criteria. The acquisition steps are very similar to the procurement of any capital equipment. The ITT should be sent to vendors and after bids are received, they are to be analyzed under the two major headings: **Technical** and **Commercial**. After analysis, successful bidders are called for negotiations, during which all aspects including cost, delivery and installation timeframe, maintenance issues, training issues, assistance in changeover, and upgrading issues are discussed, and then a final choice made. All contract terms, including the right to audit clauses, have to be finalized before a final formal report is prepared. The report has to specify the reasons for the choice made and justify it on the basis of costs and benefits.

There is, however, a subtle difference between **Online** and **Real-time Systems**.

All real-time systems are online while the vice-versa is not true. In real-time system, an event as it occurs, is processed immediately. The output is shown to the user and the system takes corrective steps, if needed. Typical examples of real-time systems include flight control systems, industrial process control, flexible manufacturing applications and cruise control systems in automobiles. Most real-time systems are special purpose and complex systems that require a high degree of fault tolerance and are typically embedded in a larger system.

Operating Systems (OS)

An operating system (OS) is a systems software that manages computer resources and provides programmers/users with an interface to access those resources. The operating system is the command and control interface between the user and other application and systems software that commands the hardware. Also called a **Resource Manager**, an Operating System helps in the efficient management of the important resources of RAM, CPU, and various I/O devices (which include secondary storage devices also) and the information that is held in the machine.

The Operating System also enables users to interact with the machine and takes care of technical aspects of a computer's operation. It shields the user of the machine from low-level details of the machine's operation and provides frequently needed facilities. A computer system can be divided roughly into four components:

- The Hardware
- The Operating System
- The Application Programs
- The Users

Types of Operating System

1. Mainframe Systems

Mainframe computer systems were the first to tackle many commercial and scientific applications. These grew from simple batch systems, where the computer runs one – and only one- application, to time-shared systems, which allow for user interaction with the computer system.

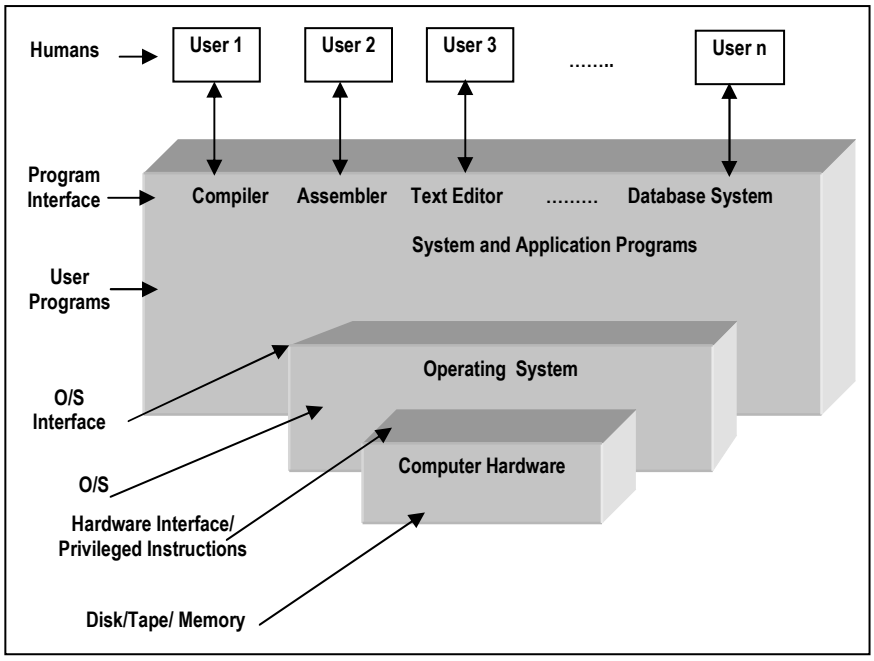


Fig. 1.31: Operating System Overview

i. **Batch Systems:** In these systems, data or programs are collected, grouped and processed at a later date. To speed up the programming, operators batched together jobs with similar needs and executed them through the computer as a group. Thus, the programmers would leave their programs with the operator who would sort programs into batches with similar needs and, as the computer became available, would run each batch. The output from each job would be sent back to the appropriate programmer. For example, Payroll, stock control and billing systems use the concept of batch systems.

ii. **Multi-programmed Systems:** It is defined as the ability to execute many programs apparently at the same time so that the CPU has always one to execute. All the jobs that enter the system are kept in the job pool which consists of all the processes residing on disk awaiting allocation of main memory. The O/S picks some jobs from the pool and keeps them in memory simultaneously. The O/S picks and begins to execute one of the jobs in memory. Eventually, the job may have to wait for some task, such as an I/O operation, to complete. In a non-multi - programmed system, the CPU would sit idle. In a multiprogramming system, the O/S switches to, and executes another job. When that job needs to wait, the CPU is switched to another job, and so on. As long as at least one job needs to be executed, the CPU is never idle. If several jobs are ready to be

brought into memory, and if there is not enough room for all of them, the process of choosing one job out of them is called **Job Scheduling**.

Iii. Time Sharing Systems: Time sharing or multi-tasking is a logical extension of multiprogramming. In this, the CPU executes multiple jobs by switching among them, but the switches occur so frequently that the users can interact with each program while it is running. A time-shared operating system allows many users to share computer simultaneously. Since each action or command in a time - shared system tends to be short, only a little CPU time is needed for each user; since the system switches rapidly on one user to the next, each user gets the impression that the entire computer system is dedicated to her/his use, even though it is being shared by many users. The program that is loaded into memory and being executed is commonly referred to as a **Process**.

2. Desktop Systems

PCs operating systems were neither multiuser nor multitasking. Instead of maximizing CPU and peripheral utilization, their goal was to maximize user convenience and responsiveness.

3. Multiprocessor Systems

Most systems that are currently in use are single-processor systems, that is, they have only one main CPU. Multiprocessor systems, also known as **Parallel systems** or **Tightly-coupled systems** have more than one processor in close communication, sharing the computer bus, the clock, and sometimes memory and peripheral devices. The main advantages of multiprocessor systems are:

i. **Increased throughput:** By increasing the number of processors, more work can be done in less time. The speed-up ration with N processors is not N, but less than N. When multiple processors cooperate on a task, a certain amount of overhead is incurred in keeping all the parts working correctly.
ii. **Economies of scale:** Multiprocessor systems save more money than multiple single-processor systems, because they can share peripherals, mass storage, and power supplies.
iii. **Increased reliability:** If functions can be distributed properly among several processors, then the failure of one processor will not halt the system, only slow it down. For example, if there are 10 processors and one fails, then each of the remaining 9 processors picks up a share of the work of the failed processor.

4. Distributed Systems

A network is a communication path between two or more systems. Distributed systems are able to share computational tasks, and provide a rich set of features to users, depending on networking for their functionality.

Networks are typecast and based on the distance between their nodes. A **LAN (Local Area Network)** exists within a room, a floor, or a building. A **WAN (Wide Area Network)**, usually exists between buildings, cities, or countries. These networks can run one protocol or several protocols and may include media like copper wires, fiber strands, and wireless transmissions between satellites, microwave dishes, and radios.

i. **Client-Server Systems:** As PCs have become faster, cheaper, and more powerful, centralized systems today act as server systems to satisfy requests generated by client systems.

   Server systems can be broadly categorized as **Compute servers and File servers**.

   - **Compute Server Systems** provide an interface to which clients can send requests to perform an action, in response to which they execute the action and send back results to the clients.
   - **File Server Systems** provide a file-system interface where clients can create update, read and delete files.

ii. **Peer-to-Peer Systems :** Peer-to-Peer networking is a method of delivering computer network services in which participants share a portion of their own resources, such as processing power, disk storage, network bandwidth, printing facilities. Such resources are provided directly to other participants without intermediary network hosts or servers. Peer-to-peer network participants are providers and consumers of network services simultaneously, which contrasts with other service models, such as traditional client-server computing where clients only consume the server's resources.

5. Clustered Systems

Clustered Systems bring together multiple CPUs to accomplish a computational task. Clustered systems consist of two or more individual systems coupled together. They share storage and are closely linked via LAN networking and provide high availability. A layer of cluster software runs on the cluster nodes. Each node can monitor one or more of the others. If the monitored machine fails, the monitoring machine can take ownership of its storage, and restart the application(s) that were running on the failed machine. The failed machine can remain down, but users and clients of the application would only see a brief interruption of service.

6. Real Time Systems

A real time system is used when rigid requirements are placed on the operation of a processor or the flow of data; thus, it is often used as a control device in a dedicated application. Systems that control scientific experiments, medical imaging systems, industrial control systems, and certain display systems are real time systems. Some

automobile-engine fuel-injection systems, home-appliance controllers and weapon systems are also real time systems. Micro C/OS-II is a typical example of RTOS.

A real time system has well-defined, fixed time constraints. Programs must be done within the defined constraints, or the system will fail. Real time systems come in two flavors: **Hard** and **Soft**.

- A **Hard Real Time System** guarantees that critical tasks will be completed on time. This goal requires that all delays in the system be bounded, from the retrieval of stored data to the time that it takes the operating system to finish any request made to it.
- In **Soft Real Time System** a critical task gets priority over other tasks, and retains that priority until it is completed.

7. Handheld Systems

Handheld systems include personal digital assistants (PDAs), such as cellular telephones with connectivity to a network like the Internet. Developers of handheld systems and applications face many challenges. These include their limited size and the speed of the processor used in the device. Some handheld devices may use wireless technology, such as BlueTooth to access email and for web browsing.

Operating System generally operates in two modes:

Supervisory mode /Monitor Mode/ System Mode/ Privileged Mode: In this privileged mode, the user has full and complete access to all the resources of the computer, bypassing all security features. Hence, this state has to be operated with extreme caution.

Restricted or General User mode: In this mode, a user can access only those resources for which rights have been granted. Normally, users are allowed to operate only in the general user mode. However, if a user wishes to perform some privileged functions, s/he has to communicate with the code running in the privileged OS area.

Core Tasks of an Operating System

The core tasks of an operating system are:

1. **Process Management :** The operating system is responsible for the following activities in connection with the process management :
    - Creating and deleting both user and system processes.
    - Suspending and resuming processes.
    - Providing mechanisms for process synchronization.
    - Providing mechanisms for process communication.
    - Providing mechanism for deadlock handling.

2. **Main Memory Management:** The operating system is responsible for the following activities in connection with the memory management :
   - Keeping track of the parts of memory that are currently being used and by whom.
   - Deciding the processes that are to be loaded into memory when memory space becomes available.
   - Allocating and de-allocating memory space as per need.
3. **File Management:** The operating system is responsible for the following activities in connection with the file management :
   - Creating and deleting files.
   - Creating and deleting directories.
   - Supporting primitives for manipulating files and directories.
   - Mapping files onto secondary storage.
   - Backing up files on stable storage media.
4. **I/O System Management:** One of the purposes of an operating system is to hide the peculiarities of specific hardware devices from users, which is done by I/O subsystem itself. The I/O subsystem consists of :
   - A memory management component that includes buffering, caching and spooling.
   - A general device called driver interface.
   - Drivers for specific hardware devices.
5. **Secondary Storage Management :** The operating system is responsible for the following activities in connection with the disk management:
   - Free space management.
   - Storage allocation.
   - Disk scheduling.
6. **Networking:** The processors in the system are connected through a **communication network**, configured in a number of different ways. A **distributed system** is a collection of processors that do not share memory, peripheral devices, or a clock. Instead, each processor has its own local memory and clocks and the processors communicate with one another through various communication lines, such as high-speed buses or networks. Different protocols like FTP (File Transfer Protocol), NFS (Network file System) ad HTTP (Hypertext Transfer Protocol) are different protocols used in communication between different computers.
7. **Protection System:** Protection is a mechanism for controlling the access of programs, processes or users to the resources defined by a computer system.

This mechanism provides means for specification of controls to be imposed and means for enforcement.

Protection mechanism improves reliability by detecting latent errors at the interfaces between computer subsystems.

8. **Command-Interpreter System:** Command interpreter is an interface between the user and the operating system that usually resides in the kernel. Many commands are given to the operating system by Control statements. When a new job is initiated in a batch system or when a user logs on to a time - shared system, a program that reads and interprets control statements is executed automatically. This program is called the **Control-Card Interpreter** or the **Command-Line Interpreter** but quite often called the S**hell**. Its function is to get the next command statement and execute it.

Services provided by an Operating System

An operating system provides an environment for the execution of programs, by making available services to the users of those programs. The services provided differ from one operating system to another. Some of these services are:

i. **Program Execution:** The system must be able to load a program into memory and to run that program. The program must be able to end its execution, either normally or abnormally (indicating error).
ii. **I/O Operations:** A running program may involve a file or an I/O device for specific devices, or special functions may be desired (such as a tape drive or CRT screen). For efficiency and protection, users usually cannot control I/O devices directly. Therefore, an operating system must provide a means to do I/O.
iii. **File-System Manipulation:** Programs need to read and write files. They also need to create and delete files by name.
iv. **Communications:** In many circumstances, one process needs to exchange information from another process. Such communications can occur either between two processes that are executing on the same computer or between processes executing on different computers tied together by a computer network.
v. **Error Detection:** Errors may occur in the CPU and memory hardware (such as power failure), in I/O devices (a lack of paper in the printer),and in the user program (such as an arithmetic overflow). For each type of error, the operating system should take the appropriate action to ensure correct and consistent computing.
vi. **Resource Allocation:** When multiple users are logged in the system or multiple jobs are running at the same time, resources must be allocated to each of them. For instance, CPU scheduling routines that take into account the speed of the

CPU, the jobs that must be executed, the number of registers available, and other factors are addressed by the operating system.

vii. **Accounting:** Statistics related to the number of users, the nature of their needs and the kinds of computer resources available is a valuable tool for researchers who wish to reconfigure the system to improve computing services.

viii. **Protection:** When several disjointed processes execute concurrently, it should not be possible for one process to interfere with the other, or with the operating system itself. Protection involves ensuring that all access to system resources is controlled. Security of the system from outsiders is also important. Such security starts with each user having to authenticate his self to the system, usually by means of a password to access the resources.

Database Management Systems

Introduction

Survival of organizations depends on their ability to quickly find the right information at the right time so that appropriate decisions can be taken. The amount of information an organization handles is exploding day - by - day; unless they use a Database Management System Software, managing the data becomes virtually impossible. Prior to databases, organizations used the file concept for storing data, in which the data used to be stored in flat-files, and computer programs were written to access this data. The main drawbacks of this method are:

- **Very high degree of data redundancy:** Since data is required by more than one application, it is recorded in multiple files, thus causing data redundancy. High data redundancy leads to multiple updating, problems of maintaining integrity and several others. Moreover, data redundancy may result in inconsistency, i.e., one might update data in one file and forget to update the same data stored in another file.
- **Difficulty in accessing data:** Programmers may have to write a new application program to satisfy an unusual request.
- **Limited Data Sharing:** In the flat file method, data sharing is very limited.
- **Security and Integrity problems:** Conventional file systems offer only limited facilities with regard to maintaining security and integrity of data.
- **High degree of dependency between data and programs:** Data in flat files are accessed using application programs. If any change in the data structure or format is made in the data file, a corresponding change has to be made in the application program and vice versa. These problems led to the development of databases and DBMS software.

Definition of a Database

A **database** is defined as a centralized repository of all the inter-related information of an organization stored with the objectives of :

i. **Permitting data sharing:** Being centralized in nature, data in a database is shared amongst users and applications.
ii. **Minimizing data redundancy:** Since only one copy of the data is shared by the users, redundancy is minimal.
iii. **Enforcing standards:** Overall standards specified by the company's policies can be enforced.
iv. **Providing higher degree of data security and integrity:** Since data is available at a single centralized location, high degree of data security and integrity can be achieved.
v. **Achieving program/data independence:** In contrast to conventional data files, degree of dependency between program and data is very low. That is, application programs are insulated from changes in the way the data is structured and stored. In other words, application programs can be altered without changes in the structure of the data and the database can be changed without the need for reprogramming.
vi. **Faster application program development**.

Example of Database

Consider a **UNIVERSITY** database in which information related to students, courses and grades is to be stored. To construct the **UNIVERITY** database, the data may be organized in five files, each of which stores data of the same type.

- The STUDENT file stores data of each student.
- The COURSE file stores data of each course.
- The SECTION file stores data on each section of a course.
- The GRADE_REPORT file stores the grades that students receive in the various sections they have completed.
- The PREREQUISITE file stores the prerequisites of each course.

These databases have many types of records and have many relationships among the records.

Definition of a DBMS

A **Database Management System (DBMS)** is a software that manages a database and provides facilities for database creation, populating data, data storage, data retrieval and data access control. It is a collection of programs that enables users to create and maintain a database. The DBMS is thus a **general-purpose** software

system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications

- **Defining** a database involves specifying the data types, structures and constraints for the data to be stored in the database.
- **Constructing** a database is the process of storing the data itself on some storage medium that is controlled by the DBMS.
- **Manipulating** a database includes such functions as querying the database to retrieve specific data, updating it to reflect changes in the mini - world, and generating reports from the data.

Sharing a database allows multiple users and programs to access database concurrently.

Other important functions provided by the DBMS include **protecting** the database and **maintaining** it over a long period of time.

Protection includes both system protection against hardware or software malfunction, and security protection against unauthorized or malicious access.

A typical large database may have a life cycle of many years, so that DBMS must be able to **maintain** the database system by allowing the system to evolve as requirements change over time.

Database Systems

The database and DBMS software is called **Database System**. A database system has four major components: **Data**, **Hardware**, **Software** and **Users,** which coordinate with each other to form an effective database system.

- **Data:** Being an important component of the system, most organizations generate, store and process a 1arge amount of data. The data acts as a bridge between machine parts that is, hardware and software and the users, who access it directly or through some application programs. The data stored in the system is partitioned onto one or more databases. A database, then, is a repository for stored data. In general, it is both **integrated** and **shared**.

  By **integrated**, it is meant that the database is a unification of several otherwise distinct data files. The individual pieces of data in the database may be **shared** among several different users in the sense that each of them may have access to the same piece of data. Such sharing is really a consequence of the fact that the database is integrated.

- **Hardware:** The hardware consists of the secondary storage devices such as magnetic disks (hard disk, zip disk, floppy disks), optical disks (CD-ROM), magnetic tapes, etc. on which data is stored together with the I/O devices

(mouse, keyboard, printers), processors, main memory, etc. which are used for storing and retrieving the data in a fast and efficient manner. Since database can range from those of a single user with a desktop computer to those on mainframe computers with thousands of users, therefore proper care should be taken for choosing appropriate hardware devices for a required database. The hardware consists of the secondary storage volumes, disks, drums, etc. on which the database resides, together with the associated devices, control units, channels, and so forth.

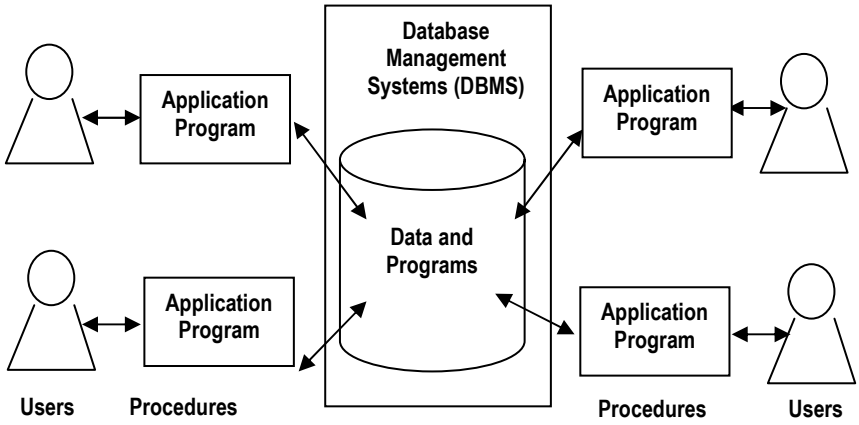


Fig. 1.32: Simplified Picture of a Database

- **Software:** The software part of a DBMS acts as a bridge between user and the database. In other words, software interacts with users, application programs, and database and files system of a particular storage media (hard disk, magnetic tapes etc.) to insert, update, delete and retrieve data. For performing operations such as insertion, deletion and updation, we can either use Query Languages like SQL, QUEL or application software like Visual Basic.
- **Users:** The broad classes of users are:
    - **Application Programmers and System Analysts:** System analysts determine the requirements of end users, especially naive and parametric end users, and develop specifications for canned transactions that meet these requirements. Application programmers implement these specifications as programs, and then they test, debug, document, and maintain these canned transactions.
    - **End Users:** These are the people who require access to the database for querying updating and generating reports. The database exists primarily for their use.

- **Database Administrator (DBA):** DBA is responsible for authorization access to the database, for coordinating and monitoring its use, and for acquiring the needed software and hardware resources.
- **Database Designers:** These are responsible for identifying the data to be stored in the database for choosing appropriate structures to represent and store this data.

Advantages of Database Systems

- Redundancy and inconsistencies can be reduced.
- Better services can be provided to users.
- The data can be shared.
- Cost of developing and maintaining systems is lowered.
- Standards can be enforced.
- Security restrictions can be applied.
- Integrity can be maintained.
- Conflicting requirements can be balanced.

Data Abstraction

The characteristic that allows program-data independence and program-operation independence is called **Data Abstraction**. Abstraction means that the system hides details of how data is stored and maintained and no information is revealed to users.

Based on the levels of abstraction, in a DBMS, we have three levels as shown in Fig. 1.33.

1. **Level 1: External level or View Level or User Level:** This is the highest level that describes portions of the database for users. Every user has a limited view of the entire database. To illustrate, an inventory employee may view the inventory data but not payroll data.

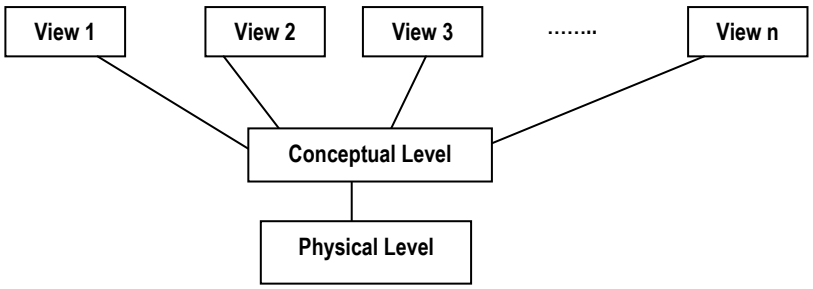


Fig. 1.33 : Levels of Data Abstraction

2. **Level 2: Conceptual Level:** This level describes how the data has to be stored, what relations exist between data items, and how data is to be modeled.
3. **Level 3: Internal Level or Physical Level:** The focus of this level is on details of how and in what format the data are stored (e.g. index, B-tree, hashing etc.,) and how these data can be accessed.

DBMS ARCHITECTURE or THREE SCHEMA ARCHITECTURE

The goal of three-schema architecture is to separate user applications and physical database. In this architecture, schemas can be defined at the following three levels: **External schema**, **Conceptual schema** and **Internal schema**.

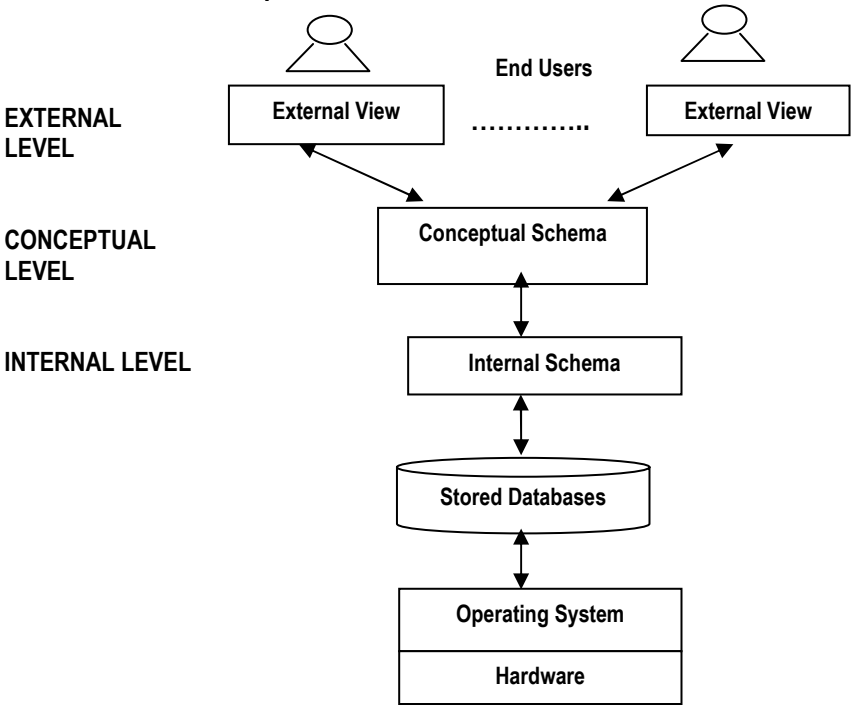


Fig. 1.34: DBMS Architecture

External Schema (Individual User View)

i. External schema provides the necessary interface for users to perform various activities on the database and provides their view of the database.
ii. It is user-friendly and focused on users who use the system and hides the rest of the database from that user group.
iii. The design of the external schema is guided by end user requirements.

Conceptual schema (Community User View)

i. The conceptual schema describes the structure of the whole database for a community of users.
ii. The conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints. In other words, it provides an abstract way of representing the database.

Internal Schema (Physical or Storage View)

i. The internal schema describes the physical structure of the database.
ii. It specifies the file organizations that are used and the complete details of the data storage and access paths for the database.

The three schema architecture is a convenient tool with which the user can visualize the schema levels in a database system. The three schemas are only descriptions of data, the only data that actually exists is at the physical level.

Did you know?

MySQL is one of the popular RDBMS that runs a server with multi-user access capabilities. The source code of MySQL is available under GNU General Public License scheme.

DATA INDEPENDENCE

It is defined as the capacity to change the schema at one level of a database system without having to change the schema at the next higher level. Two types of data independence are:

a. Logical Data Independence
It is the capacity to change the conceptual schema without having to change external schemas or application programs. We may change conceptual schema to expand the database (by adding a record type or data item), to change constraints, or to reduce the database (by removing a record type or data item).

b. Physical Data Independence
It is the capacity to change the internal schema without having to change the conceptual schema. Hence, the external schema need not be changed at all. Changes to the internal schema may be needed because some physical files have to be recognized – for example, by creating additional access structures – to improve the performance of retrieval or update.

DATA MODELS

One fundamental characteristic of the database approach is that it provides some level of data abstraction by hiding details of data storage that are not needed by most

database users. A **Data Model** is defined as a collection of concepts that can be used to describe the structure of a database in terms of data types, relationships, and constraints that hold for the data. A data model, in other words, provides a means to achieve data abstraction.

There are many data models that can be categorized according to the types of concepts used to describe the database structure. These are:

I. Database Model

A database model is a theory or specification describing how a database is structured and used. Some of the common models are:

a. Hierarchical Model:

In this model, data is organized into a tree-like structure. Its basis is the hierarchical tree structure, which is made up of nodes and branches. A node is a collection of data attributes describing an entity. The following are the characteristics of a tree structure.

i. The highest node in the tree is called the root node.
ii. Every node in the tree has only one parent.
iii. A parent node can have any number of children.
iv. No loops or cycles are permitted.
v. Every node has to be accessed only through its parent node.

Fig. 1.27 shows how data is represented using hierarchical model.

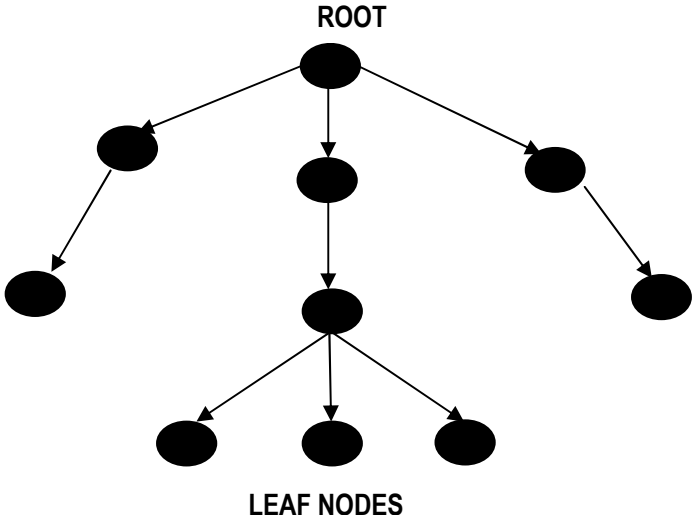


Fig. 1.35: Hierarchical Model

A hierarchical model supports only “one-to-one” and “one-to-many” relationship. The main problem in this model is that unduly complex operations are involved in the insertion of nodes in a tree. Moreover, there is also the problem of triggered deletion. When a node is deleted from a tree, its children get deleted automatically. This cascading operation is termed triggered deletion.

b. Network or Plex Model

This model has three basic components: **Record type**, **Data items** and **Links**.

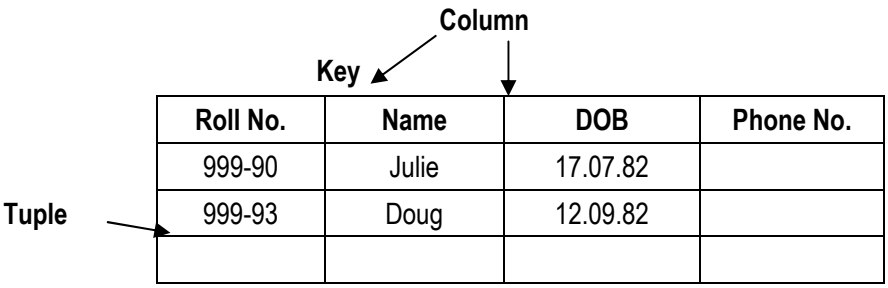
- Data are represented as a collection of **Records**.
- Relationships are represented as **Links**.
- Each Record is a collection of **Data Items**.

This model, unlike the hierarchical model supports “**many-to-many** “relationship. For example, many customers have many accounts. The main disadvantage of this model is its complexity. Also, when the database is re-organized, there is every possibility that data independence might be lost.

c. Relational Model

Relational model is the most common model used these days in a majority of commercial DBMS packages. This model has a very strong mathematical backing through relational algebra and relational calculus. This model represents the database as a collection of **Relations** and is governed by the following rules:

i. Data is represented in a two-dimensional table, in rows and columns.
ii. Columns describe the attributes.
iii. Each column in the table has a unique name.
iv. All the entries in any column are of the same type.
v. Each column has a domain, a set of possible values that can appear in that column.
vi. A row in the table is called **Tuple**.
vii. Ordering of rows and columns is insignificant.
viii. Duplicate rows are not allowed.
ix. All data items stored in the columns are atomic in nature, that is, they cannot be split further without loss of information.
x. In many tables, there is a column called the **Key Column** whose value is unique and cannot be null.



Roll No.	Name	DOB	Phone No.
999-90	Julie	17.07.82	
999-93	Doug	12.09.82	

Fig. 1.36: Example of Relational Model

Relational Data Integrity

In order to maintain integrity (accuracy) of data in the database, relational model specifies several types of integrity constraints. Its major types are:

- **Domain constraint:** This means that all the values in the column of a table must be from the same domain.
- **Entity integrity:** This rule is to ensure and assure that the data values for the primary key are valid and not null.
- **Referential Integrity:** In a relational data model, associations between tables are defined by using foreign keys. The referential integrity rule states that if there is a foreign key in one table, either the foreign key must match the primary key of the other table or else the foreign key value must be null.
- **Operational Constraints:** These constraints are enforced on the database by the business rules or by the environment where the table operates. The database must not violate these constraints.

d. Object Oriented Model:

The object-oriented model is based on a collection of objects, like the E-R model.

i. An object contains values stored in instance variables within the object.
ii. Thus objects contain objects to an arbitrarily deep level of nesting.
iii. An object also contains bodies of code called **Methods** that operate on the object.
iv. Objects that contain the same types of values and the same methods are grouped into classes.
v. A class may be viewed as a type definition for objects.
vi. The only way in which one object can access the data of another object is by invoking the method of that other object. This is called sending a message to the object.
vii. Internal parts of the object, the instance variables and method code, are not visible externally.

viii. The result is two levels of data abstraction.

- Each object has its own unique identity, independent of the values it contains, two objects containing the same values even are distinct.
- Distinction is created and maintained in physical level by assigning distinct object identifiers.

For example, consider an object representing a bank account.

- The object contains instance variables number and balance.
- The object contains a method pay-interest which adds interest to the balance.
- Under most data models, changing the interest rate entails changing the code in application programs. In the object-oriented model, this only entails a change within the pay-interest method.

e. Star Schema:

The star schema is the simplest style of data warehouse schema which consists of a few fact tables referencing any number of dimension tables. **Fact tables** contain the quantitative or factual data about a business which is often numerical, and can be of many columns and millions or billions of rows. **Dimension tables** are usually smaller and hold descriptive data that reflect the dimensions or attributes of a business.

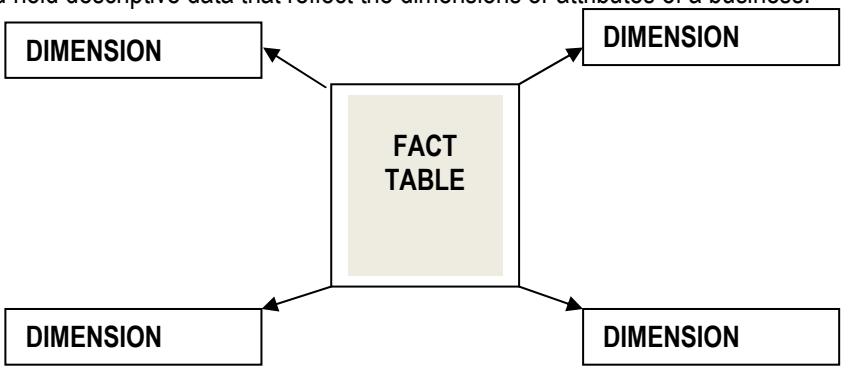


Fig. 1.37: The Star Schema

For example, a fact table in a sales database, implemented with a star schema, might contain the sales revenue for the products of the company from each customer in each geographic market over a period of time. The dimension tables in this database define the customers, products, markets, and time periods used in the fact table.

II. Data Structure Diagram

A Data Structure Diagram (DSD) is a diagram and data model that describes

conceptual data models by providing graphical notations which document entities and their relationships, and the constraints that bind them. The basic graphic elements of DSDs are boxes, representing entities, and arrows, representing relationships. Data structure diagrams are most useful for documenting complex data entities. In DSDs, attributes are specified inside the entity boxes, while relationships are drawn as boxes composed of attributes which specify the constraints that bind entities together. Fig. 1.38 illustrates a DSD.

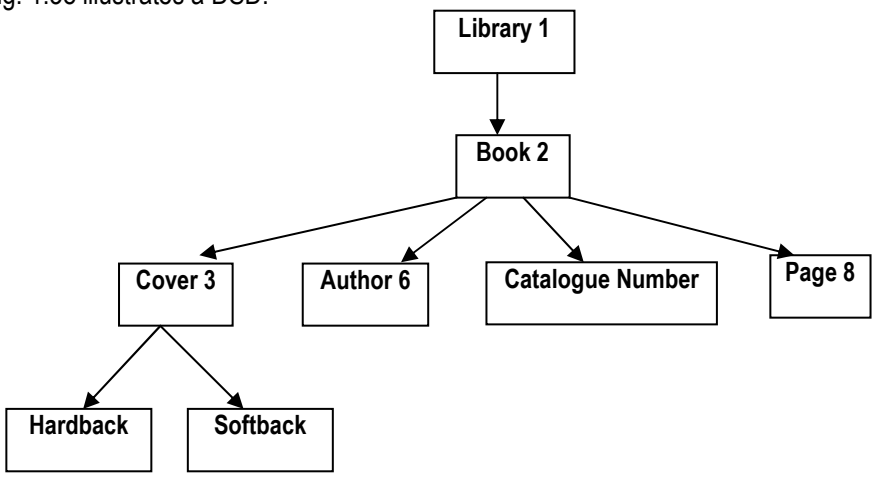


Fig. 1.38: DSD Example

III. Entity-Relationship Model

One of the most commonly used data model is the **Entity – Relationship Model** also referred to as E - R Diagram.

E-R Diagram

An Entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database. An **entity** is defined as a distinguishable object that exists in isolation and is described by a set of **attributes**. For example, color and fuel are attributes for the entity car. A **relationship** is an association among several entities. Again, taking the same example of a car, the registration number of a car associates it with the owner. The set of all entities or relationships of the same type is called the **entity set** or relationship set.

E-R diagrams have yet another element of **cardinality,** which expresses the number of entities to which an entity can be associated via a relationship set. Using the entity's attributes and cardinalities, the overall logical structure of a database can be expressed graphically by an E-R diagram.

ER diagrams often use symbols to represent three different types of information.

i. **Boxes** represent entities.
ii. **Diamonds** represent relationships and,
iii. **Ovals** represent attributes.

Types of relationships

Mapping Cardinalities: These express the number of entities to which an entity can be associated via a relationship. For binary relationship sets between entity sets A and B, the mapping cardinality must be one of :

i. **One-to-one:** An entity in A is associated with at the most one entity in B, and an entity in B is associated with at the most one entity in A.
ii. **One-to-many:** An entity in A is associated with any number in B. An entity in B is associated with at the most one entity in A.
iii. **Many-to-one:** An entity in A is associated with at the most one entity in B. An entity in B is associated with any number in A.
iv. **Many-to-many:** Entities in A and B are associated with any number from each other.

Fig. 1.39 shows the various types of relationships with examples.

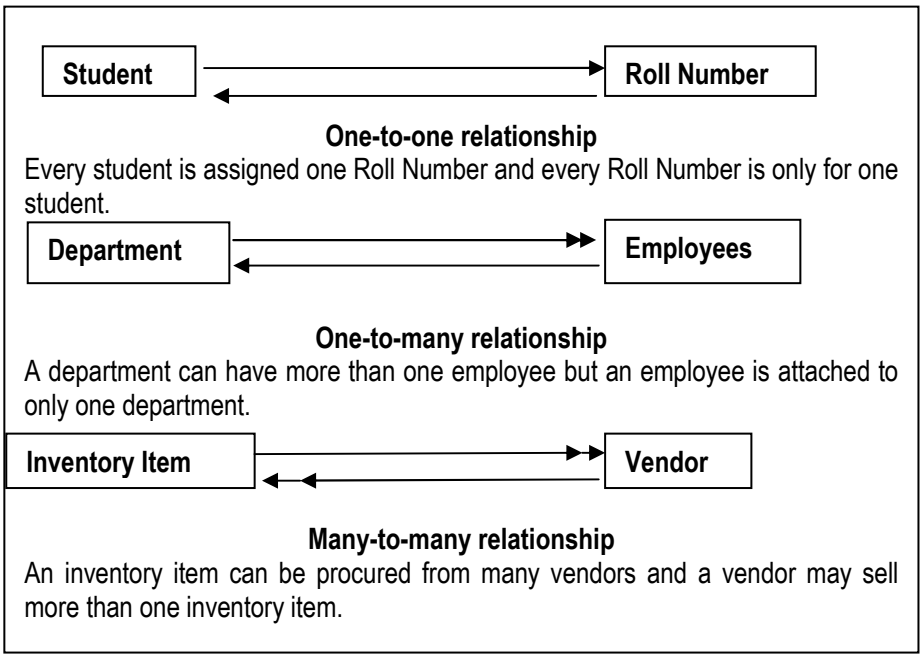


Fig. 1.39: Types of Relationships

Data Dictionary

Data Dictionary is a tool that enables one to control and manage information in a database. It is actually a documentation of the database that provides a detailed description of every data that is in it. The dictionary provides information on the following:

i. The way data is defined.
ii. Types of data present.
iii. Relationship among various data entities and their representation formats.
iv. Keys for the database.
v. People who access data and access rules for every user.

The dictionary also generates reports on:

- Various data elements, their characteristics, entities and relationships.
- Frequency of the use of data.
- Responsibilities of various users.
- Access control information.

A data dictionary is used as an effective tool by the administration in the design, implementation, and operation phases of the database.

Database Manager

A database manager is a program module of the DBMS which provides the interface between the data stored in a database and the application programs and queries submitted to the system.

In large applications, the size of databases may extend to gigabytes and whenever any operation involves data, it is moved from the hard-disk to RAM and processed. Database Manager simplifies and facilitates this process.

Hence the database manager module in a DBMS is responsible for:

- Interacting with the Operating System for storing data on the disk.
- Translating DML statements into low-level file system commands for storing, retrieving and updating data in the database.
- Enforcing Integrity and ensuring that there is no violation of consistency constraints.
- Enforcing Security.
- Backing up and recovery of the database.
- Providing concurrent control in large systems when there are concurrent users.

Did you know?

SQL Injection is a technique that exploits the vulnerabilities of a database, wherein strong SQL statements are used by the attacker to confuse the DBMS and hence return unexpected execution and privileged access.

Overall System Structure

Although a typical DBMS has three major schemas of External, Conceptual and Internal, the functions of these layers are carried out by different modules. Fig. 1.40 shows an overall system structure. Some functions (e.g. file systems) may be provided by the operating system itself.

The other major components are:

i. **File manager,** which manages allocation of disk space and data structures used.
ii. **Database manager**, which acts as an interface between low-level data and application programs and queries.
iii. **Query processor,** which translates statements in a query language into low-level instructions so that they can be executed by the database manager.
iv. **DML pre-compiler,** which converts DML statements, embedded in an application program to normal procedure calls in a host language. The pre-compiler basically interacts with the query processor.
v. **DDL compiler,** which converts DDL statements into a set of tables.

Database Languages

A DBMS provides a comprehensive set of facilities to perform the following actions:

i. Creating, Modifying, Deleting database objects.
ii. Inserting, Updating and Deleting data in the database.
iii. Performing the process of querying the database.
iv. Controlling access to the database.
v. Providing facilities for data integrity and consistency.

These facilities are grouped under database languages. The three important classifications of database languages are:

1. Data Definition/Description Language (DDL);
2. Data Manipulation Language (DML); and
3. Data Control Language (DCL).

1. Data Definition Language (DDL)

Data Definition Language (DDL) is used by the DBA and by database designers to define both conceptual and internal schemas. The DBMS will have a DDL compiler whose function is to process DDL statements in order to identify descriptions of the schema constructs and to store the schema description in the DBMs catalogue. DDL statements are used to define a database, build and modify the structure of tables and other objects in the database, and are also used for establishing constraints. Data in the database is not touched by the DDL statement. When one executes a DDL statement, it takes effect immediately.

2. Data Manipulation Language (DML)

Once the database schemas are compiled and the database is populated with data, users must have some means to manipulate the database. Typical manipulations include retrieval, insertion, deletion, and modification of data. The DBMS provides a set of operations or a language called the **Data Manipulation Language (DML)** for these purposes. DML statements are used to maintain and query a database.

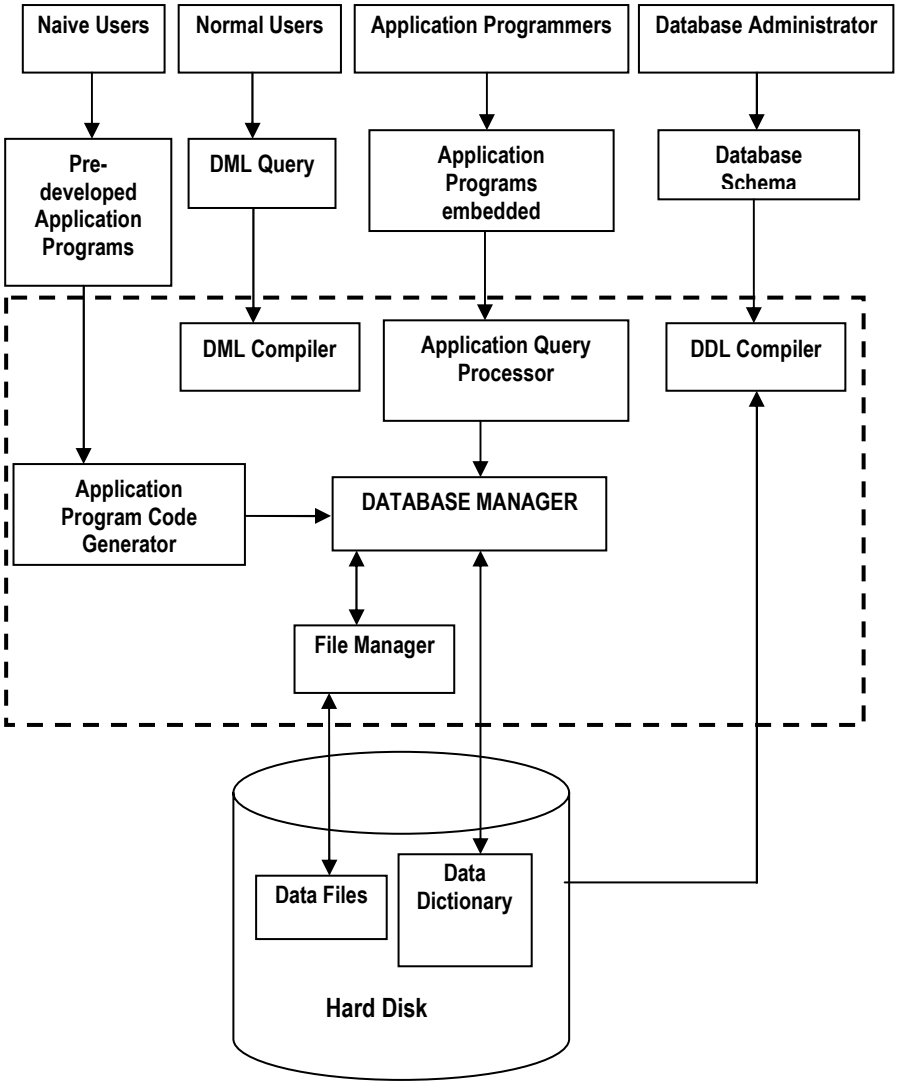


Fig. 1.40 : Overall System Architecture

3. Data Control Language

A **Data Control Language (DCL)** is a computer language and a subset of SQL, used to control access to data in a database. DCL Statements control a database, including administering privileges and committing data.

Structured Query Language (SQL)

SQL (Structured Query Language) is a database sub-language used for various actions involving relational databases. It was developed by IBM Research Lab in the mid 70s and standardized by ANSI in 1986. SQL uses the terms **table**, **row** and **column** for the formal relational model terms **relation**, **tuple**, and **attribute**.

Database Access Methodology

The process of accessing the database involves three stages:

1. **Stage 1: Identification Stage** - Here users identify themselves to the system. Normally this is done by using log-on ids.
2. **Stage 2: Authentication Stage** - After identification, the users have to authenticate themselves. Normally, passwords are the most common method for authentication. However, today apart from password, other techniques such as biometric identification techniques, Kerberos (trusted third party authentication) have also become more widespread.
3. **Stage 3: Authorization Stage** - In this stage, what operations users can perform, on which objects and where these operations can be performed are decided. For this, the system maintains an authorization matrix, which contains information related to subject, object, action and constraint.

Classification of Database Users

Every user of a database falls into one of the following categories:

i. **Application programmers**, who are experts and interact with the database by writing application programs in various High Level Languages (HLL) in which DML commands are embedded. Application programmers are well aware of the syntax and programming aspects of HLL. The module of DML compiler converts the DML statements to normal procedure calls in a host language, which is then converted into an object code.

ii. **Normal users,** who interact with the system without writing application programs. They prepare queries in DML and are then submitted to a query processor which breaks the DML statement down into object code.

iii. **Naive users,** who interact with the system by using pre-developed application programs, which generate the application code that interacts with the database manager.

iv. **Database Administrator,** the person who has the overall and central control over data and programs accessing that data.

Roles and duties of a Database Administrator

The Database Administrator (DBA) is responsible for all the technical operations in a database in an organization. The major responsibilities of a DBA are:

1. Management Tasks

a Perform the process of liaising with management.
b Perform the process of liaising with database users for their needs.
c Perform the process of liaising with database vendors.
d Perform the process of liaising with the Data Administrator (DA).
e Acquire latest information on database technologies and tools.
f Prepare the project status report on the usage of databases.
g Prepare staff requirements.
h Prepare budgets.

2. Security tasks

a Monitoring and maintaining security of databases.
b Granting access rights and revoking them to users.

3. Technical Tasks

a Monitoring the databases, perform database tuning and optimizing database performance.
b Maintaining availability (24 x 365) and ensuring integrity of databases.
c Installing any patch software, if necessary.
d Managing database backups and restorations.
e Guiding application programmers in their work.
f Performing the process of capacity planning.
g Performing the process of normalization, demoralization, indexing, and defining views.
h Framing procedures for purging and archiving data.

4. General Tasks

a Providing education and training suggestions.
b Ensuring that there is a congenial environment for work.

As mentioned in the management tasks, a DBA has to liaise with the **Data Administrator (DA**). A DA is a person who identifies the data requirements of

business users by creating an enterprise data model. He identifies the data owners and sets standards for control and usage of data.

Summary

This chapter introduces the basic concept of computers and their types. It explains the detailed architecture of a computer and various Input and output devices used in a computer system. The reader is introduced to various hardware monitoring procedures, and their acquisition plans. The theory on operating system and its functions, Database management Systems, data models, data independence and role of Database manager are highlighted in detail.

Sources:

1. Saunders D H, Computers Today, McGraw Hill International Edition, New York.
2. Leon A and Leon M, Introduction to Information Systems, Vijay Nicole Private Ltd. Chennai, 2004.
3. Bahrami A, Object Oriented Systems Development -An unified approach, McGraw Hill International edition, New York, 1999.
4. Beck, Leland L, Systems Software, Pearson Education India, New Delhi, 2002.
5. Berson A and Anderson, Sybase and Client-Server Computing, McGraw Hill International edition, New York, 1999.
6. Laudon and Laudon, Management Information Systems, Prentice Hall of India, New Delhi, 1999.
7. Pressman R, Software Engineering, a Practioner's approach, Tata McGraw Hill, New Delhi, 1999.
8. Rumbaugh J, et al, Object Oriented Modeling and Design, Prentice Hall of India, New Delhi, 2002.
9. Sommerville I, Software Engineering, Addison Wesely Publishing.
10. Stallings W, Operating Systems, 4th edition, Prentice Hall of India, New Delhi, 2003.
11. Weber R, Information Systems Control and Audit, Pearson Education India, New Delhi, 2002.
12. Abbey M and Corey M J, Oracle8, A beginners guide, Tata McGraw Hill, New Delhi, 1999.
13. Leon A and Leon M, Database Management Systems, Vikas Publishing, Chennai, 2002
14. Ramakrishnan R, Database Management Systems, McGraw Hill International Edition, New York, 2001.
15. Ullman J, Widom, A first course on database systems, Pearson Education India, New Delhi, 2002.

16. Weber R, Information Systems Control and Audit, Pearson Education India, New Delhi, 2002.
17. http://www.firstsql.com
18. C.J.Date, An Introduction to Database systems, Third Edition Vol. 1, Narosa Publishing House, New Delhi, Madras, Bombay, Calcutta.
19. Elmsari, Navathe, Somayajulu, Gupta, Fundamentals of Database Systems IV Edition, Pearson Education.
20. Silberschatz, Galvin, Gagne, Operating System concepts VI Edition,Wiley.

Questions

1. Which of the following is not a type of Software?
   a. Programming Software
   b. Application Software
   c. System Software
   d. Firmware

2. Which of the following is not a level of abstraction?
   a. Conceptual Level
   b. Chemical Level
   c. User Level
   d. Physical Level

3. Allows program-data independence and program-operation independence.
   a. Data Abstraction
   b. Encryption
   c. Cryptography
   d. None of these

4. ___ provide a file-system interface where clients can create, update, read and delete files.
   a. Compute Server Systems
   b. Peer-to-peer Systems
   c. File Server Systems
   d. None of these

5. Which of the following is not an impact printer?
   a. Dot-Matrix Printer
   b. Thermal Printer
   c. Line Printer
   d. Daisy wheel Printer

6. Which of the following is not a non-impact printer?
   a. Dot-Matrix Printer
   b . Thermal Printer
   c . Laser Printer
   d . Ink-Jet Printer

7. Operating system is an example of _____.
   a. Programming Software
   b. Application Software
   c. System Software
   d. None of these

8. Which of the following is not an operating system?
   a. Mainframe Systems
   b. Multiprocessor Systems
   c. Distributed Systems
   d. Peer-to-peer systems

9. A ___ is a software that manages a database and provides facilities for database creation, populating data, data storage, data retrieval and data access control.
   a. Mainframe Systems
   b. Peer-to-peer systems
   c. Distributed Systems
   d. Database Management System (DBMS)

10. ASCII stands for _____ .
   a. American Standard Code for Information Interoperability
   b. American Standard Code for Information Interchange
   c. American Standard Code for Interchange Information
   d. American Standard Conduct for Information Interchange

11. Which of the following is not a type of Database Model?
   a. Object-oriented Model
   b. Hierarchical Model
   c. Relational Model
   d. Network Model

12. _____ is defined as the collection of computer programs, procedures and documentation that performs different tasks on a computer system.
   a. Hardware
   b. CPU
   c. Software
   d. DBMS

13. Which stage is not involved in the process of accessing the database?
    a. Identification stage
    b. Authentication stage
    c. Authorization stage
    d. Protection stage

14. _____ is defined as the ability to execute many programs apparently at the same time so that CPU always has one to execute.
    a. Batch processing
    b. Multiprocessing
    c. Multi-user
    d. Real time

15. _____is an input device consisting of a stick that pivots on a base and reports its angle or direction to the device it is controlling.
    a. Keyboard
    b. Joystick
    c. Mouse
    d. Scanner

16. Which of the following is not an example of an input device?
    a. Keyboard
    b. Joystick
    c. Visual Display Terminal
    d. Scanner

17. Which of the following is not an example of an output device?
    a. Printer
    b. Plotter
    c. Visual Display Terminal
    d. Scanner

18. _____ is a part of the system software which informs the computer how to use a particular peripheral device.
    a. Device Driver
    b. Linker
    c. Loader
    d. Compiler

19. DDL stands for _____.
    a. Defining Data Language
    b . Data Definition Language
    c. Direct Definition Language

d. Direct Data Language

20. Which of the following is not a component of a Network Model?
    a. Record type
    b. Data items
    c. Links
    d. Tuple

21. A _____is a collection of processors that do not share memory, peripheral devices, or a clock.
    a. Distributed system
    b. Embedded System
    c. Real Time system
    d. Batch Systems

22. Which of the following is not a computer component?
    a. ROM
    b. SRAM
    c. DRAM
    d. EEPROM

Answers:

1 d	2 b	3 a	4 c	5 b	6 a
7 c	8 d	9 d	10 b	11 a	12 c
13 d	14 b	15 b	16 c	17 d	18 a
19 b	20 d	21 a	22 b		

2 Introduction to Computer Networks

Learning Objectives

To understand

- Basic concepts of Computer network and its types.
- Various modes of communication.
- The concept of multiplexing and its categories: FDM, TDM and WDM.
- Switching techniques in data transmission.
- Principles of MODEM.
- Physical and Logical topologies used in Computer network.
- Medium used in data transmission: Guided, Unguided and their categories.
- Various factors influencing the use of media.

Introduction

By themselves, computers are powerful tools. When they are connected to each other in an appropriate fashion, they become more powerful because the resources that each computer provides can be shared with other computers. The purpose of the interconnection is to create opportunities for sharing resources and information.

Computer Network

The term **"Computer Network"** means a collection of autonomous computers interconnected by some means that enables them to exchange information and share resources.

In the simplest sense, networking means connecting computers so that they can share files, printers, applications and other computer- related resources. The advantages of networking computers are:

- Users can save shared files and documents on a file server rather than storing them in their individual computers.
- Users can share resources like network printer, which costs much less than having a locally attached printer for each user's computer.

- Users can share applications running on application servers, which enables users to share data and documents, to send messages, and to collaborate.
- The job of administering and securing a company's computer resources can be concentrated on a few centralized servers.

One of the basic traits of social beings is their need to network, to share information. So when one desires information available with another being, there is need to network. Probably one of the most powerful benefits of computers is their ability to interconnect and network. Networking helps in sharing information resources across geographic locations.

To understand computer networks and the modalities by which they are connected and how they share information, it is necessary to know clearly some of the fundamentals of communication technology.

Network Characteristics

- Resource Sharing
- High Reliability
- Low Cost
- Scalability
- Communication Medium

Fundamentals of communication

Any communication system should have

1. a **Sender**, who wants to transmit a message,
2. a **Receiver**, for whom the message is intended,
3. a **Message**,
4. the **Medium** used for sending the message, and
5. a **Protocol**, that is, a set of rules for making communication possible.

The main goal of a communication system is that the message to be sent

1. must be accurately delivered only to the intended recipient(s),
2. must be delivered always on time, and
3. the contents of the message must not be tampered during transmission.

Data Transmission

Data transmission or **Digital communications** is the physical transfer of data (a digital bit stream) over a point-to-point or point-to-multipoint communication channel. Examples of such channels are copper wires, optical fibers, wireless communication

channels, and storage media. The data is often represented as an electro-magnetic signal, such as an electrical voltage signal, a radio wave or microwave signal or an infra-red signal.

While analog communications represent a continuously varying signal, a digital transmission can be broken down into discrete messages. The messages are either represented by a sequence of pulses by means of a line code (baseband transmission), or by a limited set of analog wave forms (pass band transmission), using a digital modulation method.

Data transmitted may be a digital message originating from a data source, like a computer or a keyboard. It may also be an analog signal such as a phone call or a video signal, digitized into a bit-stream, like pulse-code modulation (PCM) or more advanced source coding data compression schemes. This source coding and decoding is done by codec equipment.

Transmission Modes

A given transmission on a communications channel between two machines can occur in several ways. The transmission is characterized by:

a. The direction of the exchanges.
b. The number of bits sent simultaneously.
c. Synchronization between the transmitter and receiver.

i. On the basis of the direction of exchanges, the data is transmitted over a channel in three different modes, as shown in Fig. 2.1, **Simplex**, **Half-duplex** and **Full-duplex**.

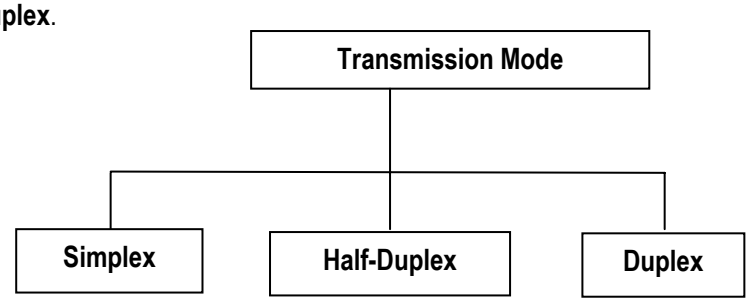


Fig. 2.1: Categories of Transmission Mode

1. **Simplex:** In Simplex communication, data transmission is always unidirectional; that is, the signal flows in one direction from any node X to any node Y, and there is no transmission or reception from Y to X. A separate transmission

channel has to be used to transmit signals from Y to X. It is like a one-way street where traffic moves only in one direction, as shown in Fig. 2.2.

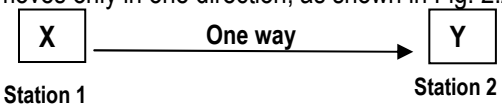


Fig. 2.2: Simplex Communication

Examples

This mode of transmission is useful in commercial transmission applications, such as radio broadcasting, television, video displays of airline arrivals and departures at airports etc. In some cases, a node could be a computer or terminal, while node Y could be an output device, such as a printer, monitor, etc.

Advantages

This mode of channel is -

- Simple, (including software)
- Inexpensive, and
- Easy to install.

Disadvantages

Simplex mode has restricted applications, for it is -

- Only a one-way communication.
- There is no possibility of sending back error or control signals to the transmitter.

2. **Half-Duplex :** In Half-Duplex communication, there are facilities to send and receive, but only one activity can be performed at a time: either send or receive. When the sender transmits, the receiver has to wait, signals flow from X to Y in one direction at a time. After Y has received the signals, it is enabled to send signals back to X at another time by switching from receiving to transmitting when X is not sending to Y. Thus, there is only one transmission medium operational at any time, as shown in Fig. 2.3. Half - Duplex mode is sometimes also known as "**Two-way-alternate" (TWA)** mode of transmission. This type of connection makes it possible to have bidirectional communications, by using the full capacity of the line.

Examples

Typical examples include walkie-talkies, internet surfing, etc.

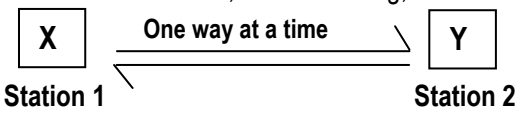


Fig. 2.3: Half-Duplex Communication

Advantages

This mode of channel -

- Helps to detect errors and request the sender to retransmit information in case of corruption of information.
- Is less costly than full duplex.

Disadvantages

- Only one device can be transmitted at a time.
- Costs more than simplex.

3. **Full-Duplex :** In Full-Duplex, data can travel in both directions simultaneously as shown in Fig. 2.4. Full-duplex transmission is like a major highway with two lanes of traffic, each lane accommodating traffic going in opposite directions. Full - duplex transmission is, in fact, two simplex connections - one connection has traffic flowing in only one direction; the other connection has traffic flowing in the opposite direction of the first connection. Each end of the line can thus transmit and receive at the same time, which means that the bandwidth is divided in two for each direction of data transmission if the same transmission medium is used for both directions of transmission.

Examples

Telephone system and mobile phones where one can hear and speak simultaneously.

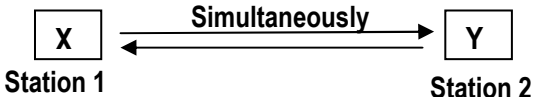


Fig. 2.4: Duplex Communication

Advantage

- It enables two-way Communication simultaneously.

Disadvantage

- It is the most expensive method in terms of equipment because two bandwidth channels are needed.

ii. On the basis of the number of bits sent simultaneously, the transmission mode is categorized into **Parallel** and **Serial Transmission,** as shown in Fig 2.5.

Serial and Parallel Transmission

The transmission mode refers to the number of elementary units of information (bits) that can be simultaneously translated by the communications channel. In fact,

processors never process a single bit at a time; generally they are able to process several, and for this reason the basic connections on a computer are parallel connections.

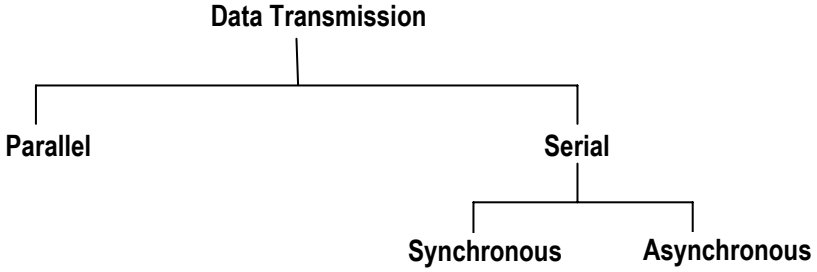


Fig. 2.5: Data Transmission Modes

1. Parallel Transmission

i. In parallel transmission, multiple bits (usually 8 bits or a byte/character) consisting of 0's and 1's are grouped and sent simultaneously on different channels (wires, frequency channels) within the same cable, or radio path, and synchronized to a clock.

ii. The concept of parallel transmission means to use 'n' channels to transmit 'n' bits at a time. These channels may be:

- N physical lines, in which case each bit is sent on a physical line (which is why parallel cables are made up of several wires in a ribbon cable).
- One physical line is divided into several sub-channels by dividing up the bandwidth. In this case, each bit is sent at a different frequency.

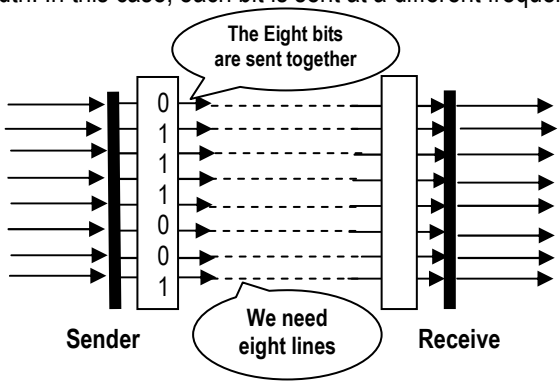


Fig.2.6. Parallel Transmission

Advantages

Speed : Parallel devices have a wider data bus than serial devices and can therefore transfer data in words of one or more bytes at a time. As a result, there is a speedup in parallel transmission bit rate over serial transmission bit rate.

Disadvantages

i. The speedup in a parallel transmission is a tradeoff versus cost since several wires cost more than a single wire, and is thus relatively expensive.
ii. Parallel transmission either takes place within a computer system (on a computer bus) or to an external device located a close distance away.

Examples

Examples of parallel mode transmission include connections between a computer and a printer (parallel printer port and cable). Most printers are within 6 meters or 20 feet of the transmitting computer and the slight cost for extra wires is offset by the added speed gained through parallel transmission of data.

2. Serial Transmission

In serial transmission, bits are sent sequentially on the same channel, so there is a need for only one communication channel rather than n channels to transmit data between two communication devices. Fig. 2.7 illustrates Serial Transmission.

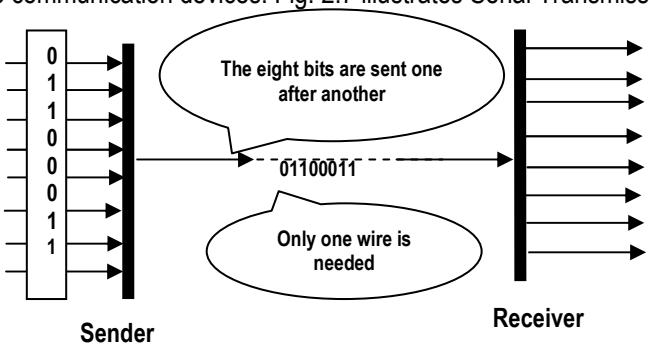


Fig. 2.7 : Serial Transmission

Advantage of Serial Transmission

Because the serial transmission needs only one communication channel, it reduces the cost of transmission over parallel by roughly a factor of 'n', where 'n' is the number of data bits.

Disadvantages of Serial Transmission

For serial transmission, some overhead time is needed since bits must be assembled and sent as a unit and then disassembled at the receiver.

iii. On the basis of the synchronization between the receiver and sender, the serial transmission can be either **Synchronous** or **Asynchronous**.

Given the problems that arise with a parallel-type connection, serial connections are normally used for data transmission. However, since a single wire transports information, the problem is how to synchronize the sender and receiver. In other words, the receiver can not necessarily distinguish the characters (or more generally the bit sequences) because the bits are sent one after the other. But Asynchronous and Synchronous types of transmission address this problem.

1. Asynchronous Transmission

Also termed as **Start-Stop communication**, an asynchronous communication technique is a technique in which the timing of a signal is unimportant and is most widely used by computers to provide connectivity to printers, modems, fax machines, etc. In other words, any communication between devices of dissimilar speeds will be of asynchronous one. For example, the communication between the computer and printer is asynchronous mode.

The basic characteristics of an Asynchronous Communication System are:

a. Sender and receiver have independent transmit and receive clocks.
b. Simple interface and inexpensive to implement.
c. Limited data rate, typically < 64 kbps.
d. Requires start and stop bits that provide byte timing.
e. Increased overhead.
f. Parity often used to validate correct reception.

How Asynchronous Transmission Takes Place

i. Asynchronous communication utilizes a transmitter, a receiver and a wire without coordination about the timing between the transmitter and the receiver. Each device uses a clock to measure the 'length' of a bit. The transmitting device transmits data whereas the receiver has to look at the incoming signal and figure out what it is receiving and coordinate and retime its clock to match the incoming signal.
ii. Without a synchronizing pulse, the receiver cannot use timing to predict when the next group will arrive. To alert the receiver to the arrival of a new group, therefore, an extra bit is added to the beginning of each byte. This bit, usually a

0, is called the **Start Bit.** To let the receiver know that the byte is finished, one or more additional bits are appended to the end of the byte. These bits, usually 1s, are called **Stop Bits**.

iii. By this method, each byte is increased in size to at least 10 bits, of which 8 are information and 2 or more are signals to the receiver. In addition, the transmission of each byte may then be followed by a gap of varying duration. This gap can either be represented by an idle channel or by a stream of additional stop bits.

Fig. 2.8 is a schematic representation of asynchronous transmission. In this example, the start bits are 0s, and the stop bits are 1s, and the gap is represented by an idle line rather than by additional stop bits.

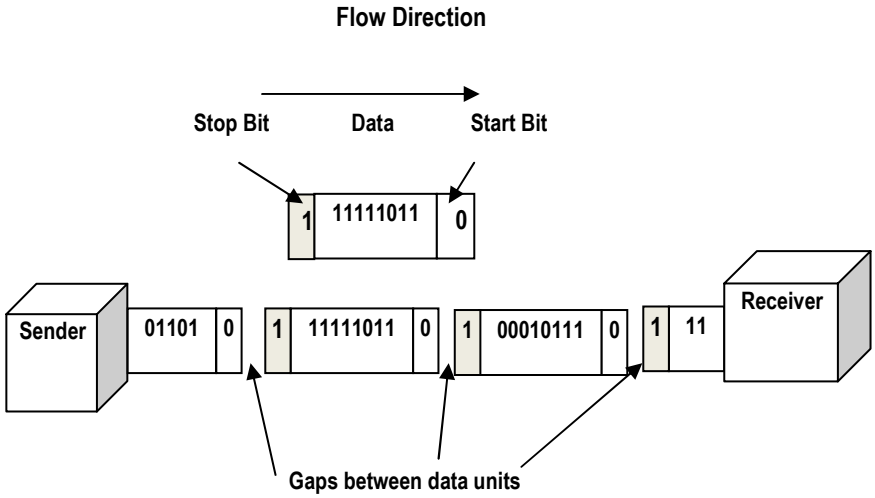


Fig. 2.8: Asynchronous Transmission

Advantages

- It is simple and does not require synchronization of the two communication sides.
- Since timing is not critical for synchronous transmission, hardware can be cheaper.
- Its set-up is fast and well suited for applications where messages are generated at irregular intervals, like data entry from the keyboard.

Disadvantages

Because of the insertion of start and stop bits into the bit stream, asynchronous

transmission is slower than other forms of transmission that operate without the addition of control information.

2. Synchronous Transmission

In Synchronous transmission, groups of bits are combined into longer "frames" which may contain multiple bytes, and those frames are sent continuously with or without data to be transmitted. In this transmission, groups of bits are sent as independent units with start/stop flags to allow for arbitrary size gaps between frames. However, in synchronous transmission, each byte is introduced onto the transmission link without a gap between it and the next one. It is left to the receiver to separate the bit stream into bytes for decoding.

In synchronous communication, the clock of the receiver is synchronised with the clock of the transmitter. On account of this, higher data transmission rates are possible with no start-stop bits. The characteristics of synchronous communication are:

a. There is synchronisation between the clocks of the transmitter & receiver.
b. It supports high data rates.
c. It is used in communication between computer and telephony networks.

How does Synchronous Transmission Take Place?

In a synchronous connection, the transmitter and receiver are paced by the same clock. The receiver continuously receives the information at the same rate at which the transmitter sends it. That is why the transmitter and receiver are paced at the same speed. In addition, supplementary information is inserted to guarantee that there are no errors during transmission.

During synchronous transmission, the bits are sent successively with no separation between each character, so it is necessary to insert synchronization elements; this is called **Character-level Synchronization**.

Fig. 2.9 shows how Synchronous Transmission takes place.

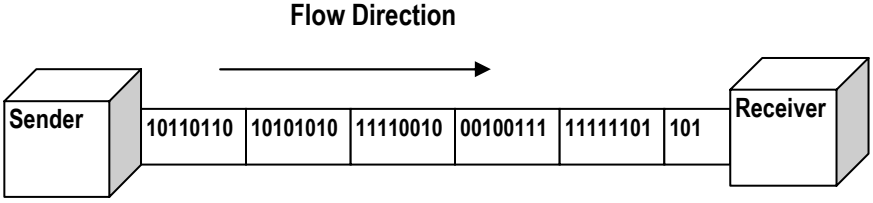


Fig. 2.9: Synchronous Transmission

Advantages

The advantage of synchronous transmission is speed. With no extra bits or gaps to introduce at the sender's end and remove at the receiver's end, it is faster than asynchronous transmission. For this reason, it is best suited for high-speed applications like the transmission of data from one computer to another.

Disadvantages

- In synchronous transmission, the data is to be recognized at the receiver's end, as there may be differences between the transmitter and receiver clocks. That is why each data transmission must be sustained long enough for the receiver to distinguish it.
- It is slightly more complex than the asynchronous one.
- Its hardware is more expensive than that of asynchronous one.

Multiplexing

Multiplexing is a set of techniques that permit the simultaneous transmission of multiple signals on a single carrier.

With increase in data-and-telecommunications usage, there is increase in traffic and also the need to accommodate individual users. To achieve this, we have to either increase individual lines each time a new channel is needed, or install higher capacity links and use each to carry multiple signals. If the transmission capacity of a link is greater than the transmission needs of the devices connected to it, the excess capacity is wasted. An efficient system maximizes the utilities of all facilities.

A device that performs multiplexing is called a multiplexer (MUX), and a device that performs the reverse process is called a demultiplexer (DEMUX), as shown in Fig. 2.10.

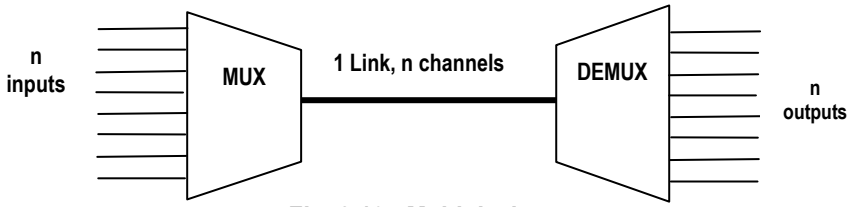
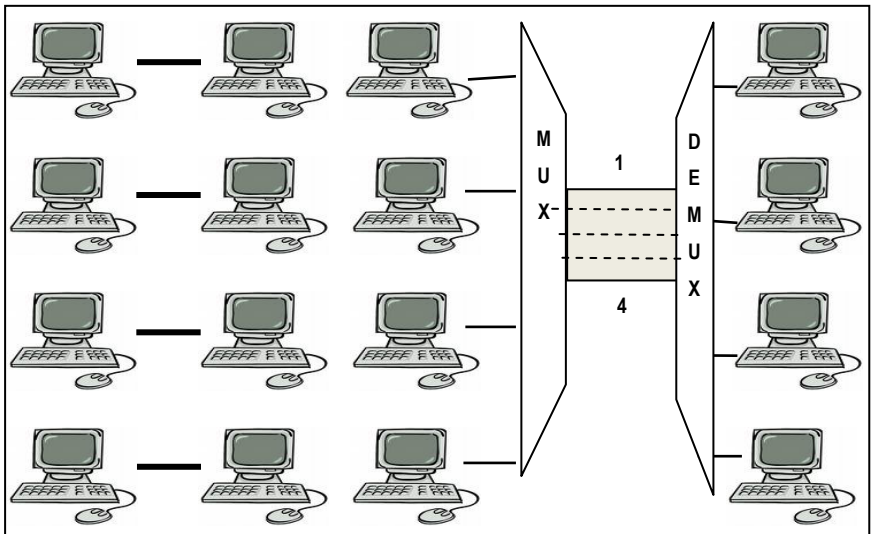


Fig. 2.10 : Multiplexing Technique

Fig. 2.11 compares a system with no multiplexing with a multiplexed system.



(a) No Multiplexing **(b) Multiplexing**

Fig. 2.11: No Multiplexing vs. Multiplexing

Signals are multiplexed by using three basic techniques, as shown below in Fig. 2.12:

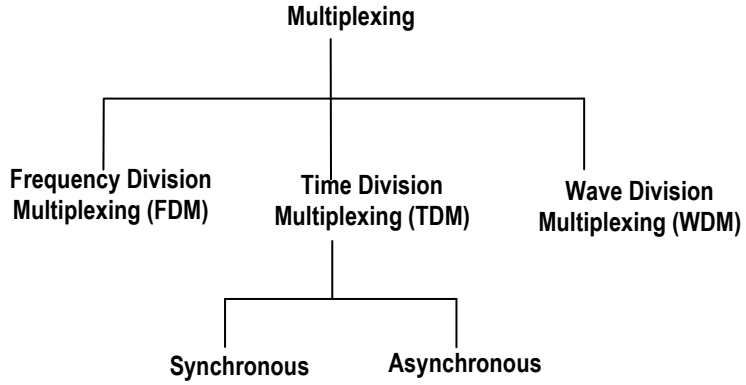


Fig. 2.12: Categories of Multiplexing

i. Frequency Division Multiplexing (FDM)

Frequency Division Multiplexing (FDM) is a form of signal multiplexing which involves assigning non-overlapping frequency ranges to different signals or to each

"user" of a medium. FDM is an analog technique that can be applied when the bandwidth of a link is greater than the combined bandwidths of the signals to be transmitted. In this scheme, signals generated by each sending device modulate different carrier frequencies, wherein these modulated signals are combined into a single composite signal that can be transported by the link. A guard-band is used to separate the channels and to ensure that they do not interfere with one another.

FDM is like people in widely separated clumps, each clump holding its own conversation at the same time, but still independent of others.

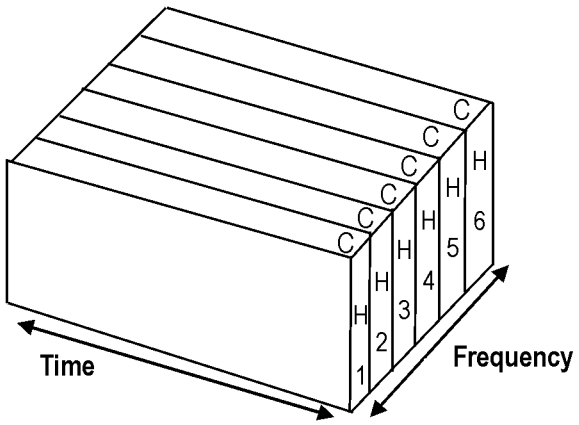


Fig. 2.13: Frequency Division Multiplexing

Fig. 2.13 shows the working of a FDM system. The transmission path is divided into three parts, each representing a channel to carry one transmission. It can be imagined as a tip where three narrow streets merge to form a three-lane highway. Each car merging onto the highway from one of the streets has its own lane and can travel without interfering with cars in other lanes.

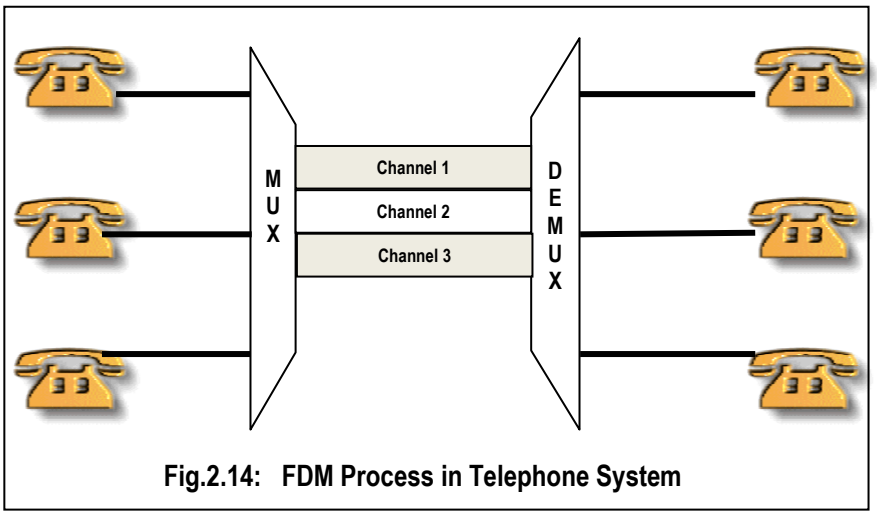


Fig.2.14: FDM Process in Telephone System

In Fig. 2.14, though the path is divided spatially into separate channels, actual channel divisions are achieved by frequency rather than by space. Typical examples of FDM are the cable television, radio broadcast and telephone system.

Characteristics of FDM

a. All signals are transmitted at the same time.
b. A multiplexer
   - accepts inputs and assigns frequencies to each device.
   - is attached to a high-speed communications line.

a. A corresponding demultiplexer
   - is at the end of the high-speed line.
   - separates the multiplexed signals into their constituent component signals.

b. The channel bandwidth is allocated even when no data is to be transmitted.

ii. Time Division Multiplexing (TDM)

Time-Division Multiplexing (**TDM**) is a type of digital or rarely analog multiplexing in which two or more signals or bit streams are transferred apparently simultaneously as sub-channels in one communication channel, but take turns on the channel. The time domain is divided into several recurrent **timeslots** of fixed length, one for each sub-channel. The users take turns in a round-robin fashion, each one periodically getting the entire bandwidth for a little burst of time. (Round robin is the simplest scheduling

algorithm which assigns time slices to each process in equal portions and order, handling all processes without priority).

For example, as shown in Fig. 2.15, a sample byte or data block of sub-channel 1 is transmitted during timeslot 1, sub-channel 2 during timeslot 2, and so on. One TDM frame consists of one timeslot per sub-channel. After the last sub-channel, the cycle starts all over again with a new frame, starting with the second sample byte or data block from sub-channel 1, and so on.

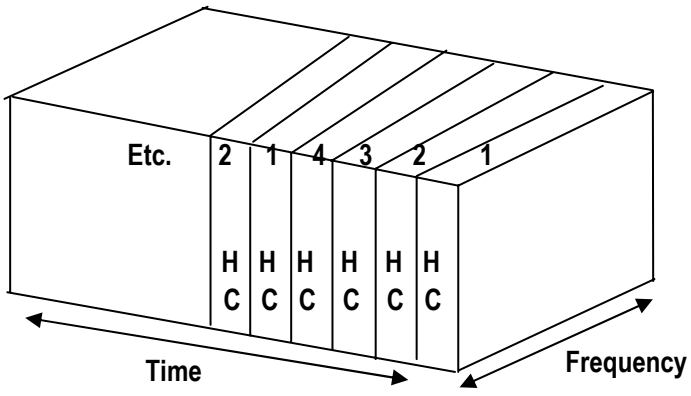


Fig. 2.15: Time Division Multiplexing

TDM is applied when the data rate capacity of the transmission medium is greater than the data rate required by the sending and receiving devices.

In TDM, time forms the basis for multiplexing. This is a technique where a short time sample is allotted to each of the users who wish to use the channel. Each user is sampled in turn on the basis of time, and then the sequence is repeated. Fig. 2.16 shows how TDM operates. One typical example of TDM is a traffic signal.

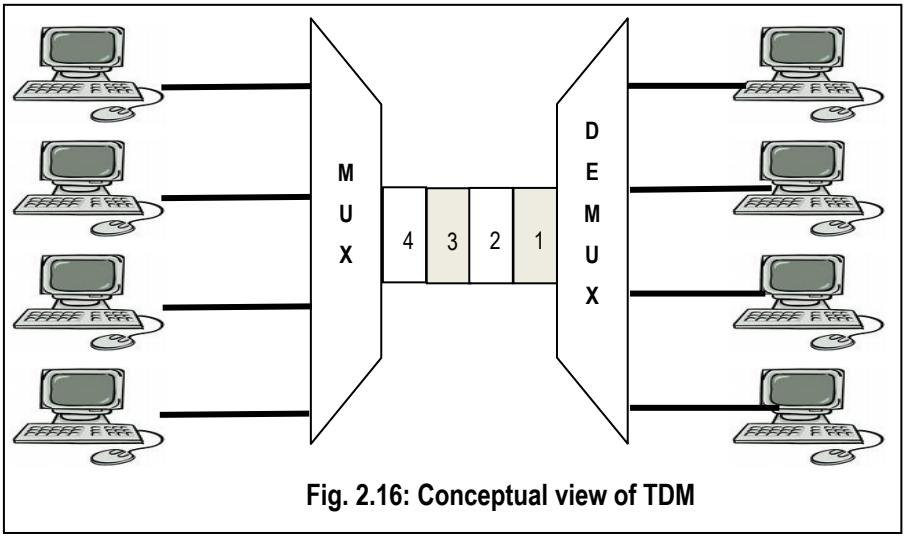


Fig. 2.16: Conceptual view of TDM

In Fig. 2.16, portions of signals 1, 2, 3 and 4 occupy the link sequentially. The concept can be compared to a ski-lift that serves several runs wherein each run has its own line and the skiers in each line take turns getting on the lift. As each chair reaches the top of the mountain, the skier riding it gets off and skis down the run for which he or she waited in a queue.

TDM can be implemented in two ways: **Synchronous TDM** and **Asynchronous TDM**.

1. Synchronous TDM

In this scheme, the multiplexer allocates exactly the same time slot to each device at all times, no matter whether a device has anything to transmit or not. For example, time slot X is assigned to device X alone and cannot be used by any other device. Each time the device's allocated time slot comes up, that particular device has the option to send a segment of its data. If a device is unable to transmit or does not have any data to transmit, its time slot remains vacant.

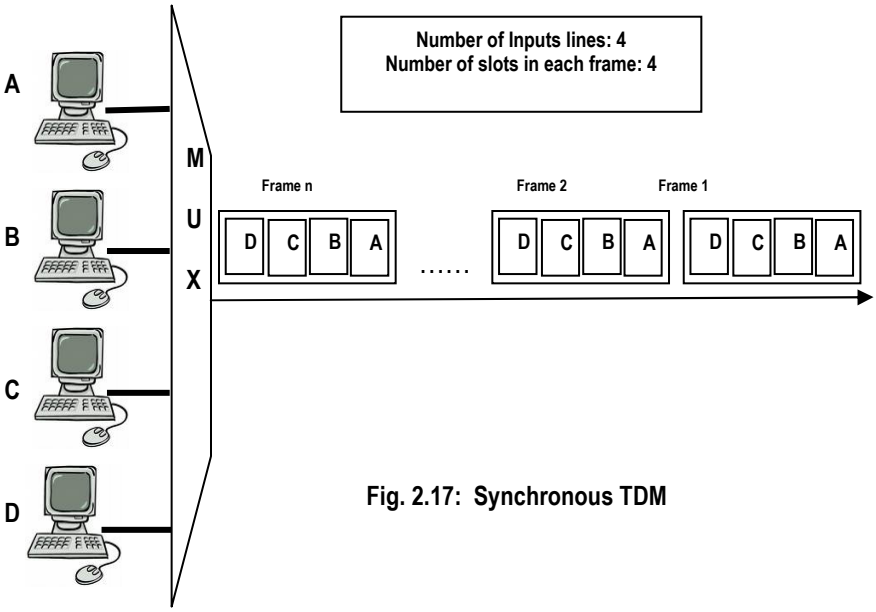


Fig. 2.17: Synchronous TDM

Time slots are grouped into frames wherein a frame consists of one complete cycle of time slots dedicated to each sender. Thus the number of slots "n" in a frame is equal to the number of input lines "n" carrying data through "n" sending devices.

In Fig. 2.17, four input devices A, B, C and D are sending signals. These signals are multiplexed onto a single path using synchronous TDM. In this example, all the inputs have the same data rate, so the number of time slots in each frame is equal to the number of input lines.

2. Asynchronous TDM or Statistical Time-Division Multiplexing

The disadvantage of Synchronous TDM is that it does not exploit the full capacity of a link, as the time slots are allotted to each of the sending device irrespective of whether it has any data to send or not. Because the time slots are pre-assigned and fixed, a connected device that is not transmitting will lead to an empty slot and a wasted path.

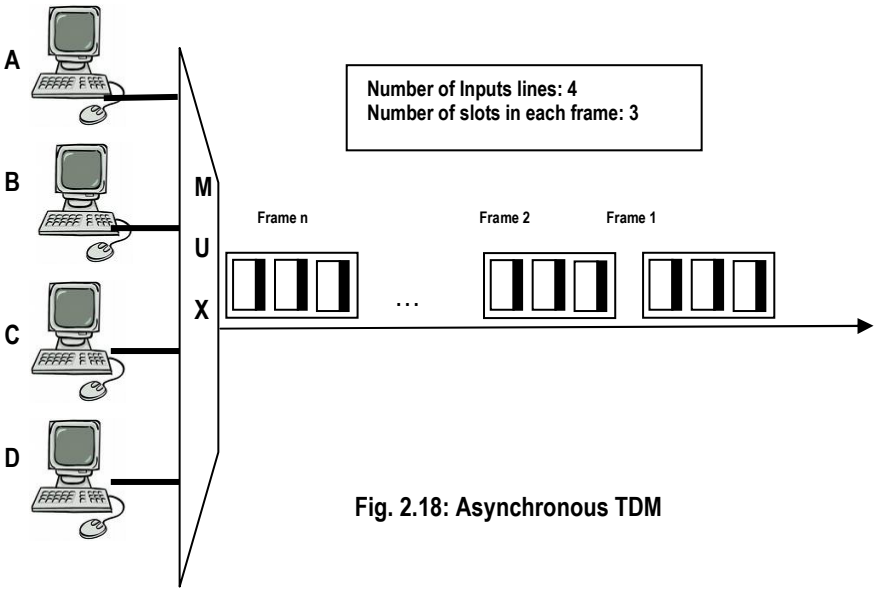


Fig. 2.18: Asynchronous TDM

Asynchronous TDM avoids this waste and is more flexible. Under this scheme, the length of time allocated is not fixed for each device, and time is given to only those devices that have data to transmit.

Unlike synchronous system, wherein if we have n input lines the frame contains a fixed number of at least n time slots, an asynchronous system having n input lines has a frame containing no more than m slots, with m less than n. The number of time slots in this scheme is based on a statistical analysis of the number of input lines that are likely to transmit at any given time. Also, more than one slot in a frame can be allocated for an input device. Fig. 2.18 is a schematic representation of Asynchronous TDM.

iii. Wavelength Division Multiplexing (WDM)

Wavelength Division Multiplexing (WDM) is conceptually like the FDM, which multiplexes multiple optical carrier signals on a single optical fibre by using different wavelengths (colours) of laser light to carry different signals.

Though the WDM scheme looks highly complex, it is actually simple. Multiple light sources are combined into one single light at the multiplexer and do the reverse at the demultiplexer. Combining and splitting of light sources are easily handled by a prism. As illustrated in Fig. 2.19, the technique is based on the fact that a laser can be designed to emit monochromatic light. Each signal to be transmitted is attached to

a laser that emits a different colour light beam; all the light beams are sent at the same time and a receiving device splits the colours into the original signals again.

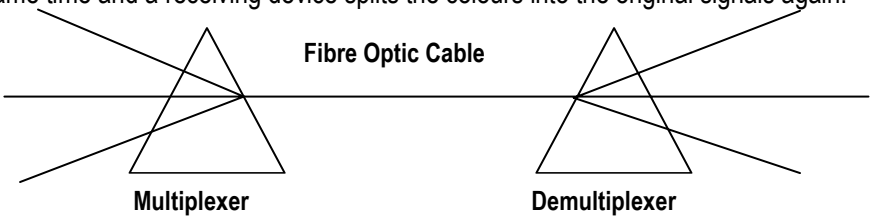


Fig. 2.19: Wavelength Division Multiplexing

Switching Techniques

Switching Techniques refer to the manner in which a communication path is established between the sender and the receiver. To connect multiple devices, one way is to install a point-to-point connection between each pair of devices (a mesh topology) or between a central device and every other device (a star topology). These methods are impractical and wasteful when applied to very large networks. A better solution to this problem is switching in which a series of interlinked nodes, called switches, are used. Switches are hardware and/or software devices capable of creating temporary connections between two or more devices linked to the switch but not to each other.

Fig. 2.20 shows the various switching techniques that are available these days:

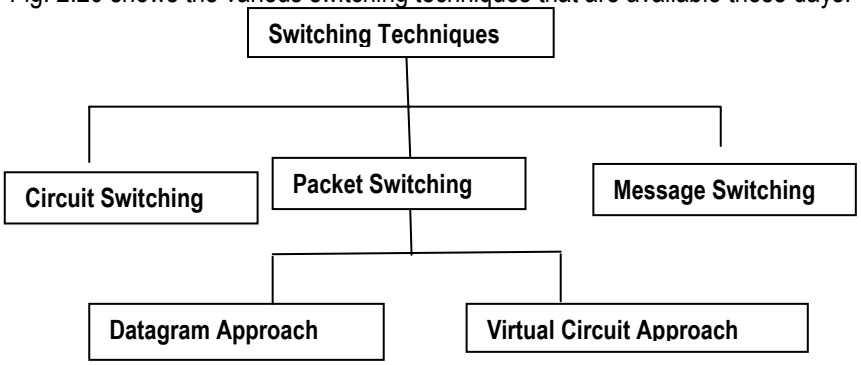


Fig. 2.20: Categories of Switching Techniques

i. Circuit Switching

Circuit Switching is a type of communication in which a permanent physical

connection is established between two devices, such as computers or phones, for the entire duration of transmission. In circuit switching, the entire bandwidth of the circuit is available to the communicating parties. The cost of the circuit is calculated on the basis of the time used. Circuit switching networks are ideal for real-time data transmissions, and are sometimes called **Connection-oriented networks**.

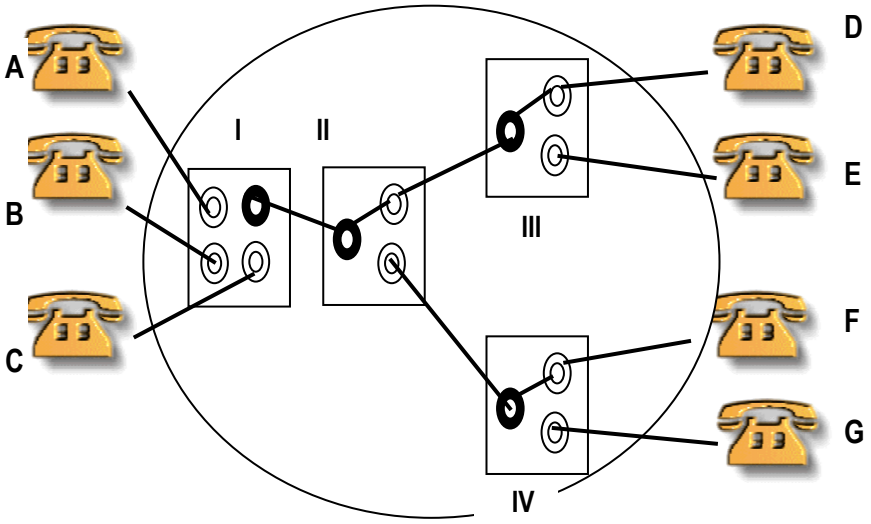


Fig. 2.21: Circuit Switched Network

One typical illustration of circuit switching is the telephone system shown in Fig. 2.21. When we dial a number, our call lands at the telephone exchange where our phone is attached. The telephone exchange acts like a big switch connecting the telephone exchange of the called number. From that exchange, a connection to the called number is established. The connection remains established until the communication parties finish the communication.

Limitations of Circuit Switching

1. Circuit switching was designed for voice communication. So it is less suited for data and non-voice transmissions.
2. In circuit switching, the link creates the equivalent of a single cable between two devices and thereby assumes a single data rate for both devices, which limits the usefulness of connection for networks interconnecting a variety of digital devices.

3. Once a circuit is established, all parts of the transmission take place through that path only, no matter whether it is efficient or available. So it is inflexible.
4. Priorities cannot be setup in data transmission as circuit switching assigns equal priority to all the transmissions.

ii. Packet Switching

Packet Switching is a better solution for data transmission as compared to circuit switched network. In this data are transmitted in **Packets**, which are defined as discrete units of data with potentially variable length blocks. The maximum length of a packet is established by the network. Longer transmissions are broken into multiple packets, and each packet contains not only data but also a header with control information (like source and destination address). The packets are routed over the network node to node, with each node storing the packet briefly and routing it again according to the information in its header. Each packet has three important portions:

- **Header:** It holds information about the data contained in the packet. The contents of a header are:
    - Length of a packet (some networks have fixed-length packets, while others rely on the header to contain this information).
    - Packet number (which packet is in a sequence of packets).
    - Destination address (where the packet is going).
    - Originating address (where the packet came from).
    - Protocol information.
    - Time to live.
- **Body:** It is the actual data contained in the packet and delivered to the destination.
- **Footer:** It is a portion of the packet that holds control information about the packet for error checking. The source computes a checksum of the body of the packet and appends the same in the footer. The receiving device again computes the checksum. If the values match, the content of the packet is not tampered with. If the values do not match, the receiver determines that there has been an integrity violation and thus sends a request to the originating device to resend the packet.

Table 2.1 compares Circuit-switched Network to Packet-Switched Network.

ISSUE	CIRCUIT-SWITCHED	PACKET-SWITCHED
Dedicated "copper" path	Yes	No
Bandwidth available	Fixed	Dynamic
Potentially wasted bandwidth	Yes	No

Store-and-forward transmission	No	Yes
Same route followed by each packet	Yes	No
Call setup	Required	Not required
When can congestion occur	At setup time	On every packet
Costing	Per minute	Per packet

Table 2.1: Circuit-Switched vs Packet-Switched Network

There are two basic approaches to Packet Switching: **Datagram Approach** and **Virtual Circuit Approach**.

1. Datagram Approach

In this approach, each packet is treated as an independent entity, and its header contains full information about its destination. The intermediate nodes examine the header of the packet, and decide to which node to send the packet so that it reaches its destination. Here, packets don't follow a pre-established route and take different routes to the destination; because of this, delivery is not guaranteed. Moreover, packets arrive at the destination in a different order. The destination node then takes up the process of arranging the packets in order.

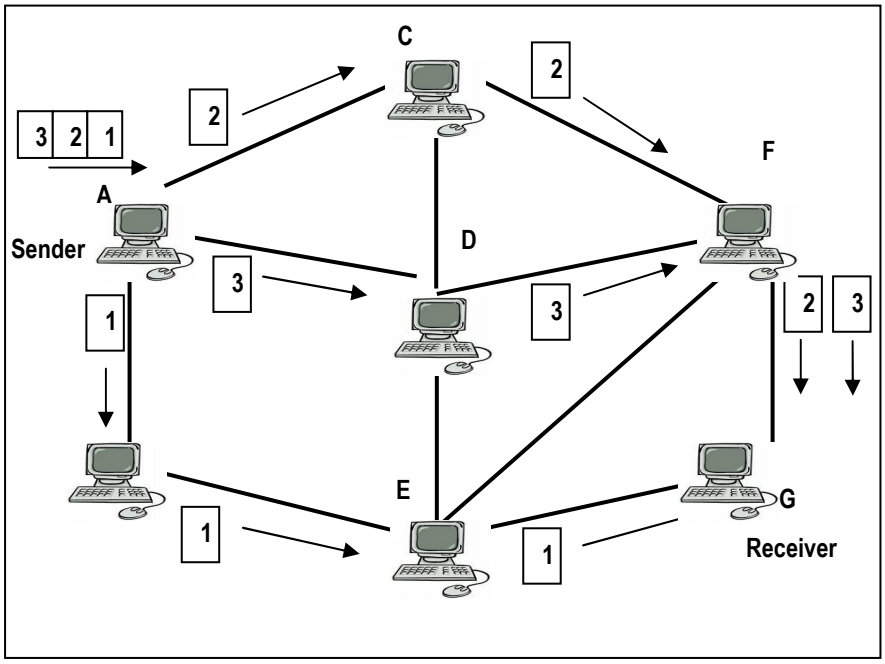


Fig. 2.22 illustrates how a datagram works.

When node A wants to send packets to node G, all the packets are pushed into the network. Each packet takes its own path to reach the destination. In the illustration, packet 1 takes the path A - B - E - G, packet 2 takes the path A - C - F - G and packet 3 takes the path A - D - F - G. Node G performs the process of re-arranging the packets in order. The working of the Datagram depends on the Internet, which uses the IP network protocol.

2. Virtual Circuit

In this scheme, an initial set-up phase is used to create a route between the source and the destination. All packets thus follow the route that has been set-up. The packets here have short headers, containing only a virtual circuit identifier (VCI), and not their destination. Each intermediate node passes the packets according to the information stored in it during the set-up phase. In this way, packets arrive at the destination in a correct sequence, and it is guaranteed that there are no errors. If an intermediate node fails, all virtual circuits that pass through it are lost.

This is illustrated in Fig 2.23, When node A wants to send packets to node G, it first establishes a path A - C - D - F - G. All packets thus follow this path only. The most common forms of Virtual Circuit networks are X.25 and Frame Relay, which are commonly used for public data networks (PDN).

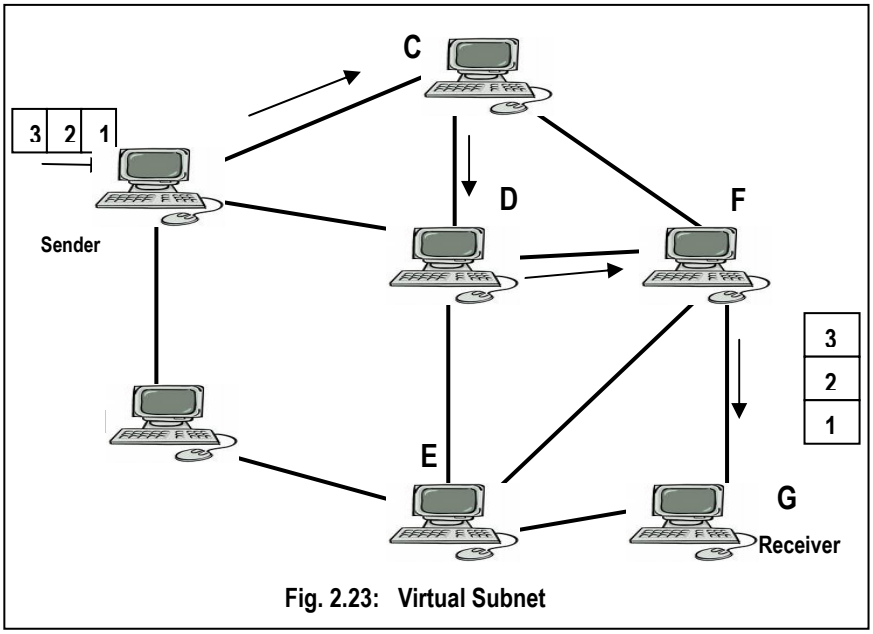


Fig. 2.23: Virtual Subnet

Issue	Datagram Subnet	Virtual Circuit Subnet
Circuit setup	Net required.	Required.
Addressing	Each packet contains the complete source and destination address.	Each packet contains a short VC number only.
State Information	Subnet does not hold state information.	Each VC requires subnet table space.
Routing	Each packet is routed independently.	All packets follow the route chosen when VC is set.
Impact of Router Failures	None, except for packets lost during the crash.	All VCs passed through the failed router are terminated.
Congestion Control	Difficult.	Easy because buffers can be allocated in advance for each VC.

Table 2.2: Datagram Subnet vs Virtual Circuit

Table 2.2 compares Datagram subnet to Virtual Circuit.

iii. Message Switching

In Message Switching, also called **Store-and-Forward communication**, no physical path is established in advance between the sender and the receiver. Instead, when the sender sends a block of data, it is stored in the switching point, and when the appropriate route is available, it is transferred to the next switching point, one hop at a time, until the message reaches its destination. Each block is received in its entirety, inspected for errors, and then retransmitted.

Store and Forward is considered a switching technique as there is no direct link between the sending and receiving devices. A message is delivered to the node along one path and then rerouted along another to its destination.

The primary use of message switching was to provide high-level network services like delayed delivery, broadcast, etc. for unintelligent devices. Since such devices have been replaced, this type of switch has virtually disappeared. Also, the delays in the process, as well as the requirement of large-capacity storage media at each node, make it unsuitable for direct communication.

Fig. 2.24 shows how message switching works. Its cost is determined by the length of the message being transmitted.

As indicated, the complete message is sent from node A to node Q when the link interconnecting them becomes available. The main problem encountered in message switching is Queuing Delay. This is on account of waiting for the link to become available. The message is stored at storage point SP 1 until the next link becomes available, with another queuing delay before it can be forwarded. It repeats this process until it reaches its destination.

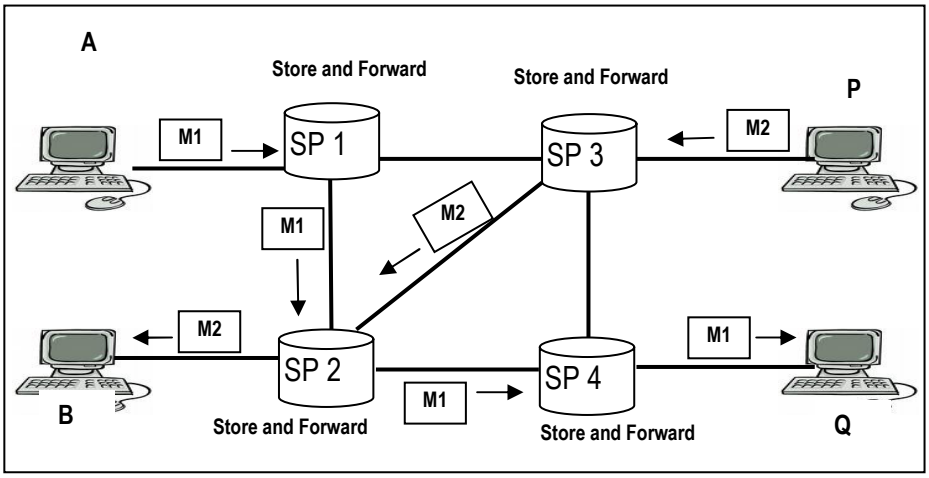


Fig. 2.24: Message Switching

MODEM

MODEM (Modulator-Demodulator) is a device that modulates an analog carrier signal to encode digital information, and also demodulates such a carrier signal to decode the transmitted information. The goal is to produce a signal that can be transmitted easily and decoded to reproduce the original digital data. A modem connects a computer to a standard telephone line so that data can be transmitted and received electronically.

How do Modems Work?

Computers work in the digital mode while telephones work in the analog mode. The major task of a modem is to convert digital signals into analog signals at the transmitting end and convert analog signals into digital signals at the receiving end.

Types of Modems

1. **External Modem:** This is the most commonly used modem, for it is simple to install and operate. These modems have their own power supply and are

connected to the serial port of the computer. External modems have indicators for the following:

i. Transmit Data (Tx)
ii. Receive Data (Rx)
iii. Acknowledge (ACK)
iv. Signal Ground
v. Data Terminal Ready (DTR)
vi. Carrier Detect (CD)
vii. Request to Send (RTS)
viii. Clear to Send (CTS)

Did you know?

The modem that we use for accessing our Broadband from home is generally the ADSL Modem: i.e., Asynchronous Digital Subscriber Line Modem. The ADSL Modem splits the traffic between channels over regular telephone lines and can carry signals for up to 5000 meters.

2. **Internal Modem:** This is directly plugged into the mother-board of the computer. It takes power from the computer's power-supply unit and operates like the external modem.

3. **PC Card Modem:** This modem, designed for portable computers, is of the size of a credit card and fits into the PC Card slot on notebook.

4. **Wireless Modem:** Some Internet Service Providers support wireless internet services, for which wireless modems are used. These modems work like the traditional wired modems, but differ from them in their structure.

5. **Cable Modem:** The cable modem uses coaxial cable television lines to provide a greater bandwidth than the dial-up computer modem. But this transmission rate fluctuates with the number of users because of the shared bandwidth on which the cable technology is based.

6. **DSL Modem:** DSL (Digital Subscriber Line) modem is exclusively used for connections from a telephone switching office to the user. It is of two categories:

   - **ADSL** or Asymmetric Digital Subscriber Line is a form of DSL, a data communication technology that enables faster data transmission through copper telephone lines than a conventional voice-band modem.
   - **SDSL** or Symmetric Digital Subscriber Line, is a collection of Internet access technologies based on DSL that offer symmetric bandwidth upstream and downstream.

   SDSL is considered the opposite of ADSL technologies where the upstream bandwidth is lower than the downstream bandwidth.

7. **GPRS Modem:** The GPRS (General Packet Radio Signals) modem is used to browse internet and for other communications using the GPRS services. The

GPRS service is provided on the cellular networks and is costlier than other communication services.

Network Categories

A computer network consists of autonomous machines called nodes, and connected in some configuration. The purpose of networking is to communicate information and share resources (both hardware and software). Table 2.3 presents the broad categorization of computer networks on different lines.

Category	Description
Message Capacity	Network classification under this category describes whether the network is a baseband network or broadband network.
Range	Refers to the geographical area of operation like Local Area Networks (LANs), Metropolitan Area Networks (MANs), Wide Area Networks (WANs), Home Networks and Internetworks.
Node Relationships	Networks are also categorized on the basis of relationship among the nodes. Networks categorized along these lines are peer-to-peer, server-based, and client/server.
Topology	Topology refers to both the network's logical topology (logical layout of nodes in the network) and physical topology (physical layout) and include Mesh, Bus, Star, Ring, and Tree.

Table 2.3: Categories of Networks

i. Categories on the basis of Message Capacity

- **Baseband Network :** A baseband network transmits one message at a time. Most LANs are baseband networks.
- **Broadband Network :** A broadband network transmits more than one message at a time by using different frequencies for each message.

ii. Categories on the basis of Range

Based on range, networks are classified into:

- Local Area Networks (LANs)
- Metropolitan Area Networks (MANs)
- Wide Area Networks (WANs)
- Wireless LANs

- Home Networks
- Internetworks

Local Area Networks (LAN)

LAN is a network that is limited in size and generally situated within a single building or campus whose area is spread over a few kilometers. Some of the characteristics of a LAN are:

- Uses regular topology.
- High data transmission speeds.
- Permanently connected.
- Security is high.
- Installation and maintenance which can be centralized or distributed is relatively easy.
- Low error rates.
- Owned and maintained by the organization.
- Majority of the installations uses guided media like UTP, Optical fibres etc.,

Metropolitan Area Network (MAN)

In MAN, the area covered is more than LAN, but less than WAN. Unlike LANs, MANs generally include provisions for both voice and data transmissions. A typical example of a MAN is the cable television network. Initially used only for cable television, this has been extended to provide Internet services. Most MAN networks use either of the two network architectures:

- **FDDI (Fiber Distributed Data Interface)**, which uses optical fiber as the basic transmission medium and supports transmission speeds of 100-plus Mbps.
- **DQDB (Distributed Queue Dual Bus),** a two-bus topology, which is specified in IEEE 802.6. DQDB supports transmission speeds ranging from 50 to 600 Mbps over distances as large as 50 kilometers.

Wide Area Network (WAN)

A WAN is a network that covers a large area typically countries or continents. These are used to interconnect LANs over long distances. The following are the characteristics of a WAN:

- It usually has an irregular topology.
- Compared to LAN, its transmission speed is slow.
- It can be connected on demand or be permanently connected.
- Most WANs are not owned by any one organisation; they work under collective or distributed ownership and management.

- WAN uses public or private networks.
- Routers are used to connect LANs to a WAN.

WANs are connected in three ways: Circuit Switching (ISDN, Switched 56, and Switched T1), Message switching (ATM (Asynchronous Transfer Mode), Frame Relay and X.25) and leased lines.

Wireless LANs

Networks that are established by using digital wireless communications are generally called Wireless LANs. In these, short-range radio is used as the medium of communication. One such short-range wireless network is the Bluetooth technology that connects keyboards, printers, digital cameras, headsets, scanners, and other devices to a computer.

In wireless LANs, every computer has a radio modem and antenna with which it can communicate with other systems. Wireless LANs are also implemented in wide area systems. The first generation wireless network was analog and used for voice only. The second generation was digital but used for voice only. The 3G wireless networks cater to both voice and data.

Home Networks

Home Networks, a recent new development, is based on the assumption that in the future almost all devices in every home will be networked. Every device will be capable of talking to every other device, and all of them will be accessible over the Internet.

Some of the devices that are likely to be networked and accessible by Internet include desktop PCs, notebook PCs, PDAs, shared peripherals, TVs, DVDs, camcorders, cameras, MP3 players, telephones (both landline and mobile), intercom devices, fax machines, microwave cooking ranges, refrigerators, clocks, lighting devices, smoke/burglar alarm, etc.

Internet works

An internet (lowercase "i") is a collection of separate physical networks, interconnected by a common protocol, to form a single logical network. The Internet (uppercase "I") is the worldwide collection of interconnected networks that use Internet Protocol to link the various physical networks into a single network.

iii. Categories on the basis of Node Relationships

- **Peer - to - Peer Architecture**: In this every node can act as a client and server; that is, all nodes are treated on par.
- **Server Based Architecture**: In this there is a dedicated file server, that runs the network, granting other nodes access to resources. Novell's NetWare is a classic example of Server-based architecture.

- **Client-Server Architecture**: In this the job is shared between the server and the nodes (client). The client queries the server and the server responds.

iv. Categories on the basis of Topology

Network Topology

In communication networks, a topology is usually a schematic description of the arrangement of a network, including its nodes and connecting lines. Before any network topology is designed, the network design engineer considers the following goals:

- To provide maximum possible reliability by providing alternative routes if a node fails.
- To be able to pinpoint faults promptly.
- To route network traffic through the least cost path within the network. This is usually provided in two ways:
  i. By minimising the actual length of the channel between its components; it entails routing the traffic through very few intermediate components.
  ii. By providing the least expensive channel option for a particular application; for instance, transmitting low priority data over low cost dial up lines, in contrast to using an expensive high speed satellite channel.
- To give the end users the best possible response time and throughput. This is especially important for interactive sessions between user applications.

Fig. 2.25 displays the hierarchical classification of Network Topologies, which are: **Physical Topologies** and **Logical Topologies**.

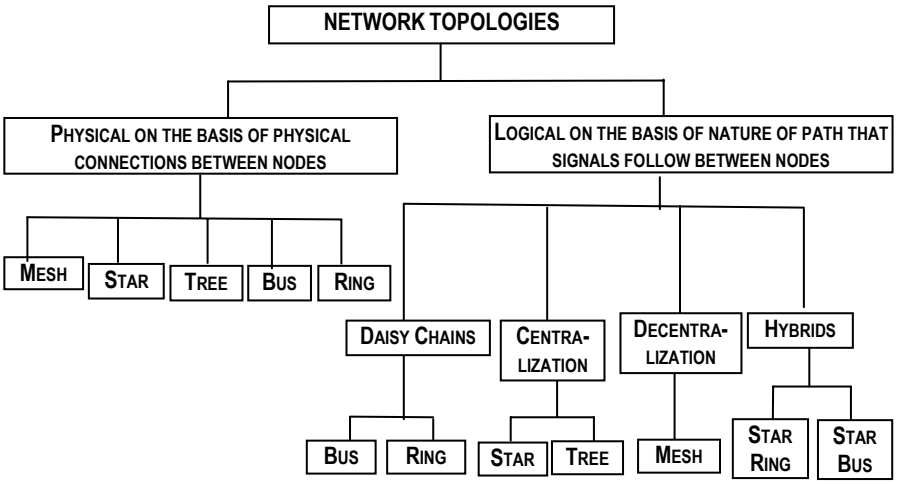


Fig. 2.25: Network Topologies Chart

i. Physical Topology

Physical topology of a network is the actual geometric representation of the relationship of all the links and linking devices. There are several common physical topologies, described below and also shown in the illustration.

a. Mesh Topology

In this topology, every node is physically connected to every other node. This is generally used in systems which require a high degree of fault tolerance, such as the backbone of a telecommunications company or an ISP. Its primary advantage is that it is highly fault tolerant: when one node fails, traffic can easily be diverted to other nodes. It is also not vulnerable to bottlenecks. Fig. 2.26 shows mesh topology.

Disadvantages

- It requires more cables for connecting devices than any other topology.
- It is complex and difficult to set up and maintain. If there are 'n' nodes in the system, the total number of connections emanating from every node is (n-1) and the total number of connections in the system is (n*(n-1)/2).
- It is difficult to introduce or remove nodes from the system as it necessitates rewiring.
- Its maintenance is expensive.

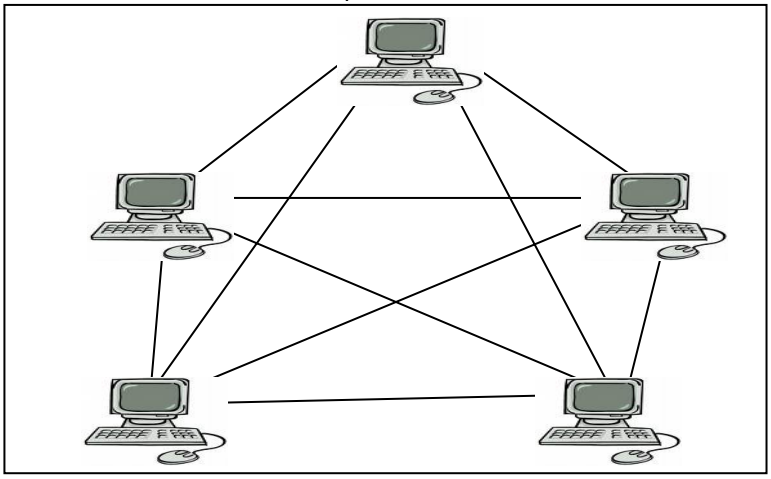


Fig.2.26: Mesh Topology

b. Star Topology

Earlier called **ARCNET (Attached Resource Computer Network)**, star topology contains a central hub to which each and every node is connected. This necessitates drawing of a separate cable from each and every node to the central hub. All inter-node data transmission has to pass through it. Fig. 2.27 shows the star topology.

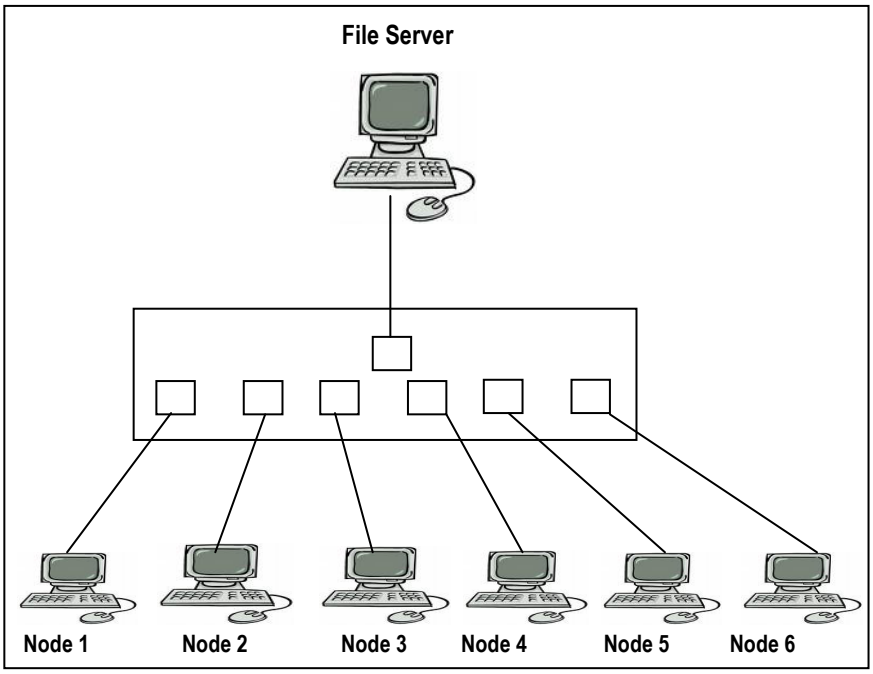


Fig. 2.27: Star Topology

Advantages

- Easy to troubleshoot.
- Allows mixing of cabling types.
- Easy to install and wire.
- No disruptions to the network when nodes are down.
- No disruptions to the network while connecting or removing devices.

Disadvantages

- Hubs become a single point of failure.
- Cabling is expensive because individual cables have to be drawn from nodes to hubs.

- More expensive than bus topology because of the high cost of hubs.
- The capacity of the hub determines the number of nodes that can be connected.

c. Tree Topology

A tree topology, also called "Expandable Star Topology", consists of groups of star -configured machines connected to one another by hubs. These are of two types : **Active** and **Passive.**

Active Hubs need electric power and have the ability to drive other hubs and nodes. Passive hubs cannot drive hubs and are used to connect machines.

The connection between Active hubs and Passive hubs is permitted, whereas the connections between Passive and Active hub and Passive to Passive hub are not permitted. Fig. 2.28 shows a tree topology with Active and Passive hubs.

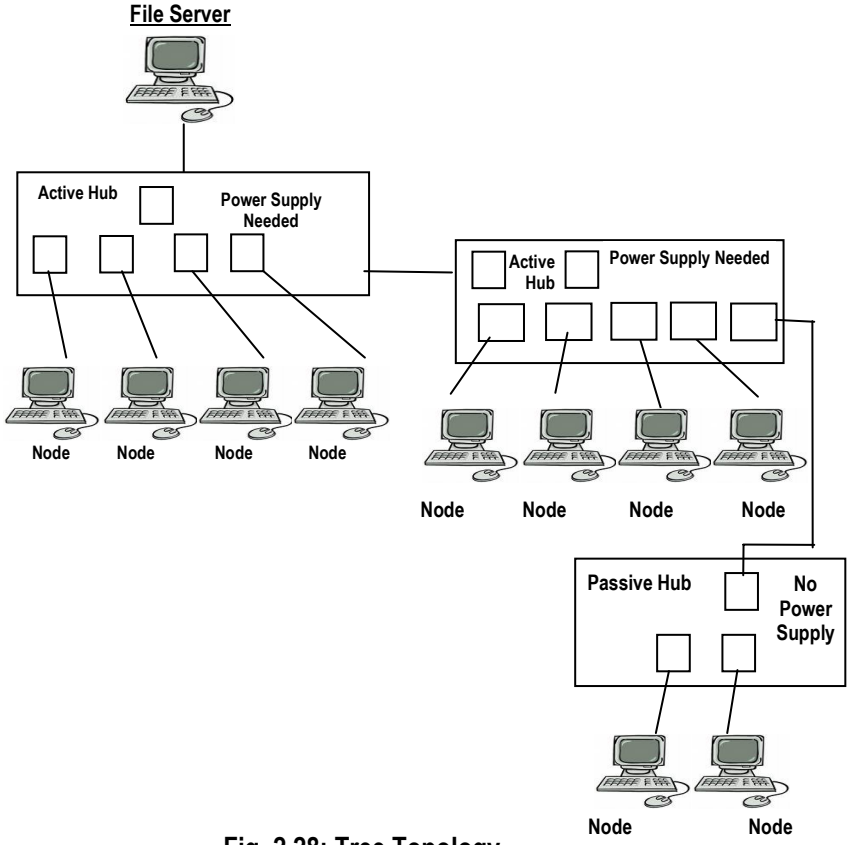


Fig. 2.28: Tree Topology

Advantages

- Point-to-point wiring for individual segments.
- Supported by several hardware and software vendors.

Disadvantages

- If the backbone line breaks, the entire segment goes down.
- The overall length of each segment is limited by the type of cables used.
- More difficult to configure and wire than other topologies.

d. Bus topology

In Bus topology, a single cable also called the backbone, runs through the entire network connecting all the workstations, servers, printers and other devices on the network. The cable runs from device to device by using "tee" connectors that plug into the network adapter cards. A device wishing to communicate with another device on the network sends a broadcast message onto the wire that all other devices see, but only the intended recipient actually accepts and processes the message.

Fig. 2.29 shows the picture of a bus topology.

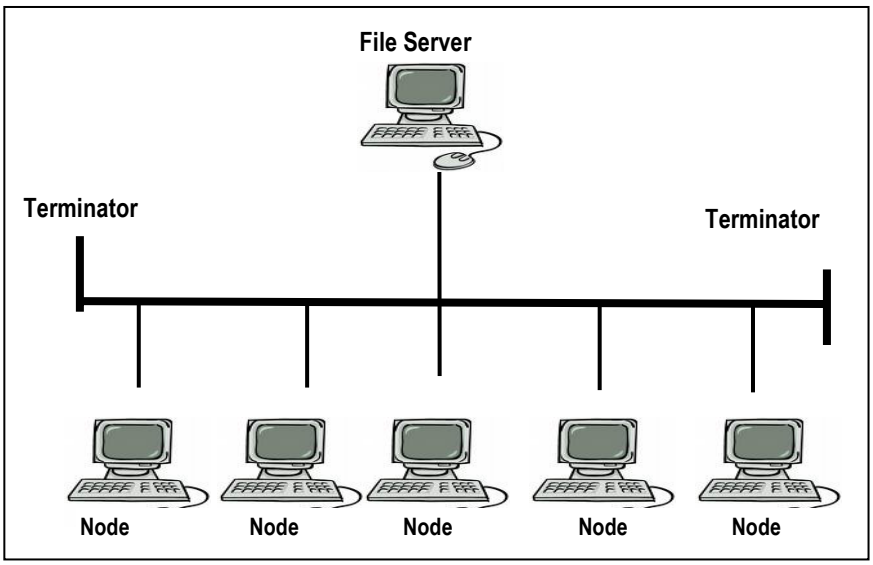


Fig. 2.29: Bus Topology

Advantages

- Less expensive when compared to star topology due to less cabling and no network hubs.

- Good for smaller networks not requiring higher speeds.
- Networks can be extended by the use of repeaters.
- Easy to install.

Disadvantages

- Limited in size and speed.
- One bad connector or failure of the backbone cable shuts down entire network.
- Difficult to troubleshoot.
- Addition of nodes negatively affects the performance of the whole network, and if there is a lot of traffic throughput decreases rapidly.

e. Ring Topology

In a ring network, every device has exactly two neighbours for communication purposes. All messages travel through a ring in the same direction (effectively either "clockwise" or "anti-clockwise"). A token, or small data packet, is continuously passed around the network. Whenever a device needs to transmit, it holds the token. Whoever holds the token has the right to communicate.

Token ring networks have the physical cabling of a star topology and the logical function of a ring through the use of multi access units (MAU). In a ring topology, the network signal is passed through each network card of each device and passed on to the next device. All devices have individual cables to the MAU. The MAU makes a logical ring connection between the devices internally. Fig. 2.30 shows the ring topology.

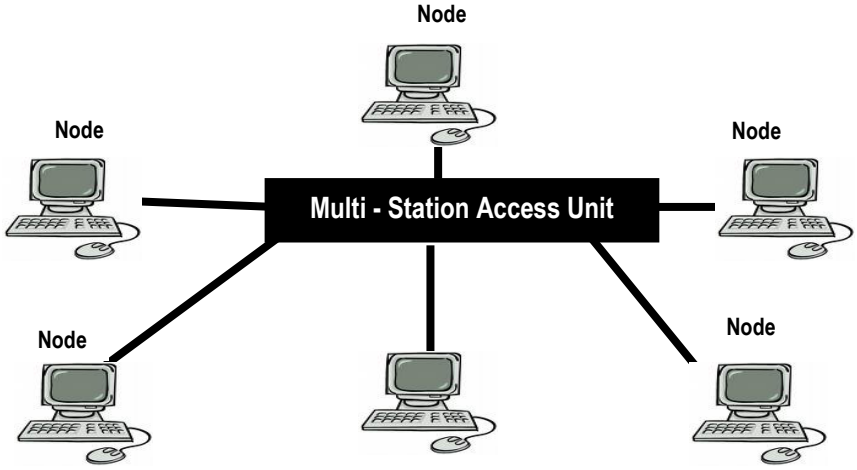


Fig. 2.30 : Ring Topology

Advantages

- Every device gets an opportunity to transmit.
- Performs better than the star topology under heavy network load.

Disadvantages

- One malfunctioning workstation or bad port in the MAU can create problems for the entire network.
- Moves, additions and change of devices can affect the network.
- Network adapter cards and MAU's are much more expensive than Ethernet cards and hubs.

ii. Logical Topology

Logical (or signal) topology refers to the nature of the paths the signals follow from node to node. For example, some networks are physically laid out in a star configuration, but they operate logically as bus or ring networks.

a. Daisy Chains: Daisy chaining is a process to add more computers to a network, or connecting each computer in a series to the next. If a message is intended for a computer partway down the line, each system bounces it along in sequence until it reaches the destination. A daisy - chained network can take two basic forms: **Linear** and **Ring**.

- **Linear Topology:** This puts a two-way link between two computers. This was expensive in the early days of computing, since each computer (except for the ones at each end as for them there will be only sender/receiver.) required two receivers and two transmitters.
- **Ring Topology:** This connects the computers at each end, and being unidirectional, reduces the number of transmitters and receivers. When a node sends a message, the message is processed by each computer in the ring. If a computer is not the destination node, it passes the message to the next node, until the message arrives at its destination.

b. Centralization

- **Star topology:** The star topology reduces the probability of a network failure by connecting all the peripheral nodes (computers, etc.) to a central node. When the physical star topology is applied to a logical bus network such as Ethernet, this central node (traditionally a hub) rebroadcasts all transmissions received from any peripheral node to all peripheral nodes on the network, sometimes including the originating node. All the peripheral nodes may thus communicate with all others by transmitting to, and

receiving from, the central node only. The failure of a transmission line linking any peripheral node to the central node will result in the isolation of that node from all others, but the remaining peripheral nodes will be unaffected. However, the disadvantage is that the failure of the central node will cause the failure of all the peripheral nodes too.

- **Tree Topology:** A tree topology can be viewed as a collection of star networks arranged in a hierarchy. This tree has individual peripheral nodes which transmit to and receive from one other node only and are not required to act as repeaters or regenerators. Unlike the star network, the functionality of the central node may be distributed.

c. Decentralization

- **Mesh Topology:** In a mesh topology, there are at least two nodes with two or more paths between them to provide redundant paths to be used in case the link providing one of the paths fails. This decentralization is often used to advantage to compensate for the single-point-failure disadvantage that is present when using a single device as a central node (e.g., in star and tree networks). The number of arbitrary forks in mesh networks makes them more difficult to design and implement, but their decentralized nature makes them very useful.

d. Hybrids

Hybrid networks use a combination of any two or more topologies in such a way that the resulting network does not exhibit any one of the standard topologies (e.g., bus, star, ring, etc.). For example, a tree network connected to a tree network is still a tree network, but two star networks connected together create a hybrid network topology. A hybrid topology is always produced when two different basic network topologies are connected. Two common examples for Hybrid network are: **Star Ring** network and **Star Bus** network.

- **Star Ring Network:** A Star Ring network consists of two or more star topologies connected by using a multi-station access unit (MAU) as a centralized hub.
- **Star Bus Network:** A Star Bus network consists of two or more star topologies connected by using a bus trunk (the bus trunk serves as the network's backbone).

Media used in communication

A transmission medium (plural transmission media) is a material substance (solid, liquid or gas) which can propagate energy waves. Media used in computer networks

are broadly classified into two: **Guided** and **Unguided,** which are further categorized as shown in Fig 2.31.

i. Guided Transmission Media

Guided Transmission Media also known as Bound Media uses a "cabling" system that guides the data signals along a specific path. The data signals are bound by the "cabling" system.

1. Unshielded Twisted Pair (UTP)

This is the most commonly used media used in networks. This consists of two wires twisted over one another which reduces the interference in signals.

Twisting the wires together results in a characteristic impedance for the cable. A typical impedance for UTP is 100 ohm for Ethernet 10BaseT cable. Unshielded Twisted Pair cable is used on Ethernet 10BaseT and can also be used with Token Ring.

The EIA/TIA (Electronic Industry Association/Telecommunication Industry Association) has established standards of UTP and rated five categories of wire, as shown in Table 2.4.

Type	Use
Category 1	Voice Only (Telephone Wire)
Category 2	Data speed up to 4 Mbps (LocalTalk)
Category 3	Data speed up to 10 Mbps (Ethernet)
Category 4	Data speed up to 20 Mbps (16 Mbps Token Ring)
Category 5	Data speed up to 100 Mbps (Fast Ethernet)
Category 6	Data speed up to 10 Gbps (Gigabit Ethernet)

Table 2.4: Categories of Unshielded Twisted Pair

Fig. 2.31 shows categorization chart of various Transmission Media.

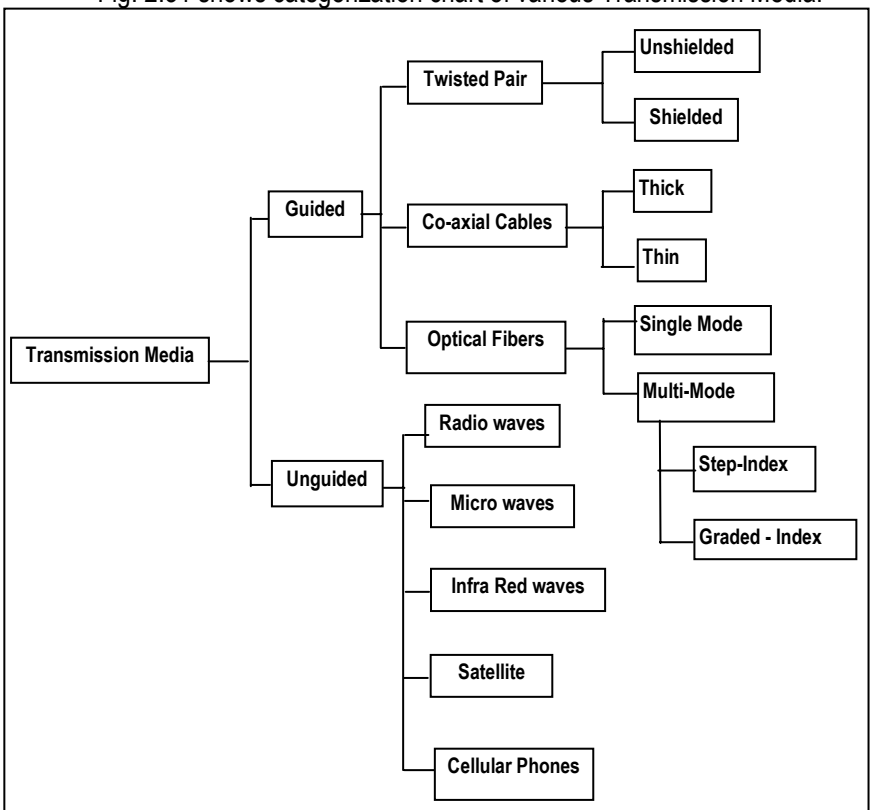


Fig. 2.31: Transmission Media

2. Shielded Twisted Pair

In case of shielded twisted pair (STP), there is a metallic shield either in the form of a foil or a braid over the twisted pair. The shield reduces the penetration of electro-magnetic stray signals (noise), to some extent.

STP or shielded twisted pair is used with the traditional Token Ring cabling or ICS - IBM Cabling System. IBM STP (shielded twisted pair) has a characteristic impedance of 150 ohms.

3. Coaxial Cable

Co-axial cable consists of a central core conductor of solid or stranded wires. The central core is held inside an insulator with the other conductor either in the form of a metal shield or a braid woven around it providing a shield. An insulating protective coating called a jacket covers the outer conductor.

Typical impedances for coaxial cables are 75 ohms for Cable TV, 50 ohms for Ethernet Thinnet and Thicknet. The good impedance characteristics of the coaxial cable allow higher data rates to be transferred than with twisted pair cable.

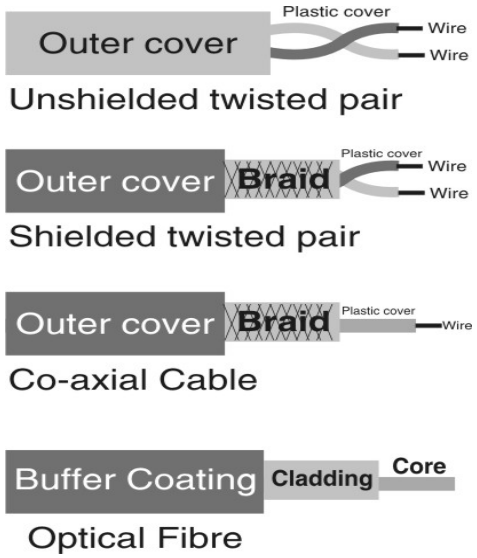


Fig. 2.32: Guided Media

Fig. 2.32 shows the picture of various guided media.

4. Optical Fiber Cable

The main disadvantage of twister pair and co-axial cables is that both are subjected to Electro-Magnetic Interference (EMI) and their bandwidth is limited. Optical Fibre overcomes these problems, because it uses light based signalling. The significant benefits are low attenuation and very high integrity over much longer intervals than metallic options.

In an optical fibre, the inner core consists of a pure glass material about the diameter of a human hair. An optical fibre has the following parts:

i. **Core :** a thin glass centre of the fibre through which light travels ;
ii. **Cladding :** the outer optical material surrounding the core that reflects the light back into the core.
iii. **Buffer coating :** plastic coating that protects the fibre from damage and moisture

The fibre optic system works on the principle of total internal reflection. The optical fibre system consists of three basic elements:

- the optical transmitter

- the fiber optic cable
- the optical receiver

Fig. 2.33 shows the optical fibre transmission system.

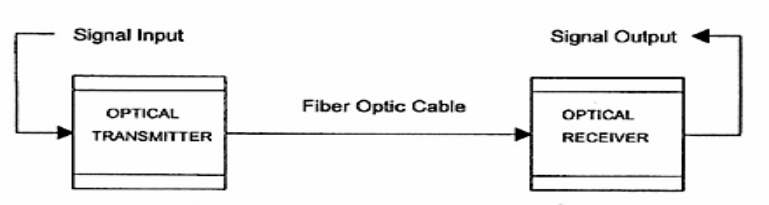


Fig. 2.33: An Optical Fibre System

Advantages

- Optical fibers can be drawn to smaller diameters than copper wires.
- **Higher carrying capacity :** Because optical fibers are thinner than copper wires, more fibers can be bundled into a given diameter cable than copper wires. This allows more packing.
- **Less signal degradation :** The loss of signal in optical fiber is less than in copper wire. **Low power :** Because signals in optical fibers degrade less than others like coaxial cables etc. , lower power transmitters can be used.

Disadvantages

- Very fragile than wires.
- Costlier than other guided media.
- Difficult to splice (i.e., it is extremely difficult to join optical fibre cables).

Factor	UTP	STP	Co-axial	Optical Fibre
Cost	Lowest	Moderate	Moderate	Highest
Installation	Easy	Fairly easy	Fairly easy	Difficult
Bandwidth	Typically 10Mbps	Typically 16Mbps	Typically 16Mbps	Typically 100Mbps
Attenuation	High	High	Lower	Lowest
EMI	Most vulnerable	Less vulnerable compared to UTP	Less vulnerable compared to STP	Not affected
Security	Easy to eavesdrop	Vulnerable to eavesdrop	Vulnerable to eavesdrop	Extremely difficult to eavesdrop

Table 2.5 compares various types of guided media for various parameters.

ii. Unguided Transmission Media

In Unguided Transmission Media, data signals travel without any guidance along a specific path. These are not bound to a cabling media and do not take a fixed path. Signals are broadcasted by transmitters and picked by receivers.

1. Radio waves

There are three types of RF (radio frequency) propagation:

- Ground Wave propagation
- Ionospheric propagation
- Line of Sight (LOS) propagation

Ground wave propagation: This propagation follows the curvature of the Earth**.**

Ionospheric propagation: The Earth is surrounded by three layers: Troposphere, where there is air; Stratosphere, where jets fly; and Ionosphere that contains charged particles. Ionosphere acts as a reflecting media that reflects the signal back to the earth. The transmitting equipment beams the signal towards the Ionosphere and the beam after hitting the Ionosphere layer gets reflected back, and the receiving station picks it up. The factors such as change in weather and time of the day have a significant influence in this propagation.

Line of Sight (LOS) propagation: In this, the transmitter and the receiver face each other. This is sometimes called space waves or tropospheric propagation. Curvature of the Earth for ground-based stations and reflected waves can cause problems.

Fig. 2.34 shows various types of Radio-Frequency propagation.

2. Microwave

Microwave transmission is a typical example of the line of sight transmission. The transmitting station must be in visible contact with the receiving station. This sets a limit on the distance between stations. If the distance is more, there is a need for repeaters.

Infrared Transmission: Infrared transmissions use Infra-Red radiation, a frequency range just below the visible light spectrum. This transmission can be used only for short distances.

Advantages:

i. Components used in IR transmission are relatively inexpensive.
ii. Signals can be reflected off surfaces (such as walls), so that direct line-of-sight is not necessary.
iii. Transmissions can be multidirectional.

Disadvantages:

i. Transmission distance is limited.
ii. Transmission cannot penetrate walls.
iii. Atmospheric conditions (such as rain or fog) can degrade the quality of the signal.

3. Satellite

Satellites act as transponders and play a crucial role in modern communication. Today, geostationary satellites orbiting at 36,000 km from the Earth's surface have good transponders which reflect signals from the source–to–destination. Fig. 2.35 shows satellite propagation.

4. Cellular Telephony

Modern day communications use cellular telephones in a big way. Computer systems are connected to the handset and from there these are connected to the **Mobile Telephone Switching Office (MTSO)**. MTSO is a central switch that controls the entire operation of a cellular system. It is a sophisticated computer that monitors all cellular calls, tracks the location of all cellular-equipped vehicles traveling in the system, arranges handoffs, and keeps track of billing information.

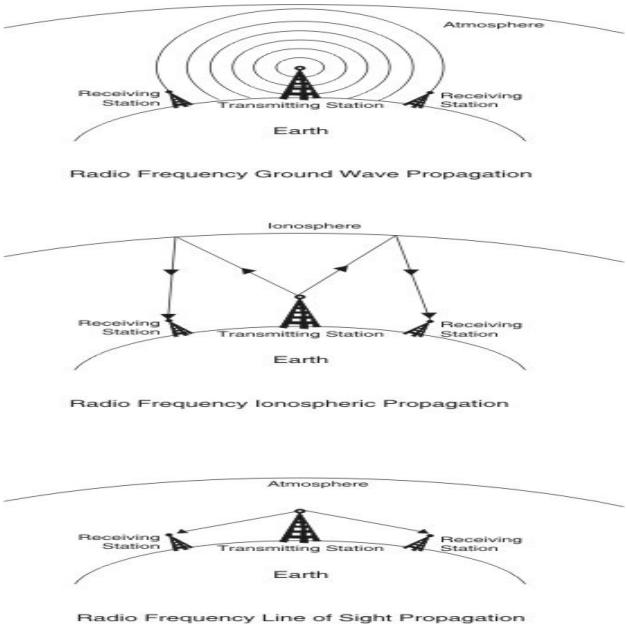


Fig. 2.34: Various types of Radio Frequency Propagation

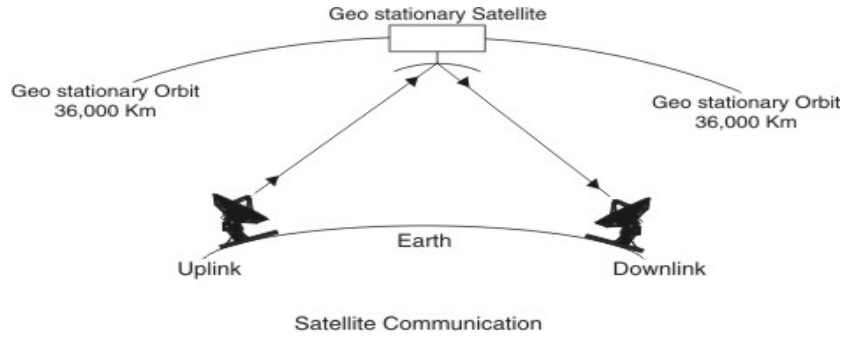


Fig. 2.35: Satellite Communication

Factors that influence the use of media

The main factors that influence the use of appropriate media include:

1. Cost
2. Quality of Service
3. Attenuation
4. Speed
5. Security
6. Immunity from Electromagnetic Interference (EMI)

Factors that degrade a signal

Three factors that degrade the quality of a signal are:

1. **Attenuation**: In attenuation, as the signal travels through the medium, it loses its strength.
2. **Delay Distortion:** This problem is of a composite signal, as it consists of multiple signals of varying frequencies. When all of them arrive at the same time, there is no distortion. But if they arrive at different times, then there is delay distortion.
3. **Noise:** Any unwanted stray signal that is present in a signal is called noise. Sometimes the noise signals interfere with the regular ones, making it difficult for the receiver to interpret the signal correctly.

Summary

This chapter explains the concept of network and its types. Data communication can take place by using various modes of data transmission, depending upon the direction of data exchanges, the number of bits sent simultaneously and the synchronization between senders and receivers. MODEM uses the concept of

multiplexing and de-multiplexing. Various transmission media are used to establish the physical and logical topologies with which various computers can be connected to form a network.

Questions

1. Which of the following is not a type of Radio Frequency Propagation?
   a. Ground Wave propagation
   b. Infrared Waves propagation
   c. Ionospheric propagation
   d. Line of Sight (LOS)

2. Which of the following is not a switching technique?
   a. Token Switching
   b. Circuit Switching
   c. Packet Switching
   d. Message Switching

3. A topology in which every device has two neighbours for purposes of communication is called_____.
   a. Bus
   b. Ring
   c. Tree
   d. Hybrid

4. Unidirectional Data Transmission is called _____.
   a. Simplex
   b. Duplex
   c. Half-Duplex
   d. None of these

5. On the basis of the number of bits sent simultaneously, the transmission mode is categorized into _____.
   a. Simplex and Duplex
   b. Client Server and Peer-to-Peer architecture
   c. Simplex and Half-Duplex
   d. Serial and Parallel Transmission

6. Telephone system is an example of _____ mode.
   a. Simplex
   b. Half-duplex
   c. Full-duplex
   d. None of these

7. Serial Transmission is categorized into _____.
   a. Synchronous and Asynchronous Transmission
   b. Parallel Transmission and Duplex Transmission
   c. Synchronous and Duplex Transmission
   d. None of these

8. Asynchronous Transmission is also called _____ .
   a. Store-and –Forward Transmission
   b. Start-and-Stop Transmission
   c. Stop - and- Store Transmission
   d. None of these

9. Microwave transmission is a typical example of _____.
   a. Line of sight transmission
   b. Ground Wave
   c. Infrared Wave
   d. Radio Waves

10. A topology in which every node is physically connected to every other node is_____ .
   a. Tree
   b. Star
   c. Mesh
   d. Bus

11. Which of the following is not a type of Multiplexing?
   a. Time Division Multiplexing
   b. Wavelength Division Multiplexing
   c. Packet Multiplexing
   d. Frequency Division Multiplexing

12. _____ is defined as the set of techniques that permits the simultaneous transmission of multiple signals on a single carrier.
   a. Multiplexing
   b. Switching
   c. Topology
   d. Synchronization

13. In _____ TDM, the multiplexer allocates exactly the same time slot to each device at all times, whether or not a device has anything to transmit.
   a. Synchronous
   b. Asynchronous

c. Serial
d. Multiplexing

14. MAU is_____.
    a. Miscellaneous Access Units
    b. Multi Access Unicode
    c. Multi - station Access Units
    d. Miscellaneous Access Unicode

15. SDSL is_____.
    a. Symmetric Digital Subscriber Line
    b. Subscriber Digital Subscriber Line
    c. Subscriber Digital Symmetric Line
    d. Symmetric Digital Symmetric Line

16. A disadvantage of star network topology is that_____.
    a. If a cable breaks, the entire network is affected
    b. If the hub fails, the entire network can fail
    c. Network expansion is difficult
    d. Isolating problems is difficult

17. The main disadvantage of peer-to-peer networking is:
    a. The networks are difficult to configure
    b. The networks are expensive
    c. The network is less secure than a server based network
    d. It follows a Master/Slave topology

18. Which one of the following topologies is the easiest to expand?
    a. Bus
    b. Ring
    c. Star
    d. Mesh

19. Which of the following is not a factor that influences the use of appropriate media?
    a. Cost
    b. Attenuation
    c. Security
    d. Noise

20. Which of the following does not degrade the quality of a signal?
    a. Attenuation
    b. Delay Distortion
    c. Security
    d. Noise

Answers:

1 b	2 a	3 b	4 a	5 d	6 c
7 a	8 b	9 a	10 c	11 c	12 a
13 b	14 c	15 a	16 b	17 c	18b
19 d	20 c				

3 Introduction to the OSI Model

Learning Objectives

To understand the :

- Basic concept of Protocols and Standards.
- Working of various Standards Creating Committees.
- Basic concept of the OSI Model and the functioning of its layers.
- Advantages & disadvantages of the OSI model.
- Various networking devices used at each layer end.
- Concept of Network Management through ISO Network Management Model.
- IEEE LAN standards.

Introduction

A **Network** may be described as a group or collection of computers connected together for sharing resources and information. Where individual networks are connected by different types of network devices, such as routers and switches, it is called **Internetwork.** As already discussed in Chapter 2, networks can be classified into Local Area Network (LAN), Wide Area Network (WAN), Metropolitan Area Network (MAN), Wireless LANs, Home Networks and Inter-networks depending on their range or scale.

Enabling effective and secure communication of various systems (within an intranetwork) with disparate technologies is a challenging task. The **O**pen **S**ystems **I**nterconnection **(OSI)** model enables easy integration of various technologies and provides solutions for managing the internetworking environment.

Protocol

In computer networks, communication occurs between entities that share application programs, file transfer packages, browsers, and database management systems in different systems. A networking protocol is defined as a set of rules that governs data communication over a network for what is communicated, how it is communicated and when it is communicated.

The key elements of protocol are:

(a) **Syntax :** It is the structure or format of the data, that is, the order in which they are presented.
(b) **Semantics :** It means each section of bits, how a particular pattern is to be interpreted, and based on that, what action to be taken.
c. **Timing :** It indicates when the data is to be transmitted and how fast it can be sent.

Standards

Standards are documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure that materials, products, processes and services are fit for their purpose. These provide a model for development that makes it possible for a product to work regardless of the individual manufacturer. These are essential for creating and maintaining an open and competitive market for manufacturers and provide guidelines to vendors, government agencies and other service providers to ensure the kind of interconnectivity necessary in today's marketplace and in international communications.

Benefits of standardization:

- Allows different computers to communicate.
- Increases the market for products adhering to the standard.

Two categories of standards:

1. **De facto** (Latin for "from the fact") standards are those that have just happened, without any formal plan. E.g., IBM PC for small office computers, UNIX for operating systems in CS departments.
2. **De jure** (Latin for ``by law'') standards are formal legal standards adopted by an authorized standardization body.

Standards Creation Committees

Standards are developed by the cooperative efforts of standards creation committees, forums, and government regulatory services. Some of the major International, Regional and National standards creating organizations are:

I. International

a. International Standards Organization (ISO):
ISO is a non - governmental worldwide federation of national standards bodies from some 100 countries, one from each country, established in 1947. Its mission is to

promote the development of standardization and related activities in the world to facilitate the international exchange of goods and services, and to develop cooperation in the spheres of intellectual, scientific, technological and economic activities. ISO's work results in international agreements which are published as International Standards.

b. International Electro - technical Commission (IEC):

The object of International Electro-technical Commission, which is composed of National Committees, is to promote international co-operation on all questions of standardization and related matters in the fields of electrical and electronic engineering and thus to promote international understanding.

c. International Telecommunication Union (ITU):

The International Telecommunication Union (ITU) is an intergovernmental organization, in which the public and private sectors cooperate for the development of telecommunications. The ITU adopts international regulations and treaties governing all terrestrial and space uses of the frequency spectrum as well as the use of the geostationary-satellite orbit, within which countries adopt their national legislation. It also develops standards to facilitate the interconnection of telecommunication systems on a worldwide scale regardless of the type of technology used.

d. Video Electronics Standards Association (VESA):

The Video Electronics Standards Association (VESA) is an international organization that sets and supports industry-wide interface standards for the PC, workstation, and other computing environments. VESA promotes and develops timely, relevant and open standards for the electronics industry, ensuring interoperability and encouraging innovation and market growth.

e. The Internet Society:

The Internet Society is a non-governmental International organization for global cooperation and coordination for the Internet and its internetworking technologies and applications. The Society's individual and organizational members are bound by a common stake for maintaining the viability and global scaling of the Internet. The Society is governed by its Board of Trustees elected by its membership around the world.

f. The World Wide Web Consortium (W3C)

The World Wide Web Consortium (W3C) works to realize the full potential of the Web. It develops common standards for the evolution of the Web by producing specifications and reference software. Although it is funded by industrial members, its products are freely available to all.

g. The Institute of Electrical and Electronics Engineers (IEEE):

The **Institute of Electrical and Electronics Engineers (IEEE)** is the world's largest technical professional society. Founded in 1884 by a handful of practitioners of the new electrical engineering discipline, the Institute presently has more than 320,000 members who conduct and participate in its activities in 147 countries. IEEE sponsors technical conferences, symposia and local meetings and educational programs to update its members' knowledge and expertise (state-of-the-art?). The purpose of all these activities is two- fold: (1) to enhance the quality of life for all peoples through improved public awareness of the influences and applications of its technologies; and (2) to advance the standing of the engineering profession and its members.

h. The Internet Engineering Task Force (IETF):

The Internet Engineering Task Force (IETF) is (the protocol engineering?) and a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. The technical work of the IETF is done in its working groups, which are organized into several areas (e.g., routing, network management, security, etc.).

II. Regional

a. Committee European de Normalization:

The European Committee for Standardization is responsible for European standardization in all fields except Electro-technical (CENELEC) and Telecommunications (ETSI). Related project of the CEN on the web is the standardization of the European character set in the fields of identification, coding, and others.

b. The European Telecommunications Standards Institute (ETSI)

The Internet Engineering Task Force (IETF) is the protocol engineering and development arm of the Internet. It is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and its smooth operation.

c. The European Workshop on Open system (EWOS)

The European Workshop on Open Systems is the European forum for one-stop development of technical guidance and pre-standards in the information and communications technologies (ICT) field, working for the benefit of vendors, planners, procurers, implementers and users.

III. National

a. Standards Australia (SAA)

Standards Australia is the Australian representative on the two major international

standardizing bodies, the International Organization for Standardization, ISO and the International Electro-technical Commission, IEC. Its original name was the Australian Commonwealth Engineering Standards Association and was funded in 1922. Its mission is to excel in meeting the needs of Australia's technical infrastructure for contemporary, internationally aligned standards and related services which enhance the nation's economic efficiency, international competitiveness, and fulfills community's desire for a safe and sustainable environment.

b. Standards Council of Canada (SCC)

The Standards Council coordinates the contribution of Canadians to the two most prominent international standards-writing forums : the International Organization for Standardization (ISO) and the International Electro-technical Commission (IEC). The SCC's activities are carried out within the context of the National Standards System, a federation of organizations providing standardization services to the Canadian public. The SCC is the manager of the System.

Open Systems Interconnection (OSI) Reference Model

In today's complex computing environment, messages are sent across heterogeneous networks with a large variety of hardware technologies, networking devices and protocols, device drivers, operating systems, application software, and so on. So there is a need for resolving these complexities universally, without causing worries to the users. The answer is the **Open Systems Inter-connection Reference Model** that makes inter-operability across heterogeneous technology environments possible.

The OSI defines in great detail the various activities that need to take place and how they should take place, to make effective communication across heterogeneous data networks a reality. The OSI model formulated by ISO enables systems with different technologies and capabilities to work seamlessly within the same network as well as across heterogeneous networks. It provides a set of open system standards that enables vendors to benchmark and compare different communication systems and for structuring inter-computer and inter-network communications processes.

The **Open System Interconnection (OSI) Reference Model** describes how information from a software application in one computer moves through a network medium to a software application in another computer. In other words, the model describes the concepts involved when the message or data or instruction from the sender's computer or device moves over a range of network environments and reaches the receiver's computer or device, as the sender sent it.

The OSI reference model is a layered model of seven layers, each specifying particular network functions. It was developed by the International Organization for Standardization (ISO) in 1984, and is now considered the primary architectural model

for inter-computer communications. The model divides the tasks involved with moving information between networked computers into seven smaller, more manageable task groups. A task or group of tasks is then assigned to each of the seven OSI layers. Each layer is reasonably self-contained so that the tasks assigned to it can be implemented independently. This enables the solutions offered by one layer to be updated without adversely affecting the other layers.

Fig. 3.1 shows the seven layers of the OSI Model. These layers belong to three subgroups:

1. **Layers 1, 2 and 3 -** Physical, Data Link and Network, are the Network support layers dealing with the physical aspect of transmitting data from one device to another.
2. **Layers 5, 6 and 7 -** Session, Presentation and Application, can be considered as the user support layers that allow interoperability among unrelated software systems.
3. **Layer 4 -** the Transport Layer, ensures end-to-end reliable data transmission while Layer 2 ensures reliable transmission on a single link.

- Layer 7 – Application Layer
- Layer 6 – Presentation Layer
- Layer 5 – Session Layer
- Layer 4 – Transport Layer
- Layer 3 – Network Layer
- Layer 2 – Data Link Layer
- Layer 1 – Physical Layer

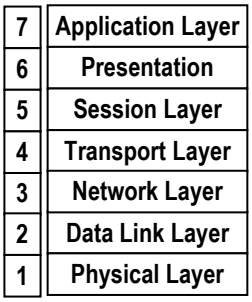


Fig. 3.1: OSI Seven Layers

The upper OSI layers (Layers 5,6 and 7) are almost implemented in software; whereas lower layers (layers 2, 3 and 4) are combination of hardware and software, and the physical layer (Layer 1) is almost hardware. The handy way to remember the seven layers is the sentence "**All people seem to need data processing.**" The beginning letter of each word corresponds to a layer.

- All - Application layer
- People - Presentation layer
- Seem - Session layer
- To - Transport layer
- Need - Network layer
- Data - Data link layer
- Processing - Physical layer

How does the OSI work?

Data or Information transferred from a communication software application in one computer system to a software application in another must pass through a category of tasks that are relatable to the OSI layers. For example, if a software application in System A needs to transmit information to a software application in System B, the application program in System A will send this information to the application layer (Layer 7) of System A. The application layer then passes this information to the presentation layer (Layer 6), which sends the data to the session layer (Layer 5), and so on down to the physical layer (Layer 1).

At the physical layer, the information is placed on the physical network medium and is sent across the medium to System B. The physical layer of System B removes the information from the physical medium, and passes the information up to the data link layer (Layer 2), which passes it to the network layer (Layer 3), and so on, until it reaches the application layer (Layer 7) of System B. Finally, the application layer passes the information to the recipient application program to complete the communication process.

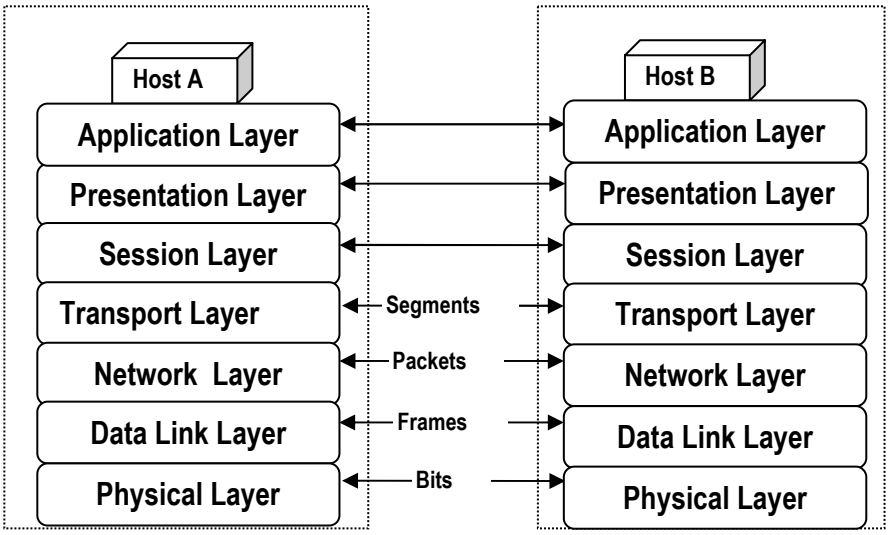


Fig. 3.2: Peer –to – Peer Communication

Interaction between the OSI layers

One layer in the OSI model generally interacts with three other layers:

- the layer directly above it,

- the layer directly below it, and
- the corresponding layer in the other system.

When information or data, such as an email, is sent from one machine to another on the Internet, the message is broken down into chunks or parts of a certain size in bytes, called a Packet. The messages have to be segmented into smaller units so that it is easy to transport these over the networks. In Fig. 3.3, L7 data refers to the data unit at layer 7, L6 data means the data unit at layer 6 and so forth. The process starts at layer 7 of the sender's machine and starts descending down sequentially from layer to layer. At each layer, a header is added to the data unit; at layer 2, a trailer is also added to the data unit. When the formatted data passes through the physical layer, it is transformed into an electromagnetic signal and transported along a physical link.

Upon reaching its destination, the process gets reversed. Here, the data units move sequentially in ascending order from layer 1 to layer 7. As each data unit reaches the next higher layer, the headers and trailers attached to each layer and the headers and trailers attached to it at the corresponding sending layer are removed, and actions appropriate to that layer are taken.

Understanding the OSI Layers

I. Application Layer

The application layer is the closest layer to the end user, which means that both the OSI application layer and the user interact directly with the software application.

Responsibilities of the Application Layer include the following:

i. This layer provides a means for the user to access information on the network through an application, but does not include the application itself. It provides an interface to the user to interact with the application and therefore the network.
ii. This layer interacts with software applications that implement a communicating component. Such application programs fall outside the scope of the OSI model. For example, the Internet browser is an example of the communication service application that functions at this layer. Some other interfaces and support for services like e-mail, remote file access and transfer, shared database management, and other types of information services are provided at application layer level only.
iii. Application layer functions include identifying communication partners, determining resource availability, and synchronizing communication. When identifying communication partners, the application layer determines the identity and availability of communication partners for an application with data to

transmit. When determining resource availability, the application layer decides whether sufficient network resources for the requested communication exist. In synchronizing communication, all communication between applications requires cooperation that is managed by the application layer.

iv. **Network Virtual Terminal :** This is the software version of a physical terminal and allows a user to log on to a remote host. To do so, the application creates a software emulation of a terminal at the remote host which enables the user computer to communicate with the software terminal, which, in turn, talks to the host, and vice versa. The remote host believes it is communicating with one of its own terminals and allows the user to log on. To illustrate, if System A is being networked to a mainframe machine that runs on the IBM operating system say z/OS 390, the command syntax and semantics in System A and commands in a Windows based PC machine are very different. The Application layer defines a Network Virtual Terminal which helps one machine to communicate with another, thus shielding the user of the problems in converting, translating the command communication and interpretation.

v. **File Transfer, Access and Management (FTAM) :** Different file systems have different file naming conventions, different ways of representing text lines, and so on. This application allows a user to access files in a remote computer (to make changes or read data), to retrieve files from a remote system, and to manage or control files on a remote computer.

vi. **Mail Services :** This application provides the basis for e-mail forwarding and storage. SMTP and POP3 are popular protocols that enable this in the TCP/IP suite.

vii. **Directory Services:** This application provides distributed databases sources and access to global information about various objects and services.

Various protocols in the Application Layer include Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Telnet, Simple Mail Transfer Protocol (SMTP), Simple Network Management Protocol (SNMP), Network News Transfer Protocol (NNTP), X.400, X.500, Server Message Block (SMB), and Network File System (NFS).

II. Presentation Layer

The primary function of this layer is to take care of the syntax and semantics of the data transmission between two systems.

Responsibilities of the Presentation Layer include the following:

i. **Translation:** This layer receives information from the application layer and converts it in an ordered and meaningful format to ensure that it is

understandable to all the systems following the OSI model. Since different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. This layer does the important job of converting the various services from their proprietary application formats to universally accepted formats, ASCII, EBCDIC.

ii. **Encryption/Decryption :** A system must be able to conserve privacy of the data sent from one system to another. Encryption is a technique provided at this level wherein the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.

iii. **Compression :** Data compression reduces the number of bits to be transmitted and thus is important in the transmission of multimedia, such as text, audio, and video. Among the well-known graphic image formats are Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), and Tagged Image File Format (TIFF).

Common data representation formats, or the use of standard image, sound, and video formats, enable the interchange of application data between different types of computer systems. Standard data compression schemes enable data that is compressed or encrypted at the source device to be properly decompressed, or deciphered at the destination.

III. Session Layer

When two applications need to communicate or transfer information between them, this establishes the session connection, maintains connection between the transfer and controls its termination. Its functions further include the management of dialogue control, token management, insertion of checkpoints, and synchronisation.

Responsibilities of the Session Layer include the following :

i. **Session Establishment:** This layer allows users on different machines to establish sessions between them in order to allow any user to log onto a remote timesharing system or to transfer files between two machines.

ii. **Dialog Control:** The session layer allows the systems to enter into a dialog that allows the communication either in half-duplex or full-duplex. It is like two people communicating over a telephone, where the telephone network circuitry and protocols establish the connection over the telephone line and maintain the communication path during the conversation period and disconnect the call when the user hangs up.

iii. **Token Management :** To manage a crucial issue of both sides not sending the token at the same time, the session layer provides tokens that can be exchanged

between the two parties. The side carrying the token may perform the critical operation of sending the data, thus avoiding collisions between data packets.

iv. **Synchronization :** The session layer allows a process to add checkpoints (synchronization points) into a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens at page 624, retransmission begins at page 601: pages 1 to 600 need not be retransmitted.

Some examples of session-layer implementations include Zone Information Protocol (ZIP), the AppleTalk protocol that coordinates the name binding process, and Session Control Protocol (SCP), the Decent Phase IV session layer protocol.

IV. Transport Layer

This layer acts as a bridge between the upper and lower layers, and has a crucial role in the OSI model. Some of the tasks performed by it include segmentation and reassembly, end - to - end message delivery, connection control and flow control. This layer establishes a logical connection between the sending and destination systems on the internetwork.

Responsibilities of the Transport Layer include the following:

i. **End-to-End Delivery:** This layer is responsible for source - to - destination (end-to-end) delivery of the entire message, whereas the network layer oversees the end-to-end delivery of individual packets, making sure that the data is delivered error-free and in a proper sequence.

ii. **Connection Control:** This layer can be either connectionless or connection-oriented. A connectionless layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection-oriented transport layer makes a connection between the two end ports for more security. A connection is defined as a single logical path between the source and destination associated with all the packets in a message.

iii. **Segmentation and Reassembly:** The transport layer divides the data into transmittable segments, each segment containing a sequence number, which enables the layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that are lost in the transmission.

iv. **Error & Flow Control:** The error and flow control at this layer is performed end to end rather than across a single link. The sender's transport layer ensures that the entire message arrives at the receiving transport layer without any error, damage, loss, or duplication.

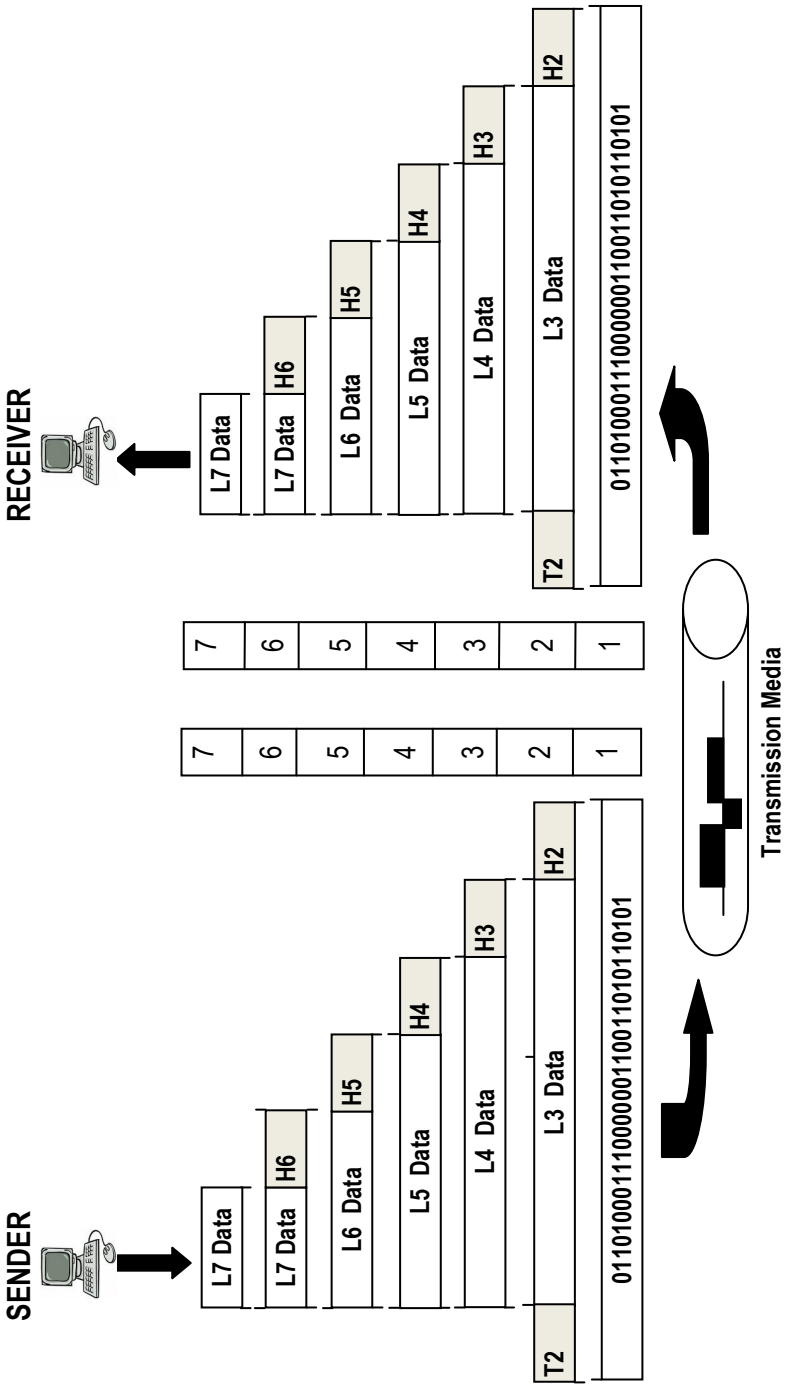


Fig. 3.3: An Exchange of data using the OSI Model

Examples of transport layer implementations are Transmission Control protocol (TCP), Name Binding Protocol (NBP), OSI transport protocols, NetBEUI, and Sequenced Packet Exchange (SPX).

Connection - Oriented vs Connection - less Communication

Connection - Oriented Communication

Connection-oriented service is modeled after the telephone system. In this, the transmitting device first establishes a connection - oriented setup with its peer system, which is called the Three way handshake. Creating a connection is a 3-step process:

- Connection Establishment.
- Data Transfer, and
- Connection Release.

The steps shown in Fig. 3.4 can be described as:

1. The first connection agreement segment is a request for synchronization.
2. The second and third segments acknowledge the request and establish the connection parameter or rules.
3. The final segment is also an acknowledgement. It notifies the destination system that the connection agreement has been accepted and that the actual connection has been established, after which data transfer begins.

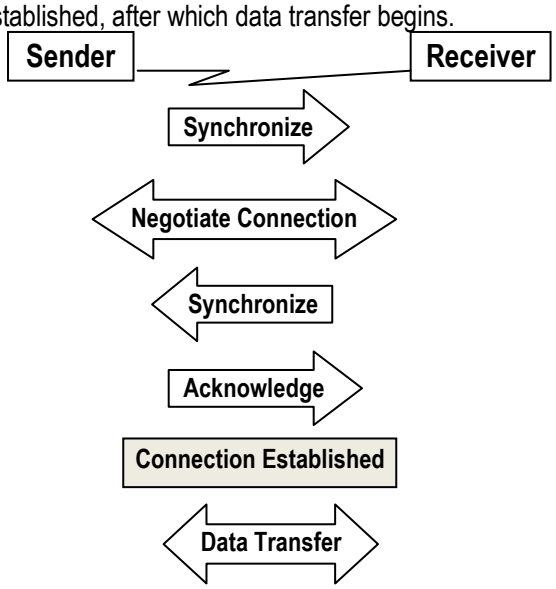


Fig. 3.4: Establishing a Connection - Oriented Service

In connection - oriented communication, the chain of nodes forms a kind of logical pathway. The nodes forwarding the data packet provide information about which packet is part of which connection. This enables the nodes to provide flow control as the data moves along the path. For example, if a node determines that a link is malfunctioning, it sends a notification message backward, through the path to the source computer. Moreover, in a connection-oriented communication, nodes have the facility to provide error correction at each link in the chain. If a node detects any error, it asks the preceding node to retransmit.

Connection - less communication: Connection - less Service is modeled after the postal system. In this, each packet carries the full destination address and each one is routed through the system independent of all others. All activities regarding error correction and retransmission are carried out by the source and destination nodes. The nodes merely acknowledge the receipt of packets and in case of errors, merely retransmit. Internal nodes do not participate in flow control too. The main advantage of connectionless mode is that connectionless communication works faster though it has some downsides.

The main disadvantages are:

- Messages are lost sometimes.
- If a message gets lost, the sender doesn't receive any information regarding it.
- Retransmission takes longer time relatively.

In a connection - oriented communication, the service should firstly establish a connection before passing on any information. Connection - less service can send the information without requiring any connection. Connection-oriented services provide reliable transmission of data whereas connectionless services do not.

The Transmission Control Protocol (TCP) is an example of a connection - oriented communication whereas the User Datagram Protocol (UDP) is an example of a connection - less communication.

V. Network Layer

The main tasks of a network layer include routing, congestion control, logical addressing, and address transformation and delivery of data from source - to - destination. These ensure that the network layer transports traffic between devices that are not locally connected.

Responsibilities of the Network Layer include the following:

i. **Routing:** If two systems are connected to the same link, there is usually no need for a network layer. But when independent networks of links are connected

together to create internetwork or a large network, that information or messages can take many routes or paths to reach their destination. The connecting devices called **Routers** or **Gateways** use a mechanism at the network layer level to determine the best path for the packet to take to reach its destination. Hence network addresses play a very critical role in the activities at this layer.

ii. **Congestion Control:** If too many packets are present in the subnet at the same time, they create a bottleneck situation. Such a congestion control is handled by the network layer.

iii. **Logical Addressing:** Logical address refers to a network layer address such as an IP address which is required when the data packets pass the network boundary, whereas Physical addressing is implemented by the Data Link layer that handles the data packets transmitted locally. The network layer is responsible to add a header including the logical address of the sender and receiver to the packet received from the upper layers.

Network layer protocols are usually routing protocols such as Border Gateway Protocol (BGP), Open Shortest Path First (OSPF), Routing Information Protocol (RIP), and Internetwork Packet Exchange (IPX) Protocol. Other types of protocols, like the Internet Protocol (IP), are also used.

(VI) Data Link Layer

The main tasks of a data link layer include error control, flow control, acknowledgement and framing of data packets. The data link layer is responsible for identifying the device that resides on the local network.

Responsibilities of the Data Link Layer include the following:

i. **Data Framing**: The Data Link Layer provides the stream of bits receive from the network layer into manageable data units called frames.

ii. **Physical Addressing:** The DLL adds a header to the frame to define the physical address of the sender and/or receiver of the frame.

iii. **Flow Control:** The DLL imposes a data flow control mechanism in case the rate of production of data frames does not match the rate of absorption of the same at the receivers' end.

iv. **Error Control:** The DLL adds reliability to the physical layer by adding mechanisms such as acknowledgement frames, detection and retransmission of damaged or lost frames, etc.

v. **Access Control:** When two or more devices are connected to the same link, DLL protocols are necessary to determine which device has control over the link at any given time.

Protocols for Data link layer include Ethernet, ATM, LocalTalk, Token Ring, SLIP, PPP, MTU, and CSLP and Fiber Distributed Data Interface (FDDI).

The IEEE standard divides DLL into two sub layers : **Logical Link Control (LLC)** and **Medium Access Control (MAC)** as shown in Fig. 3.5.

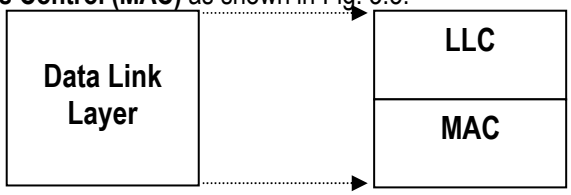


Fig. 3.5 : Sub-layers of Data Link Layer

Logical Link Control (LLC) defines how data is transferred over the cable and provides data link service to the higher layers.

Medium Access Control (MAC) determines who uses the network when multiple computers are trying to access it simultaneously (i.e. Token passing, Ethernet, CSMA/CD).

VII. Physical Layer

The focus of the Physical Layer is more towards transparent transmission of information over a communications channel. It specifies the mechanical, electrical, functional and procedural mechanisms for initiating, maintaining and de-activating the physical link between the systems.

Responsibilities of the Physical Layer include the following:

i. **Physical Characteristics of interfaces and media:** This layer defines the characteristics of the interface between the devices and the transmission medium.
ii. **Representation of bits:** To transmit data in bit form at the physical level end, bits must be encoded into electrical or optical signals. The physical layer defines the type of encoding (how 0s and 1s are changed to signals) being used in data transmission.
iii. **Data Rate:** The transmission rate, that is the number of bits sent, is also defined by the physical layer.
iv. **Synchronization of bits :** The sender and receiver clocks are synchronized at this level.

v. **Line Configuration:** It handles the connection of devices to the medium.
vi. **Physical Topology:** It is concerned with the topology used for connecting the devices.
vii. **Transmission Mode:** The physical layer also defines the direction of transmission between two devices - simplex, half duplex and duplex.

Some of the important standards that deal with the physical layer specifications are: RS-232(for serial communication lines), X.21, EIA 232, and G730.

Advantages of the ISO OSI Model

- One of the key advantages of the model is that it supports inter-operability and portability.
- Changes in component technologies and product innovations are not affected by their capability to inter-connect.
- Allows modularity because of which products performing only a part of the communication activities can interface easily with other components.
- Promotes standardisation of interface design.

Disadvantages of the ISO OSI Model

- Hinders technological advancement at the protocol suite architecture level, thus forcing innovations to conform to old clumsy standards.
- Strict conformity to OSI layering may not be possible with growing integrated technologies.
- Layering into real-time applications for purposes of standardisation may hamper performance and efficiency.

The Table 3.1 summarizes the functions of OSI Layers.

Layer	Function	Protocols
Application User Interface	• Used for applications specifically written to run over the network. • Allows access to network services that support applications. • Directly represents the services that directly support user applications. • Handles network access, flow control and error recovery. • Examples are file transfer, e-mail, NetBIOS-based applications.	DNS, FTP, TFTP, BOOTP, SNMP, RLOGIN, SMTP, MIME, NFS, FINGER, TELNET, NCP, APPC, AFP, SMB

Presentation Translation	• Translates from application to network format and vice-versa. • All different formats from all sources are made into a common uniform format that the rest of the OSI model can understand. • Responsible for protocol conversion, character conversion, data encryption / decryption, expanding graphics commands, data compression. • Sets standards for different systems to provide seamless communication from multiple protocol stacks. • Not always implemented in a network protocol.	ASCII EBCDIC ICA LPP
Session Syncs and Sessions	• Establishes, maintains and ends sessions across the network. • Responsible for name recognition (identification), so only the designated parties can participate in the session. • Provides synchronization services by planning check points in the data stream. If a session fails, only data after the most recent checkpoint need be transmitted. • Manages to transmit data at a certain time and for how long. • Examples are interactive login and file transfer connections. The session would connect and re-connect if there was an interruption; recognize names in sessions and register names in log.	NetBIOS Names Pipes Mail Slots RPC
Transport Packets; Flow control & Error-handling	• Additional connection below the session layer. • Manages the flow control of data between parties across the network. • Divides streams of data into chunks or	TCP, ARP, RARP. SPX NWLink NetBIOS / NetBEUI ATP

	packets; the transport layer of the receiving computer reassembles the message from packets. • A train is a good analogy: the data is divided into identical units (bogies). • Provides error-checking to guarantee error-free data delivery, with losses or duplications. • Provides acknowledgment of successful transmissions; requests retransmission if some packets don't arrive error-free. • Provides flow control and error-handling.	
Network Addressing; Routing	• Translates logical network address and names to their physical address. • Responsible for - o Addressing; o determining routes for sending; o managing network problems, such as packet switching, data congestion and routing. • If a router can't send data frame as large as the source computer sends, the network layer compensates by breaking the data into smaller units. At the receiving end, the network layer reassembles the data. • Think of this layer stamping the addresses on each train car.	IP, ARP, RARP, ICMP, RIP, OSFP. IGMP IPX NWLink NetBEUI OSI DDP DECnet
Data Link Data frames to bits	• Turns packets into raw bits of 0 and 1 and at the receiving end turns bits into packets. • Handles data frames between the Network and Physical layers. • The receiving end packages raw data from the Physical layer into data frames for delivery to the Network layer. • Responsible for error-free transfer of	**Logical Link Control** • error correction and flow control. • manages link control and defines SAPs. 802.1 OSI Model 802.2 Logical Link

	frames to other computers via the Physical Layer. • This layer defines the methods used to transmit and receive data on the network. It consists of the wiring, the devices used to connect the NIC to the wiring, the signaling involved to transmit / receive data and the ability to detect signaling errors on the network media.	Control **Media Access Control** • communicates with the adapter card. • controls the type of media being used: - 802.3 CSMA/CD (Ethernet) - 802.4 Token Bus (ARCnet) - 802.5 Token Ring - 802.12 Demand Priority
Physical Hardware; Raw bit stream	• Transmits raw bit stream over the physical cable. • Defines cables, cards, and physical aspects. • Defines NIC attachments to hardware and how the cable is attached to NIC. • Defines techniques to transfer bit stream to the cable.	IEEE 802 IEEE 802.2 ISO 2110 ISDN

Table 3.1: Summary Table of the Functions of OSI Layers

Networking Devices

Computer networking devices are units that mediate data in a computer network. Computer networking devices are also called network equipment, Intermediate Systems (IS) or Inter-Working Unit (IWU). Units which are the last receiver or generate data are called hosts or data terminal equipment. Table 3.2 displays the list of various Networking Devices used in Internetwork communication.

Networking Devices	OSI Layers	Definition
Common Basic Networking Devices		
Gateway 	4 - 7	A device residing at a network node for interfacing with another network that uses different protocols.
Router 	3	A specialized network device that determines the next network point to which to forward a data packet toward its destination. Unlike a gateway, it cannot interface different protocols.
Bridge 	2	A device that connects multiple network segments along the data link layer.
Switch 	2	A device that allocates traffic from one network segment to certain lines (intended destination(s)) which connect the segment to another network segment. So unlike a hub, a switch splits the network traffic and sends it to different destinations rather than to all systems on the network.
Repeater 	1	A device to amplify or regenerate digital signals received while setting them from one part of a network into another.
Hub 	1	Connects multiple Ethernet segments together making them act as a single segment. Using a hub, every attached device shares the same broadcast domain and the same collision domain. Therefore, only one computer connected to the hub is able to transmit at a time. It provides bandwidth which is shared among all the objects, compared to switches, which provide a dedicated connection between individual nodes.
Some Hybrid Network Devices		
Protocol Converter	1	A hardware device that converts between two different types of transmissions, such as asynchronous and synchronous transmissions.

Bridge Router (Brouter) 	2 - 3	Combines router and bridge functionality.
Hardware or software components that typically sit on the connection point of different networks, e.g. between an internal network and an external network.		
Proxy 	4 -7	Computer network service which allows clients to make indirect network connections to other network services.
Firewall 	4	A piece of hardware/software put on the network to prevent communications forbidden by network policy.
Network Address Translator	4	Network service provides a hardware or software that converts internal to external network addresses and vice versa.
Other hardware for establishing networks or dial-up connection.		
Multiplexer 	1	A device that combines several electrical signals into a single signal.
Network Interface Card 	1-7	A piece of computer hardware to allow the attached computer to communicate by network.
Modem 	1	A device that modulates an analog "carrier" signal (such as sound), to encode digital information, which also demodulates such a carrier signal to decode the transmitted information, as a computer communicating with another computer over the telephone network.
Line Driver	1	A device to increase transmission distance by amplifying the signal. Base-band networks only.

Table 3.2: List of some of Networking Devices

Network Management

Network management refers to the activities, methods, procedures, and tools that pertain to the operation, administration, maintenance, and provisioning of networked systems.

- Operation deals with keeping the network (and the services that the network provides) running smoothly. It includes monitoring the network to spot problems as soon as possible, ideally before users are affected.
- Administration deals with keeping track of resources in the network and how they are assigned. It includes all the "housekeeping" that is necessary to keep the network under control.
- Maintenance is concerned with performing repairs and upgrades. For example, when equipment must be replaced, when a router needs a patch for an operating system image, when a new switch is added to a network. Maintenance also involves corrective and preventive measures to make the managed network run "better", such as adjusting device configuration parameters.
- Provisioning is concerned with configuring resources in the network to support a given service. For example, this might include setting up the network so that a new customer can receive voice service.

The ISO Network Management Model

The International Organisation for Standardisation (ISO) addresses five major functional areas of the Network Management Model. These are:

i. Performance management,
ii. Accounting management,
iii. Configuration management,
iv. Fault management, and
v. Security management.

Performance Management

Performance management is monitoring, assessing, and adjusting the available bandwidth and network resource usage to make a network run more efficiently. It is majorly important particularly to the business and/or organization that wants to streamline their network's performance. Solar winds is a great tool for performance management.

Examples of performance variables that might be provided include network throughput, user response times, and line utilization.

Performance management involves three main steps:

a. Firstly, gathering of performance data for analysis.
b. Analysing the data to determine the baseline levels.
c. Finally, determining appropriate performance thresholds for every important parameter.

Accounting Management

Accounting management monitors and assesses the usage of data and/or resources for the purpose of billing. The goal of this is to measure network utilisation parameters so that an individual or a group whoever uses the network can be regulated appropriately for optimal resource utilisation. This aspect of the network management is by Internet Service Providers to bill customers for the resources they use.

Configuration Management

The goal of configuration management is to monitor network and system configuration information so that the effects on network operation of various versions of hardware and software elements can be tracked and managed. An example of this is Microsoft's System Management Server (SMS) which has the capability to monitor, manage and track every piece of software and hardware on a given network.

Fault Management

The goal of Fault Management is to detect, log and alert the system administrators of problems that might effect the systems operations.

Fault management is a three - step process:

a. Firstly, it determines symptoms and isolates the problem.
b. Then it fixes the problem and tests the solution on all important subsystems.
c. Finally, the detection and resolution of the problem are recorded.

Security Management

Security Management deals with controlling access to resources and even alerting the proper authorities when certain resources are accessed so that the network is not sabotaged intentionally or unintentionally and sensitive information is not accessed by those without appropriate authorisation. Intrusion detection systems such as Symantec's Intruder Alert have this security management capability.

Some Network Management Solutions Software

There are many products that support some or even all of these areas of network management. What most network management systems have in common is their use

of protocols, such as Simple Network Management Protocols (SNMP), SNMPv3, and Common Management Information Protocol (CMIP).

Some examples of products used for Network Management tasks are:

i. **Cisco WAN Manager:** It is a network and element management system that enables operations, administration, and maintenance of WAN multiservice networks.
ii. **NetBeacon Element Management System:** It provides network management that graphically shows all link connections, environmental conditions, port activity and status at a glance.
iii. **Concord Communications eHealth QoS:** QoS enables service providers and enterprises to offer different classes of service, with different performance commitments, across the same physical infrastructure.
iv. **Manage Engine OpManager:** It is a comprehensive Network, System and Application monitoring software that offers advanced network monitoring functionality at an affordable price.

IEEE LAN Standards

IEEE 802 refers to a family of IEEE standards dealing with Local Area Networks and Metropolitan Area Networks. More specifically, the IEEE 802 standards are restricted to networks carrying variable-size packets. The number 802 was simply the next free number IEEE could assign, though "802" is sometimes associated with the date of the first meeting held in/on February 1980.

The services and protocols specified in IEEE 802 map the lower two layers (Data Link and Physical) of the seven-layer OSI networking reference model. In fact, IEEE 802 splits the OSI Data Link Layer into two sub-layers : Logical Link Control (LLC) and Media Access Control (MAC), so that the layers can be listed like this:

- Data link layer :
  - LLC Sublayer
  - MAC Sublayer
- Physical layer

The IEEE 802 family of standards is maintained by the IEEE 802 LAN/MAN Standards Committee (LMSC). The most widely used standards are for the Ethernet family, Token Ring, Wireless LAN, Bridging and Virtual Bridged LANs. Table 3.3 lists IEEE LAN standards.

Name	Description
IEEE 802.1	Bridging Network an Network Management.
IEEE 802.2	Logical Link Control.

IEEE 802.3	Ethernet.
IEEE 802.4	Token Bus.
IEEE 802.5	Defines the MAC sub-layer for a Token Ring.
IEEE 802.6	Metropolitans Area Networks.
IEEE 802.7	Broadband LAN using Coaxial Cable.
IEEE 802.9	Integrated Services LAN.
IEEE 802.10	Interoperable LAN Security
IEEE 802.11 a /b/g/n	Wireless LAN & Mesh (Wi-Fi Certification).
IEEE 802.14	Cable Modems.
IEEE 802.15.1	Bluetooth Certification.
IEEE 802.16	Broadband Wireless Access (WiMAX Certification).
IEEE 802.20	Mobile Broadband Wireless Access.
IEEE 802.22	Wireless Regional Area Network.

Table 3.3: Some of IEEE LAN Standards

Summary

We have understood the basic concept of protocols, their usage and characteristics. The chapter deals with the seven layered OSI Model and its importance to carry out the data transmission over the network. IEEE standards are established that define the rules of data transmission and various networking devices used at each end of the machine.

Questions

1. Which of the following is not an element of Protocol?
   a. Syntax
   b. Semantics
   c. Format
   d. Timing
2. Which of the following is not an OSI Layer?
   a. Application Layer
   b. Circuit Layer
   c. Presentation Layer
   d. Transport Layer
3. OSI Stands for _____.
   a. Open systems Interconnection

b. Oval systems Interconnection
c. Open source Interconnection
d. None of these

4. The two sublayers of DLL are _____.
   a. Logical Level Control and Medium Access Control
   b. Logical Link Control and Medium Authority Control
   c. Logical Level Control and Medium Authority Control
   d. Logical Link Control and Medium Access Control

5. IETF Stands for _____.
   a. The Internet Engineering Task Force
   b. The Internet Engineering Travel Force
   c. The International Engineering Task Force
   d. The International Engineering Travel Force

6. Name the layer in the OSI model that does not interact with another layer.
   a. The layer directly above it
   b. The layer directly below it
   c. The layer itself in the same system
   d. The corresponding layer in the other system

7. The sequence of steps followed in connection-oriented service are ____.
   a. Connection Release, Data Transfer, and Connection Establishment
   b. Connection Establishment, Data Transfer, and Connection Release
   c. Connection Release, Connection Establishment, and Data Transfer
   d. Data Transfer, Connection Establishment, and Connection Release

8. Which of the following is not a part of the ISO Network Management Model?
   a. Fault Management
   b. Configuration Management
   c. Performance Management
   d. Data Management

9. Insertion of Checkpoints is handled at _____ level.
   a. Session Layer
   b. Data Link Layer
   c. Network Layer
   d. Physical Layer

10. IEEE 802.4 is for _____.
    a. Token Ring
    b. Bluetooth Certification

c. WiMax
d. Token Bus

11. OSPF stands for _____.
    a. Open Shortest Path First
    b. Open Shortest Pattern First
    c. Outsource Shortest Path First
    d. None of these

12. Trailer in a packet is added only at the _____.
    a. Session Layer
    b. Data Link Layer
    c. Network Layer
    d. Physical Layer

13. CMIP stand for _____
    a. Customer Management Information Protocol
    b. Customer Management Input Process
    c. Common Management Information Protocol
    d. Common Management Input Protocol

14. A specialized network device that determines the next network point to which a data packet is forwarded toward its destination is called _____.
    a. Gateway
    b. Router
    c. Firewall
    d. Hub

15. TIFF stands for _____.
    a. Tagged Image File Force
    b. Tagged Image File Format
    c. Tagged International File Force
    d. None of these

16. Which one of the layers handles the task of data compression?
    a. Transport Layer
    b. Data Link Layer
    c. Presentation Layer
    d. Application Layer

17. The sequence of layers in the OSI model in a descending order is _____.
    a. Network, Data Link, Physical, Application, Presentation, Session, Transport
    b. Session, Transport, Presentation, Application, Network, Data Link, hysical

c. Presentation, Application, Session, Transport, Network, Data Link, Physical
d. Application, Presentation, Session, Transport, Network, Data Link, Physical

18. How many layers does the OSI model have?
    a. 7
    b. 6
    c. 5
    d. 8

19. A _____ is a software version of a physical terminal and allows a use to log on to a remote host.
    a. Network Virtual Terminator
    b. Network Virtual Terminal
    c. Network Virtual Transmission
    d. None of these

20. Flow control and error handling are done at the_____ layer.
    a. Transport Layer
    b. Data Link Layer
    c. Presentation Layer
    d. Application Layer

21. The basic function of the network layer is _____.
    a. Fragmentation
    b. Routing
    c. Encryption
    d. Insertion of Checkpoints

22. Which layer of the OSI model ensures an undisturbed connection between two nodes during data exchange?
    a. Application layer
    b. Data link layer
    c. Session layer
    d. Presentation layer

23. If your machine in the network is connected to another machine in another network that uses a different protocol, then your data flows through a _____.
    a. Gateway
    b. Network Interface Unit
    c. Hub
    d. File server

24. Which area of the ISO Network management Model is responsible for identifying problems, logging reports and notifying the users, so that the network runs effectively?
    a. Performance Management
    b. Accounting Management
    c. Fault Management
    d. Configuration Management

Answers :

1 c	2 b	3 a	4 d	5 a	6 c
7 b	8 d	9 a	10 d	11 a	12 b
13 c	14 b	15 b	16 c	17 d	18 a
19 b	20 b	21 b	22 c	23 a	24 c

4 TCP/IP and Internet

Learning Objectives

To understand the :

- Historical view of the Internet and TCP/IP.
- Generic Top level Domains.
- Architecture of TCP/IP Protocol suite.
- Comparative working of OSI and TCP/IP.
- Overview of Internet Protocol Addressing Scheme.
- Concept of Domain Name Systems.
- Various Internet Services.
- Client-Server Architecture.
- Intrusion Detection Systems.

Introduction

In the years to come information, especially the Internet, will become the basis for personal, economic, and political advancement. A popular name for the Internet is the information superhighway. Whether we want to find the latest financial news, browse through library catalogs, exchange information with colleagues, or join in a lively political debate, catch up with our mails, go shopping, banking or do business, the Internet is the tool that takes us beyond telephones, faxes, and isolated computers to a burgeoning networked information frontier.

The Internet supplements the traditional tools of gathering information: Data Graphics, News and correspondence with other people. Used skillfully, the Internet brings information, expertise, and knowledge on nearly every subject imaginable straight to our computer.

History of Internet and TCP/IP

The Internet is a worldwide system of interconnected computer networks. Its origins can be traced to the creation of **ARPANET (Advanced Research Projects Agency Network)** as a network of computers under the auspices of the U.S. Department of Defense in 1969. ARPA established a packet-switching network of computers linked by point-to-point leased lines that provided a basis for early research into networking.

The conventions developed by ARPA to specify how individual computers could communicate across that network became **TCP/IP**.

As networking possibilities grew to include other types of links and devices, ARPA adapted new TCP/IP to meet the demands of the new technology. As involvement in TCP/IP grew, the scope of ARPANET expanded to become what is now known as the **Internet**.

The Internet is a combination of several technologies and an electronic version of newspapers, magazines, books, catalogs, bulletin boards, and much more. Today, the Internet connects millions of computers around the world in a nonhierarchical manner unprecedented in the history of communications. It is a product of the convergence of media, computers, and telecommunications. It is not merely a technological development but the product of social and political processes.. From its origins in a non-industrial, non-corporate environment and in a purely scientific culture, it has quickly diffused into the world of commerce.

An Internet under TCP/IP operates like a single network connecting many computers of any size, type and shape. Internally, an Internet is an interconnection of independent physical networks linked together by internetworking devices.

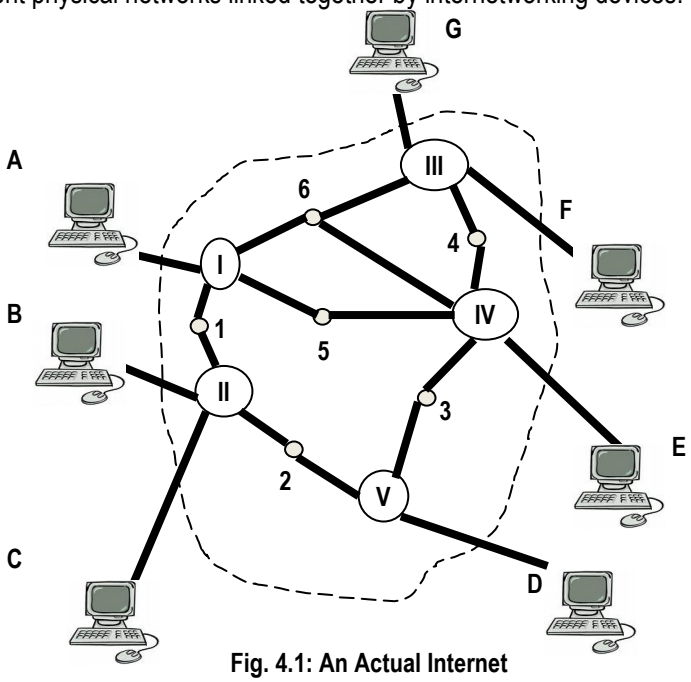


Fig. 4.1: An Actual Internet

Fig. 4.1 shows the topology of a possible internet with A, B, C, D, E, F and G be the hosts. The solid circles in the figure, numbered 1, 2, 3 and so on are internetworking devices like routers or gateways. The larger ovals containing roman numerals (I, II, III etc.) represent separate physical networks.

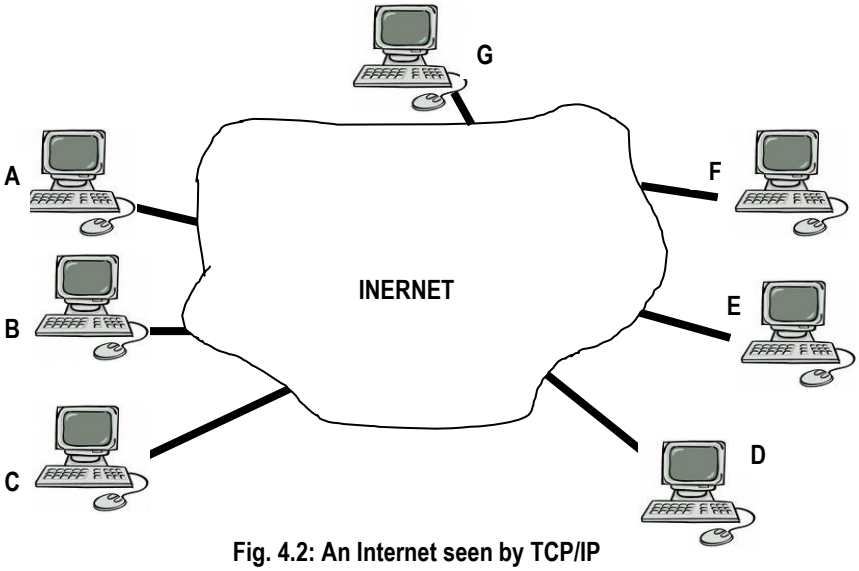


Fig. 4.2: An Internet seen by TCP/IP

Who was the first to use the Internet?

Charley Kline at UCLA sent the first packets on ARPANet as he tried to connect to Stanford Research Institute on Oct 29, 1969. The system crashed as he reached the G in LOGIN!

To TCP/IP, the same internet appears quite differently as shown in Fig. 4.2. TCP/IP considers all interconnected physical networks as one huge network. It allows different computers with heterogeneous technologies, geographically dispersed, to connect and talk to each other. It connects all the parts to this larger logical network rather than to their individual physical networks.

Internet Administration

The specialty of Internet is that it has no single owner and no central operator. Everyone operates a portion of the Internet. There is no central control and regulation. Everyone tries to “regulate” a portion of the Internet.

However, some bodies help in managing the Internet. These are:

- The **Internet Society (ISOC)**, (www.isoc.org) a non-governmental international organization providing coordination for the Internet, and its internetworking technologies and applications.
- The **Internet Activities Board (IAB)** (www.iab.org) governs administrative and technical activities on the Internet.
- The **Internet Engineering Task Force (IETF)** (www.ietf.org) has the primary responsibility for the technical activities of the Internet, including writing specifications and protocols.

Delhi was the first national commercial online service to offer Internet access to its subscribers. It opened up an email connection in July 1992 and full Internet service in November 1992. The first email spam was sent by Digital Equipment Corporation's marketing manager Gary Thuerk in 1978 to 393 recipients on ARPANET.

- The **Internet Research Task Force (IRTF)** (www.irtf.org) helps in promoting research of importance for the evolution of the Internet.
- The **Internet Engineering Planning Group (IEPG)** (www.iepg.org) coordinates worldwide Internet operations. The group also assists Internet Service Providers (ISPs) to interoperate within the global Internet.
- The **Forum of Incident Response and Security Teams** (www.first.org) is the coordinator of a number of Computer Emergency Response Teams (CERTs) representing many countries, governmental agencies, and ISPs throughout the world.
- The **World Wide Web Consortium (W3C)** (www.w3.org) takes a lead role in developing common protocols for the World Wide Web to promote its evolution and ensure its interoperability.
- In the Internet, every host follows a hierarchical naming structure comprising a top-level domain (TLD), domain and sub domain (optional), and host name. The body that handles governance of **global Top Level Domain (gTLD)** registrations is the **Internet Corporation for Assigned Names and Numbers (ICANN)**. (www.icann.org)

Archie is an information system offering an electronic directory service for locating information residing on anonymous FTP sites.

Transmission Control Protocol / Internetworking Protocol (TCP /IP)

A protocol is an agreed-upon set of conventions that defines the rules of communication. The Internet's technological success depends on its principal

communication tools, the **Transmission Control Protocol (TCP**) and the **Internet Protocol (IP)**, generally known as **TCP/IP**.

TCIP is a set of protocols developed to allow cooperating computers to share resources across the networks. It establishes the standards and rules by which messages are sent through the networks. The most important traditional TCP/IP services are : **File Transfer, Remote Login,** and **Mail Transfer**.

- **File Transfer Protocol (FTP):** The **file transfer protocol (FTP)** allows a user on any computer to receive files from or send files to another computer. Security is handled by requiring the user to specify a user name and password for other computer.
- **Remote Login:** The **terminal network** protocol **(TELNET)** allows a user to log in on any other computer on the network. The user starts a remote session by specifying a computer to which it wants to connect. From that time until the end of the session, anything that the user types is sent to the other computer.
- **Mail Transfer:** This allows a user to send messages to users on other computers. Originally, people used only one or two specific computers. They would maintain "mail files" on those machines. The computer mail system is simply a method for a user to add a message to another user's mail file.

Other important services are resource sharing, diskless workstations, computer conferencing, transaction processing, security, multimedia access, and directory services.

TCP/IP Protocol Suite Architecture

TCP/IP (Transmission Control Protocol/Internet Protocol) is a communication protocol developed under contract from the U.S. Department of Defense to internetwork dissimilar systems. Invented by Vinton Cerf and Bob Kahn, this de facto UNIX standard is the protocol of the Internet and the global standard for local area networks and wide area networks, the major exception being the traditional networks of the telephone companies. However, telephone companies that deploy voice over IP (VoIP) networks are, in fact, using TCP/IP as well.

TCP/IP protocol suite is a bundle of protocols that are segmented into five layers. The layers and the protocols within the five layers are given in Fig. 4.3.

I. Application Layer

The **Application layer** is the topmost layer of the TCP/IP protocol suite and runs various applications which provide them the ability to access the services of the other

layers and define the protocols that applications use to exchange data. There are many Application layer protocols, and new ones are always being developed.

Protocols used in Application Layer

a. SMTP : It stands for **Simple Mail Transfer Protocol** which provides a mechanism for sending messages to other computer users based on e-mail addresses. SMTP provides for e-mail exchanges between users on the same or different computers and supports :

- Sending a single message to one or more recipients.
- Sending messages that include text, voice, video, or graphics.
- Sending messages to users on networks outside the Internet.

b. TELNET : It is an abbreviation of **TE**rmina**L NET**work and a general - purpose client-server application program. It establishes a connection to a remote system in such a way that the local terminal appears like a terminal at the remote system. For example, a user wants to run different application programs at a remote site and create results that can be transferred to their local site. One way to satisfy these demands is to create client-server application programs for each desired service, which is not feasible.

A better solution is a general-purpose client-server program that lets a user access any application program on a remote computer. After logging on, a user can use the services available on the remote computer and transfer the results back to the local computer.

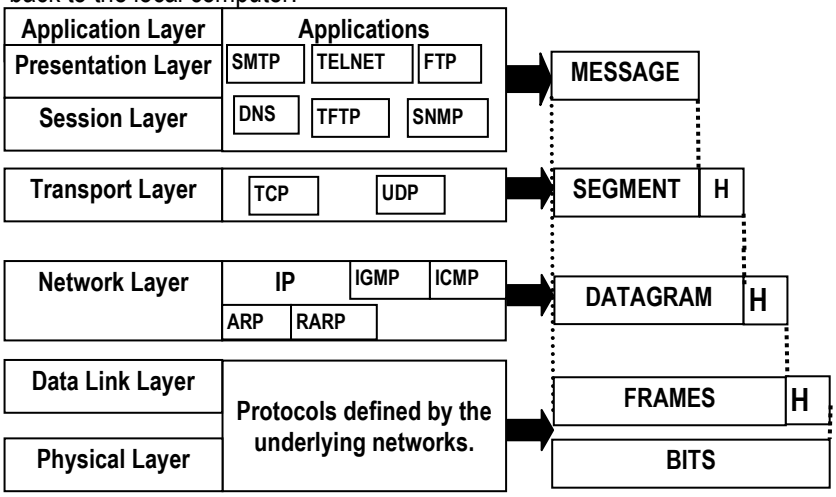


Fig. 4.3: TCP/IP and OSI Model

c. **FTP (File Transfer Protocol) :** It is a standard mechanism provided by the TCP/IP for copying a file from one host to another. Transferring files from one computer to another is one of the most common tasks performed in a networking or internetworking environment. It is because two systems may use different file naming conventions, different directory structures, and represent text and data differently.

   FTP establishes two connections between the hosts : **Data Transfer** and **Control Information (Commands and Responses)**. Separation of command and data transfer makes FTP highly efficient. The control connection remains on during the entire interactive FTP session. It opens when commands that involve transferring files are used, and it closes when the file is transferred.

d. **DNS : Domain Name System** maps a name to an address and an address to a name. The addressing system on the Internet generates IP addresses, which are usually indicated by numbers such as 128.201.86.290. Since such numbers are difficult to remember, a user-friendly system that has been created is known as the Domain Name System (DNS). It provides the mnemonic equivalent of a numeric IP address and ensures that every site on the Internet has a unique address. For example, an Internet address might appear as crito.uci.edu. If this address is accessed through a Web browser, it is referred to as a URL (Uniform Resource Locator), and will appear as http://www.isa.icai.org

   The Domain Name System divides the Internet into a series of component networks called Domains that enable e-mail and other files to be sent across the entire Internet. Each site attached to the Internet belongs to one of the domains. Universities, for example, belong to the edu domain. Other domains are gov (government), com (commercial organizations), mil (military), net (network service providers), and org (nonprofit organizations).

e. **TFTP : Trivial File Transfer Protocol** is designed for occasions when there is need to simply copy a file. For example, when a diskless workstation or a router is booted, we need to download the bootstrap and configuration files, which do not involve the kind of sophistication provided by the FTP. TFTP is so simple that it can fit into a read-only memory of a diskless workstation and can be used at a bootstrap time. TFTP can read or write a file for the client. **Reading** means copying a file from the server site to the client site, and **writing** means copying a file from the client site to the server site.

f. **SNMP :** It stands for **Simple Network Management Protocol,** a framework for managing devices in an internet using the TCP/IP protocol suite. It provides a set of fundamental operations for monitoring and maintaining a network and works

on the concept of manager and agent. A manager, usually a host, controls and monitors a set of agents, usually routers.

Its an application – level protocol ion which a few manager stations control a set of agents Still not clear. SNMP frees management tasks from both the physical characteristics of the managed devices and the underlying networking technology. It can be used in a heterogeneous internet made of different LANs and WANs connected by routers or gateways made by different manufacturers.

II. Transport Layer

The **Transport layer** (also known as **Host-to-Host Transport layer**) is responsible for providing the Application layer with session and datagram communication services. The core protocols of the Transport layer are **Transmission Control Protocol (TCP)** and the **User Datagram Protocol (UDP)**.

Protocols used in Transport Layer

a. User Datagram Protocol (UDP) : UDP provides a one – to - one or one – to - many, connectionless, unreliable communications service. It is used when the data to be transferred is small (such as the data that would fit into a single packet), and the overhead of establishing a TCP connection is not desired or the applications or upper layer protocols provide reliable delivery. It is the simpler of the two standard TCP/IP transport protocols that acts as an end-to-end transport level protocol and adds only port addresses, checksum error controls, and length information to the data from the upper layer. The packet produced by the UDP is called a **User Datagram**. It does not provide any sequencing or reordering functions nor does it specify the damage packet when reporting an error, and is thus unreliable. UDP can discover that an error has occurred on the basis of checksum, but does not contain an ID or sequencing number for which a particular data segment has got corrupted.

If the quantity of the data being transmitted is small, the overhead of creating connections and ensuring reliable delivery may be higher than the work of retransmitting the entire data set in case of failure of delivery. In such situations, programmers prefer UDP.

b. Transmission Control Protocol (TCP) : TCP provides a one – to - one, connection-oriented, reliable communications service. It is responsible for the establishment of a TCP connection, the sequencing and acknowledgment of packets sent, and the recovery of packets lost during transmission. It is responsible also for breaking up the message into datagrams, reassembling

datagrams at the other end, resending anything that gets lost, and putting things back in the right order.

IP is responsible for routing individual datagrams, which are identified by a unique sequence number to facilitate reassembly in the correct order. The whole process of transmission is done through routers. Routing is the process by which two communication stations find and use the optimum path across any network of any complexity. Routers support fragmentation, and have the ability to subdivide received information into smaller units where this is required to match the underlying network technology. These operate by recognizing that a particular network number relates to a specific area within the interconnected networks. They keep track of the numbers during the entire process.

TCP is reliable because there is a mechanism in it called **PAR (Positive Acknowledgment with Re-transmission),** which sends the data to the recipient again and again until it receives a **Data OK** signal from the recipient. This protocol is connection-oriented because it establishes a handshake to exchange data between the communicating parties.

Reliable and Unreliable Transport Methods

The TCP/IP suite provides two transport methods : TCP ensures that data arrives intact and complete, while UDP just sends out packets. TCP is used for everything that must arrive in perfect form, and UDP is used for streaming media, VoIP and videoconferencing, where there is no time to retransmit erroneous or dropped packets in real time.

III. Network Layer

Protocols used in Network Layer

At the network layer, TCP/IP supports the internetwork protocol (IP) which contains four supporting protocols : ARP, RARP, ICMP and IGMP.

a. Internetwork Protocol (IP) : It is an unreliable and connectionless datagram protocol which provides no error checking or tracking. IP assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees. That is why IP has to be paired with a reliable protocol like TCP to provide a secure service. This is like the delivery service of a post office. The post office does its best to deliver mail but does not always succeed. If an unregistered letter is lost, it is up to the sender or would - be recipient to discover the loss. The post office itself does not keep track of every letter and cannot notify a sender of loss or damage. When the letter is delivered, the receiver mails the postcard back to the sender to indicate

that he got it. If the sender does not receive the postcard, he or she assumes the letter has been lost and sends another one.

IP transports data in packets called **Datagrams**, each of which is transported separately. Datagrams may take different routes and may arrive out of sequence or get duplicated, but IP does not keep track of the routes and has no facility for reordering datagrams once they arrive. As it is a connectionless service, IP does not create virtual circuits for delivery. There is no call set up to alert the receiver about an incoming transmission.

b. ARP: Address Resolution Protocol associates an IP address with the physical address which identifies each device on a LAN and is usually imprinted on the **Network Interface Card.** By changing the NIC, in case the card fails, the physical address of a machine can be altered. The IP addresses, on the other hand, have universal jurisdiction and cannot be changed. ARP is used to locate the physical address of the device when its Internet Address is known. Anytime a host, or a router, needs to find the physical address of another host on its network, it formats an ARP query packet that includes IP address and broadcasts it over the network. Every host on the network receives and processes the APR packet, but only the intended recipient recognizes its internet address and sends it back to its physical address, the host holding the datagram adds the address of the target host both to its cache memory and to the datagram header, and then sends the datagram on its way.

c. RARP: The **Reverse Address Resolution Protocol** allows a host to discover its internet address if it knows its physical address. Usually a host has its internet address stored in the hard disk, but diskless computers cannot have that. So the host wishing to know its internet address broadcasts an RARP query packet that contains its physical address to every host on its physical network. A server of the network recognizes the RARP packet and returns it the host's internet address.

d. ICMP : The **Internet Control Message Protocol** is a mechanism used by hosts and routers to send notification of datagram problems back to the sender. ICMP allows IP to inform a sender that a datagram is undeliverable. A datagram travels from router to router until it reaches one that can deliver it to the final destination. If a router is unable to route or deliver the datagram because of unusual conditions or because of network congestion, ICMP allows it to inform the original source.

e. IGMP : The **Internet Group Message Protocol** is a companion to the IP Protocol. IP involves two types of communication : **Unicasting**, in which there is

one-to-one communication and **Multicasting**, in which the same message is transmitted to a large number of receivers simultaneously. IGMP helps a multicast router identify the hosts in a LAN that are members of a multicast group.

IP Makes It Routable

TCP/IP is a routable protocol, and the IP "network" layer in TCP/IP provides this capability. The header prefixed to an IP packet contains not only source and destination addresses of the hosts, but source and destination addresses of the networks they reside in. Data transmitted using TCP/IP can be sent to multiple networks within an organization or around the globe via the Internet, the world's largest TCP/IP network.

The IP Identifies Everything

Every node in a TCP/IP network requires an IP address which is either permanently assigned or dynamically assigned at the startup. Before presenting the architecture of TCP/IP suite, it is necessary to know why TCP/IP is popular and widely adopted. The reasons are:

- Open protocol standards, freely available, and supported by developers, are hardware, software and network independent.
- TCP/IP can be run over an Ethernet, a DSL connection, a dial-up line, an optical network, and virtually on any other kind of physical transmission medium.
- There exists a common addressing scheme that allows any TCP/IP device to uniquely address any other device in the entire network, even if the network is as large as the worldwide Internet.

IV. Data Link Layer

The **Data Link Layer** transforms the physical layer, a raw transmission facility, into a reliable link and is responsible for node - to - node delivery. It makes the physical layer appear error free to the upper layer.

Specific responsibilities of the DLL include the following:

i. **Framing :** The DLL divides the stream of bits received from the network layer into manageable data units called frames.
ii. **Physical Addressing :** If frames are to be distributed to different systems on the network, the DLL adds a header to the frame to define the physical address of the sender (source address) and/or receiver (destination address) of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects one network to the next.

iii. **Flow Control :** If the rate at which the data are absorbed by the receiver is less than the rate produced in the sender, the DLL imposes a flow control mechanism to prevent overwhelming the receiver.
iv. **Error control :** The DLL adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to prevent duplication of frames. Error control is normally achieved through a trailer added to the end of the frame.
v. **Access Control :** When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

V. Physical Layer

The physical layer coordinates the functions that are required to transmit a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the interface and transmission medium. It also defines the procedures and functions that physical devices and interfaces have to perform for the transmission. The physical layer is concerned with the following:

i. **Physical characteristics of interfaces and media:** The physical layer defines characteristics of the interface between the devices and the transmission medium and also defines the type of transmission medium.
ii. **Representation of bits:** The physical layer data consists of a stream of bits (0 and 1) without any interpretation. To be transmitted, bits must be encoded into electrical or optical signals.
iii. **Data Rate:** The transmission rate – the number of bits sent each second, is also defined by the physical layer.
iv. **Synchronization of bits:** The sender and receiver must be synchronized at the bit level.
v. **Line Configuration:** The physical layer is concerned with the connection of devices to the medium.
vi. **Physical Topology:** The physical topology defines how devices are connected to make a network like a mesh, star, ring topology, etc.
vii. **Transmission Mode:** The physical layer also defines the direction of transmission between the two devices: simplex, half-duplex and duplex.

PROTOCOL STACK

Protocol stack is defined as the set of protocols used in a communications network. A protocol stack is a prescribed hierarchy of software layers, starting from the application layer at the top (the source of the data being sent) to the data link layer at the bottom (transmitting the bits on the wire). The stack resides in each client and

server, and the layered approach lets different protocols be swapped in and out to accommodate different architecture networks, as shown in Fig. 4.4.

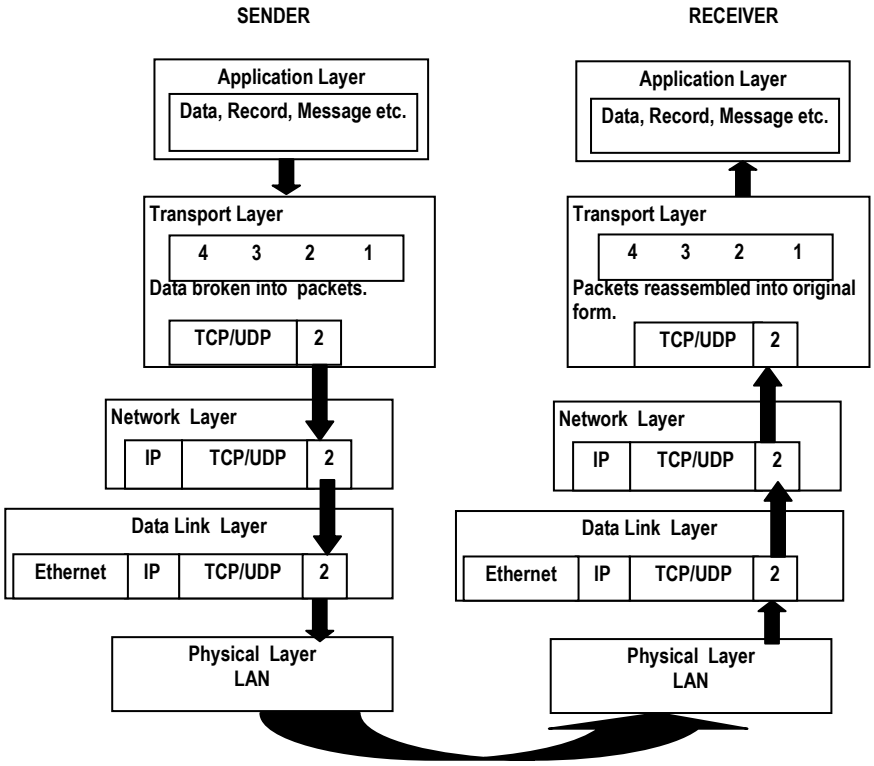


Fig. 4.4 : Protocol Stack of TCP/IP

Telnet is one of the most sensitive and oldest forms of Internet connections for connecting remote machines. Even today it is popular for establishing remote connections. Hackers like it because it helps them to gain remote access to and control over servers located even behind firewalls. A hacker could also hijack an authentic telnet session that is in progress.

Comparison between the OSI model and TCP/IP protocol suite

Transmission Control Protocol (TCP) was designed before the OSI Model, because of which the layers in the TCP/IP protocol do not match exactly with those in the OSI model. The TCP/IP protocol is made of five layers: physical, data link,

network, transport and application. Fig. 4.5 shows the pictorial comparison of the OSI and TCP/IP.

a. The **application layer** in TCP/IP can be equated with the combination of session, presentation, and application layers of the OSI model.
b. At the **transport layer**, TCP/IP defines two protocols: **TCP** and **User Datagram Protocol (UDP)**.
c. At the **network layer**, the main protocol defined by TCP/IP is Internetworking Protocol (IP), although there are some other protocols that support data movement in this layer.
d. At the **physical layer** and **data link layers**, TCP/IP does not define any specific protocol. It supports all the standard and proprietary protocols. A network in a TCP/IP internetwork can be a local area network (LAN), a metropolitan area network (MAN) or a wide area network (WAN).

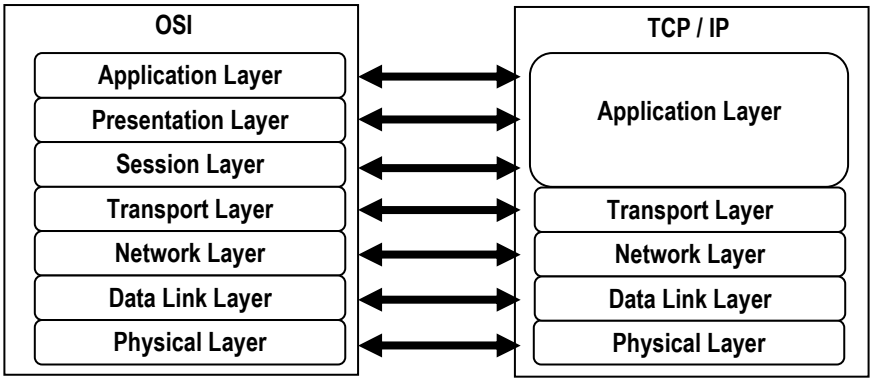


Fig. 4.5: OSI Model vs TCP/IP

TCP breaks down and reassembles packets, whereas IP is responsible for ensuring that the packets are sent to the right destination. Data travels across the Internet through several levels of networks until it reaches its destination. E-mail messages arrive at the mail server (similar to the local post office) from a remote personal computer connected by a modem, or a node on a local-area network. From the server, the messages pass through a router, a special-purpose computer ensuring that each message is sent to its correct destination. A message may pass through several networks to reach its destination. Each network has its own router that determines how best to move the message closer to its destination, taking into account the traffic on the network. A message passes from one network to the next, until it arrives at the destination network.

IP Addressing scheme

Internet Protocol Addressing Scheme gives the address of a device attached to an IP network (TCP/IP network). Every client, server and network device are assigned an IP address, and every IP packet traversing an IP network contains a source IP address and a destination IP address.

Every IP address that is exposed to the public Internet is unique. In contrast, IP addresses within a local network use the same private addresses. Thus a user's computer in company A can have the same address as a user in company B and thousands of other companies. However, private IP addresses are not reachable from the outside world.

Logical vs Physical : An IP address is a logical address that is assigned by software residing in a server or router. In order to locate a device in the network, the logical IP address is converted to a physical address by a function within the TCP/IP protocol software. The physical address is actually built into the hardware.

Static and Dynamic IP : Network infrastructure devices such as servers, routers and firewalls are typically assigned permanent "static" IP addresses. The client machines can also be assigned static IPs by a network administrator, but most often are automatically assigned temporary "dynamic" IP addresses via software that uses the "dynamic host configuration protocol". Cable and DSL modems typically use dynamic IP with a new IP address assigned to the modem each time it is rebooted.

The Dotted Decimal Address : x.x.x.x

Although the next version of the IP protocol offers essentially an unlimited number of unique IP addresses, the traditional IP addressing system named **Internet Protocol Version 4 (IPv4)** uses a smaller 32-bit number that is split into three fields : **class type**, **netid**, and **hostid**. These parts are of varying lengths, depending on the class of the address. IPv4 addresses are usually represented in "dotted decimal" notation (four numbers, each ranging from 0 to 255), separated by decimal points; for example, 204.171.64.2. Each part represents 8 bits of the address, and is therefore called an **octet**. Instead of the domain name of a Web site, the actual IP address can be entered into the browser. However, the Domain Name System (DNS) allows the users to enter computerlanguage.com instead of an IP address, and the domain (the URL) computerlanguage.com is converted to the numeric IP address.

IPv4 Networks

In the early stages of development of the Internet protocol, network administrators interpreted an IP address as a structure of network number and host number. The

highest order octet (most significant eight bits) was designated the **network number** and the rest of the bits were called the **rest field or host identifier** and were used for host numbering within a network. This method soon proved inadequate as additional networks developed that were independent of the existing networks that had already a designated network number. In 1981, the Internet addressing specification was revised with the introduction of classful network architecture.

Classes

Classful network design allowed for a larger number of individual network assignments. The first three bits of the most significant octet of an IP address was defined as the **class** of the address. Three classes (**A, B, and C**) were defined for universal **unicast** addressing. Depending on the class derived, the network identification was based on octet boundary segments of the entire address. Each class used successively additional octets in the network identifier, thus reducing the possible number of hosts in the higher order classes (B and C). The following table gives an overview of this system. There are currently five different field-length patterns in use, each defining a **class of address**. The different classes are designed to cover the needs of different types of organizations.

Classes A, B, C, D and E

Based on the split of the 32 bits, an IP address is Class A, B or C, the most common of which is Class C.

- **Class A** addresses are numerically the lowest and provide only one byte to identify class type and netid, and leaves three bytes available for hostid numbers.
- **Class B** provides two bytes to identify class types and leaves remaining two bytes available for hostid numbers.
- **Class C** provides three bytes to identify class types and leaves remaining one byte available for hostid numbers.
- **Class D** is reserved for multicast addressing. Multicasting allows copies of a datagram to be passed to a select group of hosts rather than to an individual host. It is similar to broadcasting, but broadcasting requires that a packet be passed to all possible destinations, and multicasting allows transmission to a selected subset.
- **Class E** addresses are reserved for future use.

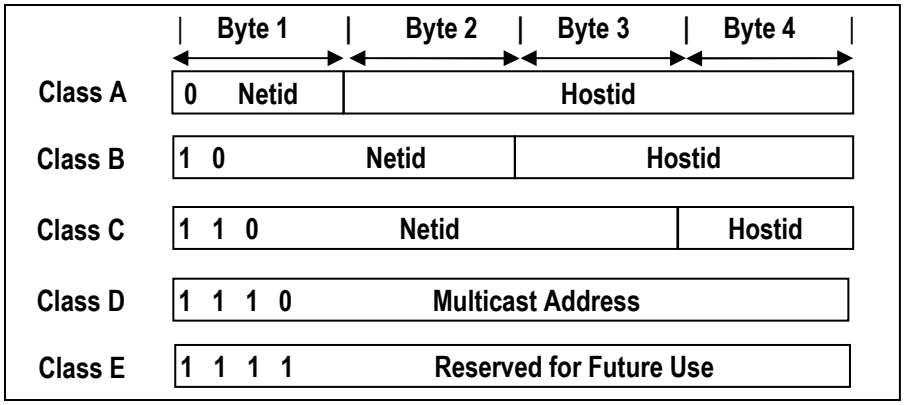


Fig. 4.6: Internet Classes

Class division shows that class A networks can accommodate far more hosts than class B or class C networks. More than two million Class C addresses are assigned, quite often in large blocks to network access providers for use by their customers. The fewest are Class A networks, which are reserved for government agencies and huge companies.

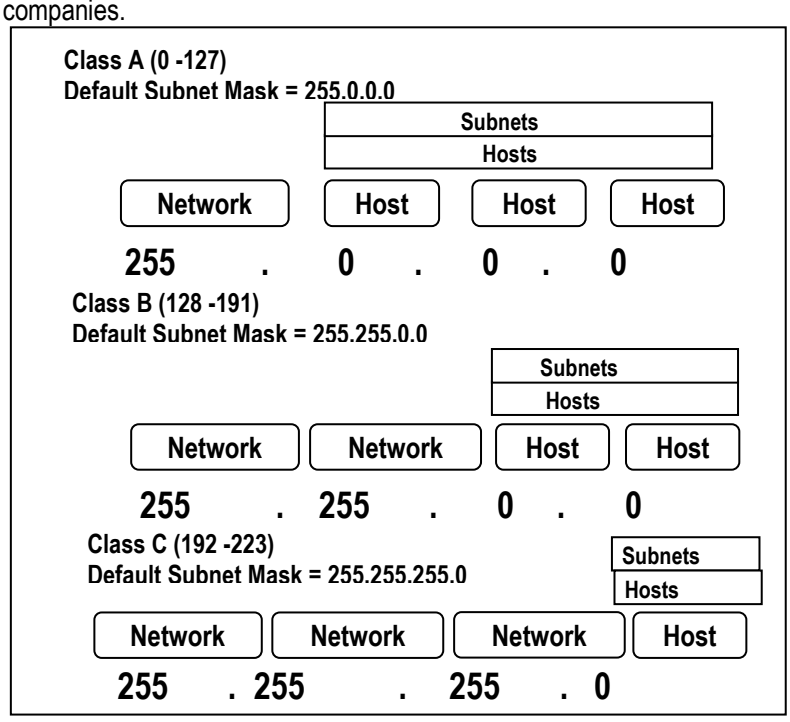


Fig. 4.7 : Classes Divisions

Although people identify the class by the first number in the IP address (as shown in Table 4.1), a computer identifies class by the first three bits of the IP address (A=0; B=10; C=110). This class system (a.b.c.d) has also been greatly expanded, eliminating the huge disparity in the number of hosts that each class can accommodate.

Private Addresses

There are certain blocks of IP addresses that are set aside for internal private use for computers and are not directly connected to the Internet. InterNIC has reserved these IP addresses as private addresses for use with internal web sites or intranets. Most ISPs will block the attempt to address these IP addresses. These IP addresses are used for internal use by companies that need to use TCP/IP but do not want to be directly visible on the Internet. These IP ranges are given in Table 4.1.

Class	Range of first octet	Private Start Address	Private End Address	Number of IP addresses
A	0 - 127	10.0.0.0	10.255.255.255	1, 67, 77,216
B	128 - 191	172.16.0.0	172.31.255.255	10, 48,576
C	192 - 223	192.168.0.0	192.168.255.255	65,536

Table 4.1: Classes and their Ranges

Reserved addresses

Although the previous table presents the IP address classes, still some addresses are missing. For example, the range of IP address from 127.0.0.0 to 127.255.255.255 is not there. This is because several address values are reserved and have a special meaning. The following are reserved addresses:

- IP address 0.0.0.0 refers to the default network and is generally used for routing.
- IP address 255.255.255.255 is called the **Broadcast address**.
- IP address 127.0.0.1 is called the **Loopback address**.

Even with Classless IP, not every IP address is usable. Addresses with a first byte greater than 223 are reserved for Multi-cast and other special applications. There are also two Class A networks that are reserved for special purposes. The network address 0.0.0.0 designates a default gateway. This is used in routing tables to represent "All Other Network Addresses".

Another special Class A network is 127.0.0.0, the loopback address, is used to simplify programming, testing, and troubleshooting. It allows applications to

communicate with the local host in the same manner as communicating with a remote host, without the need to look up the local address.

There are other addresses that are used for special purposes.

- In every network the host address which is all Zeros identifies the network itself. This is called the Network Address and is used in routing tables to refer to the whole network.
- A host address which is all Ones is called the Broadcast Address or Announce Address for that network. For example, on the Class C network 205.217.146.0, the address 205.217.146.255 is the broadcast address.

Packets addressed to the broadcast address will be received by all hosts on that network. The network address and the broadcast address are not used as actual host addresses. These are invalid addresses on the internet. Routers don't route them.

Internet Protocol v.6

While IP v.4 has been in operation for over two decades now and has been fairly resilient in spite of its shortfalls, problems have started impairing business requirements. The most prominent problem is the limited number of IP addresses available. IP v.6 developed by the IETF will replace IP v.4 and is a significant improvisation to remedy the shortcoming of inadequate IP address.It also offers auto configuration, higher mobility, quality service, security and privacy compatibility, efficient and capable of optimization. Various countries, including India, are under migration to IPv.6. In India, the IPv.6 migration is led by the Telecom Regulatory Authority of India. While there will be no direct costs for migration to IPv.6, organizations will have to get compatible hardware and software.

The Domain Name System

DNS (Domain Name System) is a hierarchical, distributed database that contains mappings of DNS domain names to various types of data, such as Internet Protocol (IP) addresses and is an **Internet Engineering Task Force (IETF)** standard. DNS allows the use of friendly names, such as www.microsoft.com, to easily locate computers and other resources on a TCP/IP-based network.

DNS is a system for converting host names and domain names into IP addresses on the Internet or on local networks that use the TCP/IP protocol. For example, when a Web site address is given to the DNS either by typing a URL in a browser or behind the scenes from one application to another, DNS servers return the IP address of the server associated with that name.

All hosts in the Internet are addressed using dotted representation. Since they are 32 bits in length, almost all users find it difficult to memorize the numeric addresses. For example, it is easier to remember www.yahoo.com than 66.218.71.90.

The Domain Name System (DNS) was created to overcome this problem. It is a distributed database that has the host name and IP address information for all domains on the Internet.

In this hypothetical example, www.caindia.com would be converted into the IP address 204.0.8.51. Without DNS, we would have to type the four numbers and dots into our browser to retrieve the Website.

When we want to obtain a host's IP address based upon the host's name, a DNS request is made by the initial host to a local name server. If the information is available in the local name server, it is returned else the local name server then forwards the request to one of the root servers. The root server, then, returns the IP address. This is depicted in Fig. 4.8.

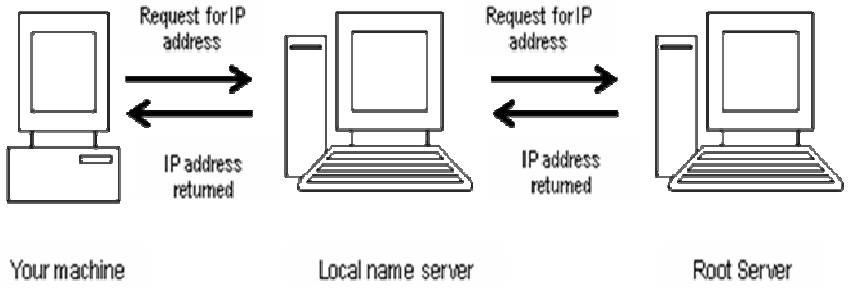


Fig. 4.8: Obtaining the numeric IP address.

The DNS system is a hierarchy of database servers that start with the root servers for all the top level domains (.com, .net, etc.). These servers point to authoritative servers residing within ISPs and companies that resolve the host names to complete the name resolution. For example, in **www.caindia.com**, caindia.com is the domain name, and www is the host name. The domain name is the organization's identity on the Web, and the host name is the name of the actual Web server within that domain.

A DNS server is configured with a "zone file" for each domain that contains "resource records." These are of several types; the most common are described below.

DNS and Reverse DNS

The Address a. record associates a domain name with an IP address, which is the primary purpose of the DNS system. The Pointer (PTR) record provides data for

reverse DNS, which is used for logging the domain name and verification purposes. Also called "inverse DNS," the PTR record is an option.

Reverse Domain Name System is the opposite of the standard DNS query. Most of the time, the DNS is queried with a domain name to return the host's IP address. With reverse DNS, also called "inverse DNS," the system is sent an IP address, and the domain name is returned. Reverse DNS is used to log incoming traffic by domain name for statistical purposes. It is also used to prevent spam by checking that the e-mail message is coming from the domain name indicated in the message header. It is only an option and not mandatory in a DNS server.

Ports

A port is a 16-bit number, which along with an IP address forms a socket. Since port numbers are specified by 16-bit numbers, the total number of ports is given by 2^{16} which are equal to 65536 (i.e. from 0 to 65535). Port numbers in the range 0-1023 are known as **Well Known Ports**. Port numbers in the range 1024-49151 are called **Registered Ports**, and these have been publicly defined as a convenience for the Internet community. The remaining numbers, in the range 49152-65535, are called **Dynamic and/or Private Ports** and can be used freely by any client or server.

Table 4.2 lists some of the common port numbers.

Port #	Protocol	Service
7	TCP	echo
9	TCP	discard
21	TCP	ftp-data
23	TCP	telnet
25	TCP	SMTP
43	TCP	whois
53	TCP/UDP	DNS
70	TCP	gopher
79	TCP	finger
80	TCP	http
110	TCP	pop3
123	UDP	ntp
161	UDP	SNMP

179	TCP	bgp
443	TCP	https
520	UDP	rip
1080	TCP	socks
33434	UDP	traceroute

Table 4.2: Some Popular Port Numbers

Because the port number forms part of the packet header, it is readily interpreted not only by the sending and receiving computers, but also by other aspects of the networking infrastructure. Because different services commonly listen to different port numbers, the practice of attempting to connect in sequence to a wide range of services on a single computer is commonly known as Port Scanning; this is usually associated either with malicious cracking attempts or with a search for possible vulnerabilities to help prevent such attacks. Port connection attempts are frequently monitored and logged by computers connected to networks.

Service Ports

A number is assigned to user sessions and server applications in an IP network. The port number resides in the TCP or UDP header of the packet.

Source Ports

The source port, which can be a random number, is assigned to the client and is used to keep track of user sessions. The combination of port number and IP address is called a "socket."

Destination Ports

The destination port is used to route packets on a server to the appropriate network application. For example, port 80 is the standard port number for HTTP traffic, and port 80 packets are processed by a Web server. Destination ports are typically well-known ports (0-1023) for common Internet applications such as HTTP, FTP and SMTP. It can also be a registered port (1024-49151) that vendors use for proprietary applications.

In the TCP and UDP protocols used in computer networking, a port is a special number present in the header of a data packet. Ports are typically used to map data to a particular process running on a computer.

Ports can be readily explained with an analogy: think of IP addresses as the street address of an apartment building, and the port number as the number of a particular

apartment within that building. If a letter (a data packet) is sent to the apartment (IP) without an apartment number (port number) on it, then nobody would know who it is for (which service it is for). In order for the delivery to work, the sender needs to include the apartment number along with the address.

A server used for sending and receiving email may provide both an SMTP (for sending) and a POP3 (for receiving) service; these will be handled by different server processes, and the port number will be used to determine which data is associated with which process.

In both TCP and UDP, each packet header will specify a source port and a destination port, each of which is a 16-bit unsigned integer (i.e. ranging from 0 to 65535) and the source and destination network addresses (IP-numbers), among other things. A process may "bind" to a particular port to send and receive data, meaning that it will listen for incoming packets whose destination the port matches that port number, and/or send outgoing packets whose source port is set to that port number. Processes may also bind to multiple ports.

Source Becomes Destination and Vice Versa

On the return trip response from the server, the destination port number and IP address become the source port number and IP address. Likewise, the source port and IP become the destination port and IP. The software that responds to a port number is said to be "listening" or "looking" for its packets.

The first ever ISP was CompuServe. It still exists, under AOL Time Warner.

Generic top-level domain

A **generic top-level domain (gTLD)** is one of the categories of top-level domains (TLDs) maintained by the **Internet Assigned Numbers Authority (IANA)** for use on the Internet. Overall, IANA currently distinguishes the following groups of top-level domains:

- infrastructure top-level domain (.arpa)
- country - code top-level domains (ccTLD)
- sponsored top - level domains (sTLD)
- generic top - level domains (gTLD)
- generic-restricted top - level domains

The core group of generic top - level domains consists of the .com, .info, .net, and .org domains. In addition, the domains .biz, .name, and .pro are also considered

generic; however, these are designated as generic-restricted, and registrations within them are supposed to require proof of eligibility within the guidelines set for each.

Historically, the groups of generic top - level domains include domains that were created in the early development of the domain name system, notably .edu, .gov, .int, .mil. However, these domains now have all been sponsored by appropriate agencies or organizations and are now considered sponsored top-level domains, much like the many newly created "themed" domain names. This entire group of non-country-code top-level domains, domains that do not have a geographic or country designation, are still often referred by the term generic.

A **generic top - level domain (gTLD)** is a top-level domain used by a particular class of organizations. These are three or more letters long, and are named after the type of organization they represent (for example, .com for commercial organizations). Table 4.3 shows the current gTLDs.

Generic Top Level Domains	
CURRENT	
Generic	.biz , .com , .info , .name , .net , .org , .pro
Sponsored	.aero , .asia , .cat, .coop, .edu, .gov, .int, .jobs, .mil, .mobi , .museum , .tel , .travel
Infrastructure	.arpa
Deleted/retired	.nato
Reserved	.example , .invalid, .localhost, .test
Pseudo	.bitnet , .csnet, .local, .root , .uucp, .onion, .exit
PROPOSED	
Location	.berlin , .lat, .nyc
Language & Nationality	.bzh , .cym , .gal , .lli, .sco
Technical	.geo, .mail
Others	.kids , .post , .shop , .web , .xxx

Table 4.3: Generic Top Level Domains

Internet Services

Internet provides a variety of services. Fig. 4.8 shows only a partial list of services. We can subscribe (while some of them are free, some are not) to these and use the services.

Did you know?

Cyber Squatting is a computer crime that involves criminal minded persons who register or buy out domain names that represent existing trademarks or organization names that have not yet been taken. Some of the organizations troubled by cyber squatters include SBI Cards, Sushmita Sen and Tata Group.

Internet services are provided by Internet Service Providers that employ a range of technologies to enable consumers to connect to their network. For home users and small businesses, the most popular options include dial-up, DSL, broadband wireless, cable modem, and Integrated Services Digital Network (ISDN).

Customers with more demanding requirements, such as medium-to-large businesses, or other ISPs are likely to use DSL, Ethernet, Metro Ethernet, Gigabit Ethernet, Frame Relay, ISDN, ATM, satellite Internet access and Synchronous Optical Networking (SONET).

With the increasing popularity of downloading music and online video and the general demand for faster page loads, higher bandwidth connections are becoming more popular. Typical home user connections are:

- DSL
- Broadband wireless access
- Cable modem
- FTTH
- ISDN

Typical business type connections are:

- DSL
- SHDSL
- Ethernet technologies

When using a dial-up or ISDN connection method, the ISP cannot determine the caller's physical location in detail than by using the number transmitted using an appropriate form of Caller ID. Other means of getting connected such as cable or DSL require a fixed registered connection node, usually associated with the ISP with a physical address.

ISP Interconnection

ISPs may engage in peering, where multiple ISPs interconnect at peering points or Internet exchange points (IXs), allowing routing of data between each network, without charging one another for the data transmitted - data that would otherwise have passed through a third upstream ISP, incurring charges from the upstream ISP.[Consider revising]

Network hardware, software and specifications, as well as the expertise of network management personnel, are important for ensuring that data follows the most efficient route, and upstream connections work reliably. A tradeoff between cost and efficiency is possible.

World Wide Web

The **W**orld **W**ide **W**eb (WWW, also referred to as W3) is based on a technology called Hypertext. The Web may be thought of as a very large subset of the Internet, consisting of hypertext and hypermedia documents. A hypertext document is a document that has a reference (or link) to another hypertext document, which may be on the same computer or in a different computer that may be located anywhere in the world. Hypermedia is a similar concept except that it provides links to graphic, sound, and video files in addition to text files.

In order for the Web to work, every client must be able to display every document from any server. This is accomplished by imposing a set of standards known as a protocol to govern the way the data are transmitted across the Web. Thus data travel from client to server and back through a protocol known as the **H**yper **T**ext **T**ransfer **P**rotocol (http). In order to access the documents that are transmitted through this protocol, a special program known as a browser is required, which browses the Web.

- What makes the World Wide Web appealing and innovative is its use of hypertext as a way of linking documents to each other. A highlighted word or phrase in one document acts as a pointer to another document that amplifies or relates to the first document. In this way, the user tailors the experience to suit his or her needs or interests.
- The other very appealing aspect of the World Wide Web is the use of graphics and sound capabilities. Documents on the WWW include text, still images, videos, and audios. People who create WWW documents often include a photograph of themselves along with detailed professional information and personal interests. (This is often called a person's home page.)

What makes the WWW work?

> **Did you know?**
> The first e-mail spam was sent by Digital Equipment Corporation's marketing manager Gary Thuerk in 1978 to 393 recipients on ARPANET.

WWW is another example of client/server computing. Each time a link is followed, the client requests a document (or graphic or sound file) from a server (also called a Web server) that's part of the World Wide Web that "serves" up the document. The server uses a protocol called HTTP or HyperText Transport Protocol. The standard for creating hypertext documents for the WWW is HyperText Markup Language or HTML. HTML essentially codes plain text documents so that they can be viewed on the Web.

WWW Clients or Browsers

The program used to access the WWW is known as a browser because it "browses" the WWW and requests these hypertext documents. Browsers can be graphical, like Netscape and Mosaic, allowing us to see graphics and hear the audio; text-only browsers (i.e., those with no sound or graphics capability) are also available. All of these programs understand 'http' and other Internet protocols such as FTP, gopher, mail, and news, making the WWW a kind of "one stop shopping" for Internet users.

A look at Search Engines

> **Google** got its name from the mathematical figure googol, which denotes the number 'one followed by a hundred zeros'.

The World Wide Web is "indexed" through the use of search engines, which are also referred to as "spiders," "robots," "crawlers," or "worms". These search engines comb the Web documents, identifying the text that is the basis for keyword searching. Each search engine works in a different way. Some engines scan for information in the title or header of the document; others look at the bold "headings" on the page for their information. The fact that search engines gather information differently means that each will probably yield different results. Therefore, it's wise to try more than one search engine when doing Web searching. Fig. 4.8 displays some of the important Internet Services.

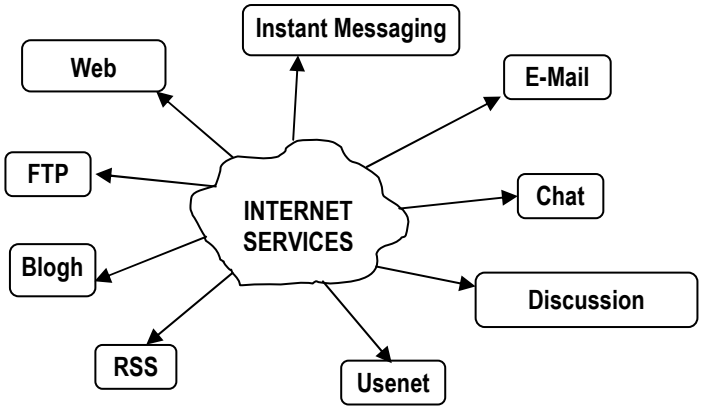


Fig 4.8: Partial List of Internet Services

1. Electronic mail on the internet

Electronic mail, or e-mail, is probably the most popular and widely used Internet function. E-mail, email, or just mail, is a fast and efficient way to communicate with friends or colleagues. One can communicate with one or more persons at a time, and receive and send files and other information, even subscribe to electronic journals and newsletters and send an e-mail message to a person in the same building or on the other side of the world.

How does E-mail Work?

E-mail is an asynchronous form of communication, meaning that the person whom we want to read our message doesn't have to be available at the precise moment we send our message. This is a great convenience for both the sender and the recipient. On the other hand, the telephone, which is a synchronous communication medium, requires the sender and the receiver to be on the line at the same time in order for make communication possible (unless one leaves a voice message). Most of the e-mail programs share a basic functionality which allows the users to:

- send and receive mail messages
- save messages in a file
- print mail messages
- reply to mail messages
- attach a file to a mail message

Reading an Internet Address

To use Internet e-mail successfully, we need to understand how the names and addresses for computers and people on the Internet are formatted. Mastering this

technique is just as important as knowing how to use telephone numbers or postal addresses correctly.

Fortunately, after we get the hang of them, Internet addresses are usually no more complex than phone numbers and postal addresses. Like those methods of identifying a person, an organization, or a geographic location - usually by a telephone number or a street address - Internet addresses have rules and conventions for use.

Sample e-mail Address: *isa@caindia.edu*

The e-mail address has three parts:

1. a user name [isa in the example above]
2. an "at" sign (@)
3. the address of the user's mail server domain [caindia.edu in the example above]

The right-most segment of the domain name usually adheres to the naming conventions listed below:

- EDU Educational sites
- COM Commercial sites
- GOV Government sites
- NET Network administrative organizations
- MIL Military sites
- ORG Organizations not covered by the categories given above (e.g. nonprofit organizations)
- XX where xx is the country code (e.g., for INDIA).

The search engine "**Lycos**" is named after Lycosidae, the Latin name for the wolf spider family. The US International Broadcasting Bureau created a proxy service to allow Chinese, Iraians and other 'oppressed' people to circumvent their national firewalls, relaying forbidden pages behind silicon curtains.

2. FTP (File Transfer Protocol)

FTP is a standard network protocol used to exchange and manipulate files over an Internet Protocol computer network, such as the Internet. It is built on a client-server architecture and utilizes separate control and data connections between clients and server applications.

3. Chatting

Internet Relay Chat (IRC), the other method for Internet conversation, is less common than talk because someone must set up the Chat before others can join in. Chat sessions allow many users to join in the same free-form conversation, usually

centered on a discussion topic. When users see a topic that interests them, they type a command to join and then type another command to choose a nickname. Nicknames allow people in the session to find others on IRC Networks or Channels.

4. RSS (Really Simple Syndication)

RSS is a family of web feed formats used to publish frequently updated works - such as blog entries, news headlines, audio, and video in a standardized format. An RSS document (which is called a "feed", "web feed" or "channel") includes full or summarized text, plus metadata such as publishing dates and authorship. Web feeds benefit publishers by letting them syndicate content automatically. They benefit readers who want to subscribe to timely updates from favored websites or to aggregate feeds from many sites into one place.

5. Usenet

Usenet is a world-wide distributed discussion system that consists of a set of “newsgroups” with names that are classified hierarchically by subject. “Articles” of “messages” are posted to these newsgroups by people on computers with the appropriate software, which are then broadcast to other interconnected computer systems via a wide variety of networks. Some newsgroups are “moderated”, wherein the articles are first sent to a moderator for approval before appearing in the newsgroup. Usenet is available on a wide variety of computer systems and networks, but the bulk of modern Usenet traffic is transported over either the Internet or UUCP.

6. Blog

A **Blog** is a type of website, usually maintained by an individual with regular entries of commentary, descriptions of events, or other material such as graphics or video. A typical blog combines text, images, and links to other blogs, Web pages, and other media related to its topic. The possibility of readers leaving comments in an interactive format is an important part of many blogs.

7. Instant Messaging

Instant messaging (IM) is a form of real-time communication between two or more people based on a typed text. The text is conveyed via devices connected over a network such as the Internet. Instant messaging (IM) is a collection of technologies that create the possibility of real-time text-based communication between two or more participants over the internet or some form of internal network/intranet. The difference between chatting and instant messaging is that an e-mail is the perceived synchronicity of the communication by the user. Some systems allow the sending of messages to people not currently logged on (offline messages), thus reducing differences between Instant Messaging and e-mail.

Commerce on the Internet

Commerce on the Internet is known by a few other names, such as E-business, E-tailing (Electronic retailing), and E-commerce. The strengths of E-business depend on the strengths of the Internet. Internet commerce is divided into two major segments : Business -to-Business (B2B) and Business-to-Consumer (B2C). In each there are some companies that have started their businesses on the Internet, and those that have existed previously and are now transitioning into the Internet world. Some products and services, such as books, compact disks (CDs), computer software, and airline tickets, are particularly suited for online business.

Future of the Internet

What does the future hold for the Internet? Predictions are that in the future nearly every Internet-connected device will communicate wirelessly. Low-power radio cells rather than fiber or copper wire, will connect and relay information. The future appears to hold a wireless Internet because of bandwidth problems with cable or wire.

The personal computer will continue to evolve, but in near future, there will be a great demand of lot of other Internet smart appliances like - Internet wristwatches matching the person with the message, Televisions recording our favorite shows, various kitchen appliances working on Internet commands, automobiles having internal connectivity and carrying a very large cache of favorite music, talk, interactive games, and pictures, while passenger having an option of looking out the window at the real world or looking in the window of his in-car display etc.

The Client/Server Model

The most popular Internet tools operate as client/server systems.

Web Client: A program called a Web client is a software that displays documents and carries out requests. Network clients make requests to a server by sending messages, and the servers respond by acting on each request and returning results. It also sets up a Telnet session, or downloads a file.

Web Server: It is a Web software that provides data, or "serves up" information to the client. One server generally supports numerous clients, and multiple servers can be networked in a pool to handle increased processing load as the number of clients grows.

A client computer and a server computer are usually two separate devices, each customized for their designed purpose. For example, a Web client works best with a large screen display, while a Web server does not need any and can be located

anywhere in the world. However, in some cases a given device can function both as a client and a server for the same application. Likewise, a device that is a server for one application can simultaneously act as a client to others for different applications.

Some of the most popular applications on the Internet follow the client-server model including email, FTP and Web services. Each of these clients features a user interface (either graphic- or text-based) and a client application that allows the user to connect to servers. In the case of email and FTP, users enter a computer name (or sometimes an IP address) into the interface to set up connections to the server.

All the basic Internet tools - including Telnet, FTP, Gopher, and the World Wide Web -are based upon the cooperation of a client and one or more servers. In each case, we interact with the client program and it manages the details of how data is presented to us or the way in which we can look for resources. In turn, the client interacts with one or more servers where the information resides. The server receives a request, processes it, and sends a result, without having to know the details of our computer system, because the client software on our computer system handles those details.

The advantage of the client/server model lies in distributing work so that each tool can focus or specialize on particular tasks : the server serves information to many users while the client software for each user handles the individual user's interface and other details of the requests and results.

Client/server software architecture is a robust, versatile and modular infrastructure that is intended to improve usability, flexibility, interoperability, and scalability.

Before explaining what a Client/Server Software Architecture is, it is essential to know its predecessors.

Mainframes / Machines operating using time-shared concept: With mainframe/ machines with time-shared concept, the entire architecture is dependent on the monolithic central computer. Users interact with the host through a terminal. The keystrokes are captured and sent to its central machine, which does all the processing and displays the results. Modern day mainframes use both dumb terminals as well as PCs. The main limitation of mainframes includes high cost (for both installation and maintenance) and poor support for graphical user interfaces.

Network based architectures / File sharing architecture: The concept of networks followed the mainframes / machines that adopted time-shared concepts. There is a file server whose main job is to share information with the connected machines. It operates on dedicated or non-dedicated mode. In the file sharing mode, users request files from the server, and the server provides the required information.

Client / Server Architecture

A **client** is a requester of services, whereas a **server** is a provider of services. Client / Server architecture of a computer network is the one in which many clients (remote processors) request and receive service from a centralized server (host computer). Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns. Servers wait for requests to arrive from clients and then respond to them. Ideally, a server provides a standardized transparent interface to clients so that clients need not be aware of the specifics of the system (i.e., the hardware and software) that is providing the service. Clients may often be situated at workstations or on personal computers, while servers may be located elsewhere on the network, usually on more powerful machines. This computing model is especially effective when clients and the server each have distinct tasks that they routinely perform. It is an environment in which the application processing is divided between client workstations and servers. It implies the use of desktop computers interacting with servers in a network in contrast to processing everything in a large centralized mainframe.

For example, in a hospital data processing, a client computer can run an application program for entering patient information while the server computer runs another program that manages the database in which the information is permanently stored. Many clients can access the server's information simultaneously, and, at the same time, a client computer can perform other tasks, such as sending e-mail. Because both client and server computers are considered intelligent devices, the client-server model is completely different from the old "mainframe" model, which utilizes a centralized mainframe computer that performs all the tasks for its associated "dumb" terminals.

In this architecture, there is a server and a client. The file server in the Network based architecture is replaced by a database server. To query the database server, database management system (DBMS) software is used. The basic components of a client/server system are:

i. Client
ii. Server
iii. Robust and good communication system
iv. GUI based operating system
v. Open-database Connectivity drivers (ODBC) and Application Programming Interfaces (APIs)

Characteristics of a Client Server System

In a Client/Server system

1. The software architecture is primarily GUI based.

2. The job is shared between the client and the server.
3. The network has to be robust and must have zero failures.
4. A client can attach to any number of servers and a server can support multiple clients.
5. In case of any arbitration, the server's decision is final.

How does a Client Server System work?

1. The user submits a job to the client.
2. The client processes the job and determines what is to be done. This is the client-side part of the job.
3. The job is then transferred to the server via the communication system.
4. Upon receiving the same, the server processes what is asked for. This is the server - side part of the job.
5. The results are passed back to the client via the communication system.
6. The client then performs some tasks on the job to provide the results to the user in an appropriate manner.

This clearly indicates that the job is shared between the server and the client. On account of this, there is reduced network traffic. Remote Procedure Calls (RPCs) or Structured Query Language (SQL) statements are typically used to communicate between the client and server. Fig. 4.9 shows the working of a client server system.

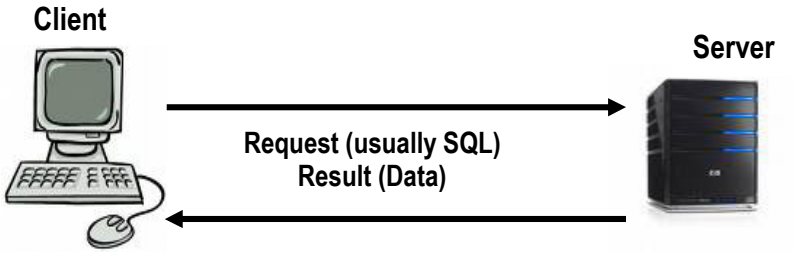


Fig. 4.9: Working of a Client Server System

Client Server architectures are broadly classified into 2-Tier architecture and N-Tier architectures.

Two-tier architecture

A two-tier architecture is one where a client talks directly to a server without any intervening server. It is usually used in small environments (usually less than 50 users). In a two-tier client/server architecture, the user interface (GUI) is in the user's machine and the database management services are usually in the server.

The disadvantages of two-Tier architecture are:

When the number of users exceeds a certain number, performance suffers.

If the implementation of processing management services is through proprietary database products, then there is a restriction with regard to flexibility and choice of DBMS for applications.

Three-tier architecture

Three-tier architecture, also called n-tier architecture, overcomes the problems of the two - tier architecture. In the three-tier architecture, a middle tier (also called as middleware) was added between the client and the server. The middle tier performs the job of queuing, application execution, and database staging. Middleware has been implemented in a variety of ways. The important ones include transaction processing monitors and message servers.

3 - Tier architecture with Transaction Processing (TP) monitor as the middleware This is the most basic type of three-tier architecture. The Transaction Processing monitor provides the services of message queuing, transaction scheduling, and prioritisation. Here the client first connects to the TP monitor instead of the database server. The transaction is accepted by the monitor, which queues it and then takes responsibility for managing it to completion, thus freeing the client.

i. **Three-tier architecture with message server as the middleware.** Here, the middleware prioritises the messages and process them asynchronously. The message server connects to the database server and other data sources.
ii. **Three-tier with ORB (Object Request Broker) Architecture** In order to improve interoperability, Object Request Broker (ORB) architectures have come into play. These days there are two main technologies - Common Object Request Broker Architecture (CORBA) and COM/DCOM by Microsoft.

Industry is working on standards to improve interoperability between CORBA and COM/DCOM. The Object Management Group (OMG) has developed a mapping between CORBA and COM/DCOM that is supported by several products.

Types of Servers

In Client/Server environments, we have the following types of servers which do specific functions. They are:

a. Application
b. File
c. Database

d. Print/fax
e. Communications
f. Security
g. Web

a. Application server: Its major task is to run applications. In large organizations, there are multiple application servers to service their business needs.

b. File server: It supports sharing of files among the needy, and acts as a repository of all data in an organization.

c. Database server: This server manages the database of an organization.

d. Print/Fax server: The main task of Print/Fax server is to receive document/items to be faxed for printing / faxing from the client, and then to prioritize them, queue them, and then execute what is to be done (print/fax). It also intimates the application in case of any errors.

e. Communication server: This server takes care of the communication requirements. One needs this server because of heterogeneity in hardware, software, communication protocols and varying line conditions.

f. Security server: This server takes care of the security issues. As the number of users increase, authenticating each of the users is a resource intensive job. The workload is heavy when other forms of authentication techniques such as biometrics are used for conventional password technique.

g. Web server: As organizations go global, they host their data on the web and do business from there. The Web server takes care of all the issues connected with the web.

Intrusion Detection Systems (IDS)

The dictionary meaning of the word intrusion is “Entry to another's property without right or permission”. With respect to Information Technology, the term intrusion refers to “Unauthorized attempt to access, alter, render unavailable, or destroy information on a system. This also encompasses making efforts to destroy the system itself.”

Intrusion Detection is the process of detecting unauthorized, inappropriate or anomalous activity that is taking place in an Information Technology Environment. A system that helps in intrusion detection is called **Intrusion Detection System**.

IDS is a software and/or hardware designed to detect unwanted attempts at accessing, manipulating, and/or disabling of computer systems, mainly through a network, such as the Internet. These attempts may take the form of attacks, say by crackers, malware and/or disgruntled employees. An IDS cannot directly detect attacks within properly encrypted traffic.

An intrusion detection system is used to detect several types of malicious behaviors that can compromise the security and trust of a computer system. This includes network attacks against vulnerable services, data driven attacks on applications, and host based attacks such as privilege escalation, unauthorized logins and access to sensitive files, and malware (viruses, trojan horses, and worms).

An IDS is composed of several components: **Sensors,** which generate security events, **Consoles,** which monitors events and alerts and control the sensors, and a central **Engine** that records events logged by the sensors in a database and uses a system of rules to generate alerts from security events received. There are several ways to categorize an IDS, depending on the type and location of the sensors and the methodology used by the engine to generate alerts. In many simple IDS implementations all the three components are combined in a single device or appliance.

Classification of intruders

Intruders are broadly classified as :

i. **Outsiders:** These refer to the people who do not belong to the organization but intrude into the system. Typical activities of these people include defacing web pages, forwarding spam mails and making the system unavailable for use. Outsiders generally come from the Internet, dial-up lines, physical break-ins, vendors, customers, resellers, and include all those who are linked to the network.
ii. **Insiders:** These are people who belong to the organization and have legitimate rights to use the resources of the system. They misuse the privileges granted to them or impersonate higher privileged users or create pathways for external intruders to hack into the system. Researches have shown that 80% of security breaches are committed by insiders.

Approaches for intrusion

There are three primary ways by which an intruder breaks into a system:

a. **Physical Intrusion:** In this, the intruder has a physical access to a system and can log-on into the system.
b. **System Intrusion:** In this type the intruder already has a low-privilege user account on the system. If the system does not have a good security mechanism, the intruder uses a known exploit to gain additional administrative privileges.
c. **Remote Intrusion:** This type of intruder penetrates a system through remote machines. This is the most typical and widely adopted method because the intruders need not be physically present at the Information Processing Facility.

Once an intruder has penetrated into the system, his most likely activities are:

a. Copy needed data into a media.
b. E-mail or transmit the needed data to another location.
c. Delete files.
d. Format hard disks.
e. Plant Trojans.
f. Install mechanisms to bypass security procedures.

Terminologies used in Intrusion Detection Systems

- **Alert/Alarm:** A signal suggesting that the system has been or is being attacked.
- **True attack stimulus:** An event that triggers an IDS to produce an alarm and react as though a real attack were in progress.
- **False attack stimulus:** The event signaling an IDS to produce an alarm when no attack has taken place.
- **False (False Positive):** An alert or alarm that is triggered when no actual attack has taken place.
- **False negative:** The failure of an IDS to detect an actual attack.
- **Noise:** Data or interference that can trigger a false positive.
- **Site policy:** Guidelines within an organization that control the rules and configurations of an IDS.
- **Site policy awareness:** The ability an IDS has to dynamically change its rules and configurations in response to changing environmental activity.
- **Confidence value:** A value an organization places on an IDS based on past performance and analysis to help determine its ability to effectively identify an attack.
- **Alarm filtering:** The process of categorizing attack alerts produced from an IDS in order to distinguish false positives from actual attacks.

Types of Intrusion-Detection Systems

1. A **Network Intrusion Detection System (NIDS)** is an independent platform which identifies intrusions by examining network traffic and monitors multiple hosts. This system analyzes data packets that travel over the actual network. These are examined and sometimes compared with empirical data to verify their malicious or benign nature. Because they are responsible for monitoring a network, rather than a single host, Network-based intrusion detection systems (NIDS) tend to be more distributed than host-based IDS. Software, or appliance hardware in some cases, resides in one or more systems connected to a network, and are used to analyze data, such as network packets. Instead of analyzing information that originates and resides on a computer, network-based

IDS uses techniques like “packet-sniffing” to pull data from TCP/IP or other protocol packets traveling along the network. This surveillance of the connections between computers makes network-based IDS great at detecting access attempts from outside the trusted network. In general, network-based systems are best at detecting the following activities:

- **Unauthorized outsider access:** When an unauthorized user logs in successfully, or attempts to log in, s/he can be tracked with host-based IDS. However, detecting the unauthorized users before they log on attempt is best accomplished with network-based IDS.
- **Bandwidth theft/denial of service:** These attacks from outside the network single out network resources for abuse or overload. The packets that initiate/carry these attacks can best be noticed by network-based IDS.

Some possible downsides to network-based IDS include encrypted packet payloads and high-speed networks, both of which inhibit the effectiveness of packet interception and deter packet interpretation. Examples of network-based IDS include Shadow, Snort, Dragon, NFR, RealSecure, and NetProwler.

2. A **Protocol-based Intrusion Detection System (PIDS)** consists of a system or agent that sits at the front end of a server, monitoring and analyzing the communication protocol between a connected device (a user/PC or system) and the server. For a web server this would typically monitor the HTTPS protocol stream and understand the HTTP protocol relative to the web server/system it is trying to protect. Where HTTPS is in use, then this system would need to reside in the "shim", or interface, between where HTTPS is un-encrypted and immediately prior to its entering the Web presentation layer.
3. An **Application Protocol-based Intrusion Detection System (APIDS)** consists of a system or agent that would typically sit within a group of servers, monitoring and analyzing the communication on application specific protocols. For example, in a web server with a database this would monitor the SQL protocol specific to the middleware/business logic as it transacts with the database.
4. A **Host-based Intrusion Detection System (HIDS)** consists of a software agent on a host which monitors all activity of the host on which it is installed. It identifies intrusions by analyzing system calls, application logs, file-system modifications (binaries, password files, capability/acl databases) and other host activities and state. An example of a HIDS is OSSEC.
5. **Signature Based IDS :** The IDS does the job of detecting intrusions by collecting information on the kind of traffic within the internal network and what is happening on host systems and then collate and compare the sequenced action

against the attack signature database to see if there is a match for a known attack signature. This kind of IDS is also known as the Signature based IDS.

6. **Statistical Anomaly IDS**: Some smart IDS can also detect 'unknown' attack patterns using heuristic technology; these are called Statistical Anomaly IDS and use simulation techniques to 'predict' possible attacks.
7. A **Hybrid Intrusion Detection System** combines two or more approaches. Host agent data is combined with network information to form a comprehensive view of the network. An example of a Hybrid IDS is the Prelude.

Approaches for IDS

There are various approaches for handling anti-intrusions and designing IDS. Fig. 4.10 gives a broad overview of the same.

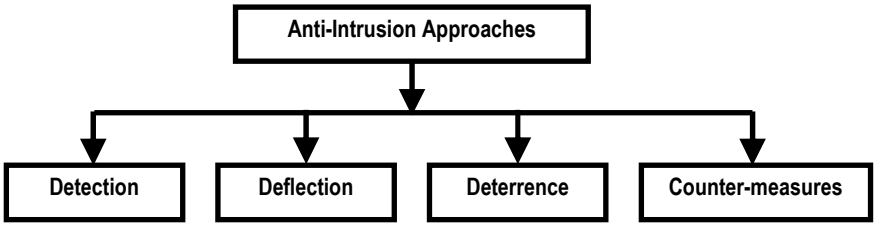


Fig. 4.10: Anti-Intrusion Approaches

An "Intrusion Detection System (IDS)" is a system for detecting intrusions. IDS can be broken down into the following categories:

a. **Network Intrusion Detection Systems (NIDS)** monitors packets on the network and attempts to discover if a hacker/cracker is attempting to break into a system or cause a denial of service attack.
b. **System Integrity Verifiers (SIV)** monitors system files to find if an intruder has opened up a backdoor for attack.
c. **Log File Monitors (LFM)** monitors log files generated by network services and looks for patterns in the log files that suggest an attack.

How do intruders get into systems?

The following are the common ways which any intruder utilizes.

1. Software bugs
   a. Buffer overflows
   b. Unexpected combinations
   c. Unhandled input
   d. Race conditions

2. System configuration bugs
    - Default configurations
3. Poor administrators
4. Exploiting Trust relationships
5. Design flaws

Although a firewall prevents most unwanted access into the network, it is impossible for it to entirely block them from accessing secured networks. The limitation of the firewall is that it is a perimeter defence mechanism. If a packet is successful in by-passing or fooling the firewall, unless controls are in place, it can gain unrestricted access to the internal network and perform any actions on any network connected devices. Hence there is need for a detective mechanism to continuously scan the traffic within the secure network, to identify any unauthorised packets or packets performing unauthorised actions. . A device that performs this function is called the Intrusion Detection System (IDS).

The IDS is somewhat similar to the anti-virus software in its working. While an anti-virus looks for unauthorised or malicious pieces of code sitting at the perimeter of the network or a host, the IDS continuously scans the network for 'attack signatures'. An attack signature is nothing but a sequence of actions that can lead to compromising the security of an information asset. IDS may be a hardware and/or software based system that is designed to detect unwanted attempts at accessing, manipulating, and/or disabling of computer systems, mainly through a network.

The working of IDS

In short, intrusion detection systems detect possible intrusions. More specifically, IDS tools aim to detect computer attacks and/or computer misuse, and to alert the concerned individuals upon detection. An IDS installed on a network has the same purpose as a burglar alarm system installed in a house. Through various methods, both detect an intruder/attacker/burglar, and both issue some type of warning or alert.

Although IDSs may be used in conjunction with firewalls, which aim to regulate and control the flow of information into and out of a network, the two security tools are not the same. Firewalls are like a fence or a security guard placed in front of a house. They protect a network and attempt to prevent intrusions, while IDS tools detect whether or not the network is under attack or has, in fact, been breached. IDS tools thus form an integral part of a thorough and complete security system. They don't fully guarantee security, but when used with security policy, vulnerability assessments, data encryption, user authentication, access control, and firewalls, they can greatly enhance network safety.

Intrusion detection systems serve three essential security functions: they **monitor, detect,** and **respond** to unauthorized activity by company insiders and outsider intrusion as depicted in Fig. 4.11. Intrusion detection systems use policies to define certain events that, if detected will issue an alert. In other words, if a particular event is considered to constitute a security incident, an alert will be issued if that event is detected. Certain intrusion detection systems have the capability of sending out alerts, so that the administrator of the IDS will receive a notification of a possible security incident in the form of a page, email, or SNMP trap. Many intrusion detection systems not only recognize a particular incident and issue an appropriate alert, they also respond automatically to the event. Such a response might include logging off a user, disabling a user account, and launching a script.

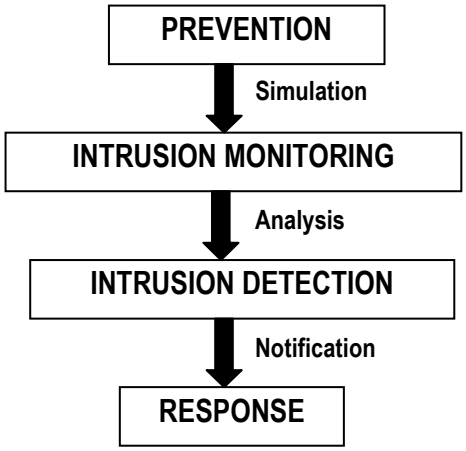


Fig. 4.11: Security Functions of IDS

IDS Services

1. Misuse Detection or Signature Detection

Commonly called Signature Detection, this method uses specifically known patterns of unauthorized behavior to predict and detect subsequent similar attempts. These specific patterns are called signatures. For host-based intrusion detection, one example of a signature is "three failed logins." For network intrusion detection, a signature can be as simple as a specific pattern that matches a portion of a network packet. For instance, a packet content signatures and/or header content signatures can indicate unauthorized actions, such as improper FTP initiation. The occurrence of a signature might not signify an actual attempted unauthorized access (for example, it can be an honest mistake), but it is a good idea to take each alert seriously. Depending on the robustness and seriousness of a signature that is triggered, some alarm, response, or notification is sent to the proper authorities.

2. Target Monitoring

These systems do not actively search for anomalies or misuse, but instead look for the modification of specified files. This is more of a corrective control, designed to uncover an unauthorized action after it occurs to reverse it. One way to check the covert editing of files is by computing a cryptographic hash beforehand and comparing this to new hashes of the file at regular intervals. This type of system is the easiest to implement, because it does not require constant monitoring by the administrator. Integrity checksum hashes can be computed at whatever intervals one wishes, and on either all files or just the mission/system critical files.

3. Stealth Probes

This technique attempts to detect any attackers that choose to carry out their mission over prolonged periods of time. Attackers, for example, check for system vulnerabilities and open ports over a two-month period, and wait another two months to actually launch the attacks. Stealth probes collect a wide-variety of data throughout the system, checking for any methodical attacks over a long period of time. They take a wide-area sampling and attempt to discover any correlating attacks. In effect, this method combines anomaly detection and misuse detection in an attempt to uncover suspicious activity.

Summary

This chapter deals with the historical view of the Internet and TCP/IP. It gives the reader a detailed understanding on the generic top-level domains and discusses the TCP/IP suite in detail. A comparative analysis of the OSI Model and TCP/IP suite provides an insight about the functioning of the protocols over the network. Intrusion detection systems play a vital role in the process of protecting the systems over Internet.

Questions

1. Which of the following is not a layer in TCP/IP suite?
   a. Session Layer
   b. Application Layer
   c. Data Link Layer
   d. Transport Layer
2. Which of the following is a traditional scheme of IP Addressing?
   a. IPv6
   b. IPv4
   c. Broadcasting
   d. None of these

3. gTLD stands for_____.
   a. general Top-level Directory
   b. general Trust-level Directory
   c. general Trust-level Domain
   d. generic Top-level Domain

4. ARP stands for _____ .
   a. Address Reverse Protocol
   b. Address Rapid Protocol
   c. Address Resolution Protocol
   d. Address Reserve Protocol

5. Which of the following is not a protocol in the Application Layer of TCP/IP suite?
   a. SMTP
   b. DNS
   c. UDP
   d. TELNET

6. IANA stands for _____.
   a. Internet Assigned Numbers Authority
   b. Intranet Assigned Numbers Authority
   c. Internet Assigned Numbers Association
   d. Intranet Assigned Numbers Association

7. Which of the following is not a server type?
   a. Database
   b. Object
   c. Communications
   d. Security

8. ORB and OMG stand for _____ and _____ respectively.
   a. Object Request Broker, Open Management Group
   b. Open Request Broker, Object Management Group
   c. Object Request Broker, Object Management Group
   d. Open Request Broker, Open Management Group

9. _____provides three bytes to identify class types and leaves the remaining one byte available for hostid numbers.
   a. Class A
   b. Class B
   c. Class D
   d. Class C

10. _____provides one byte to identify class types and leaves the remaining three bytes available for (hosted?) numbers.
    a. Class B
    b. Class A
    c. Class D
    d. Class C

11. Internet Protocol Version 4 (IPv4) uses a small 32-bit number that is split into three fields: _____.
    a. class type, netid, and hostid
    b. class type, version id and hostid
    c. class id, netid, and version id
    d. None of these

12. IEPG stands for _____.
    a. Intranet Engineering Planning Group
    b. Internet Electrical Planning Group
    c. Internet Engineering Planning Group
    d. None of these

13. SNMP is a protocol used in the_____ layer of TCP/IP suite.
    a. Network
    b. Application
    c. Physical
    d. Data Link

14. _____ is an unreliable and connectionless datagram protocol which provides no error checking or tracking.
    a. User Datagram Protocol
    b. Transmission Control Protocol
    c. File Transfer Protocol
    d. Internetwork Protocol

15. TFTP stands for____.
    a. Trivial File Transmission Protocol
    b. Trivial File Transfer Protocol
    c. Trivial File Transfer Procedure
    d. Trivial File Transmission Procedure

16. Which of the following is not an Anti-Intrusion Approach?
    a. Detection
    b. Deflection

c. Solution
d. Counter Measures

17. SIV stands for _____.
    a. System Integrity Verification
    b. Solution Integrity Verifiers
    c. System Integral Verification
    d. System Integrity Verifiers

18. Which of the following is not an intrusion approach for IDS?
    a. Physical Intrusion
    b. Chemical Intrusion
    c. System Intrusion
    d. Remote Intrusion

19. The data transmission rate is controlled by _____ in TCP/IP suite.
    a. Data Link Layer
    b. Application Layer
    c. Network Layer
    d. Physical Layer

20. The _____ is a mechanism used by hosts and routers to send notification of datagram problems back to the sender.
    a. ICMP
    b. TCP
    c. SMTP
    d. TFTP

Answers:

1 a	2 b	3 d	4 c	5 c	6 a
7 b	8 c	9 d	10 b	11 a	12 c
13 b	14 d	15 b	16 c	17 d	18 b
19 d	20 a				

5 Introduction to Firewalls

Learning Objectives

To understand

- The concept of Firewall
- The types of Firewall -
  - Packet filtering Firewall
  - Circuit level gateway
  - Application level gateway
  - Stateful inspection
- The Common implementation structures of a Firewall
  - Single homed Firewall
  - Dual homed Firewall
  - Demilitarized zone or screened-subnet Firewall
- General Controls associated with Firewalls
- Limitations of Firewalls
- Cost Associated with Firewalls
- Life cycle of a Firewall

Introduction

All of us are concerned about the security of data. It is because there are too many peeping Toms in the unsecured network. This chapter explains the technology that protects us against such intruders. Two major aspects of protection are a. protecting information on various types of equipment and devices and b. protecting information that is travelling on networks.

Because tools for penetrating networks are easily available to people, their security has posed a challenging task for network administrators. Coupled with the need to protect sensitive information from external and internal personnel, there is need to enhance network security. That is to protect the internal organisational network and the information assets located therein from external threats like unauthorised access by protecting the paths of access from the external to the internal or secure network. The technology to protect the perimeter from unauthorised access is the firewall.

Firewalls protect the internal network from intentional access that could compromise confidentiality, availability and integrity of the information present in the organisation's internal network. This includes information assets and R&D network. A firewall can either be hardware, software or a combination of both, often referred to as an appliance. Whatever the case, it is an access control mechanism that maintains selective access to the secure or internal access based on rules that are built into it. The firewall sits at the conjunction of two networks **:** the **Secure Network** that is required to be protected and the **Insecure Network**, such as the Internet, as shown in Fig. 5.1.

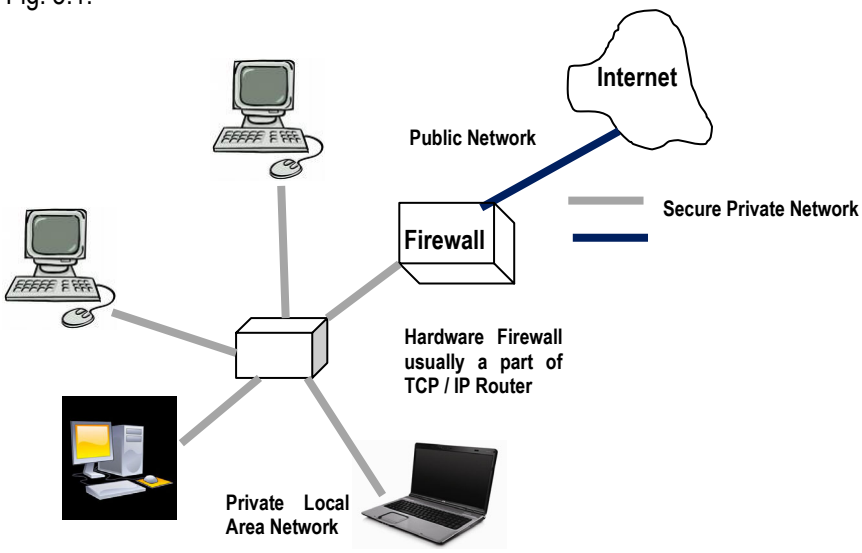


Fig.5.1: Hardware Firewall

As an access control mechanism, the firewall enforces a security policy between an organisation's network and an unsecured environment. It monitors both incoming and outgoing traffic. Based on the rules configured on it, it determines

i. Which inside machines, applications and services may be accessed from outside;
ii. Which outsiders are permitted access to the machines and services; and
iii. Which outside services insiders may have access?

Configuration of the firewall determines how secure or otherwise the network can be. A firewall may be configured in either of the two ways :

- Allow traffic onto the network if it meets certain defined criteria or
- Allow traffic onto the network if it does not meet the criteria defined.

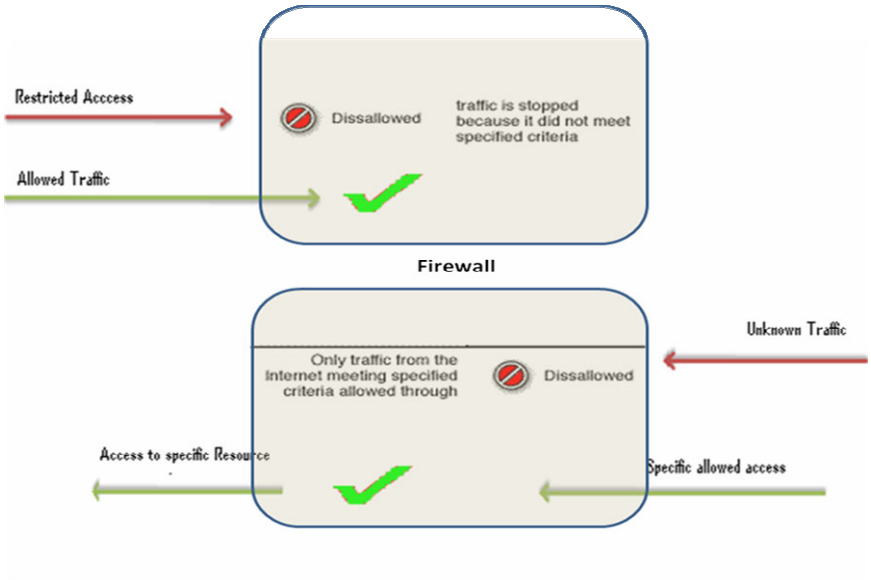


Fig. 5.2: Working of a Firewall

Messages between two or more computers over the networks travel in segments called **packets** and each packet contains a part of the message or instruction called the **payload**, a header and a footer. In the OSI Reference Model, each header contains information about the packet. This generally includes the source, the destination, the port and the service contained in the payload. The firewalls primarily use this header information to filter the packets entering the secure network. Depending on the rule-base, the firewall may either allow a packet to pass or if the packet does not pass the filter, then deny it entry. Fig. 5.2 depicts the working of firewalls.

Anyone who connects to the Internet through a single computer or a private network to the Internet should have a firewall protection. Dial up users who have been victims of malicious attacks and who have lost entire days of work, perhaps having to reinstall their operating system, know that this is not true. Irresponsible pranksters can use automated robots to scan random IP addresses and attack whenever the opportunity presents itself.

Characteristics of a Firewall

For a firewall to be effective, it is desirable that it has the following characteristics:

1. All traffic from inside to outside and from outside to inside shall pass only through the firewall. There should not be any other alternate route. This requires that

before installing a firewall, the perimeter of the internal network is defined. For example, the internal network could consist of a certain number of users within a department, some users working from other unconnected departments and possibly from different locations and may also include mobile users. Hence perimeter definition is a challenge in itself. This involves ensuring the only path for packets from within the network and into it is through a pre-determined network computer or networking device such as a router.

2. The overall security policy of the organisation shall determine what traffic must be permitted to pass through the firewall. Every other traffic must be blocked and this shall be appropriately configured onto the firewall's rule base or ACL.
3. The firewall must be resilient and immune to attacks and penetration.
4. The firewall should maintain a log of traffic that it negotiated and the action taken on it.

The four general techniques that a firewall uses to control access and enforce the site's security policy are:

a. **Service Control :** It determines the types of Internet services that can be accessed, inbound or outbound. It may filter traffic on the basis of IP address and TCP port number, may provide proxy software that receives and interprets each service request before passing it on, or may host the server software itself, such as the Web or mail service.
b. **Direction Control :** It determines the direction in which particular service requests may be initiated and allowed to pass.
c. **User Control :** It controls access to a service according to the user who is attempting to access it. This feature may be applied to users inside the firewall perimeter or to the incoming traffic from external users.
d. **Behaviour Control:** It controls the use of a particular service. For example, filtering e-mail to eliminate spam, etc.

Types of Firewalls

There are three popular types of firewalls:

1. Packet Filtering Router.
2. Circuit-level gateway.
3. Application Level gateway or proxy server.

I. Packet-Filtering Router

i. A packet filtering firewall is the first generation firewall, and works at the network layer of the OSI model or the IP layer of TCP/IP.
ii. It is a fast and cost effective firewall configuration.

iii. It is usually a part of a device called a router, whose function is to forward packets from one network to another, depending on the destination address of the packet which is embedded onto the header of the packet. The header of a packet usually contains the following information: IP source address, the IP destination address, the encapsulated protocol (TCP, UDP, ICMP, or IP Tunnel), the TCP/UDP source port, the TCP/UDP destination port, the ICMP message type, the incoming interface of the packet, and the outgoing interface of the packet. Fig. 5.3 depicts the rules of a packet filtering router.

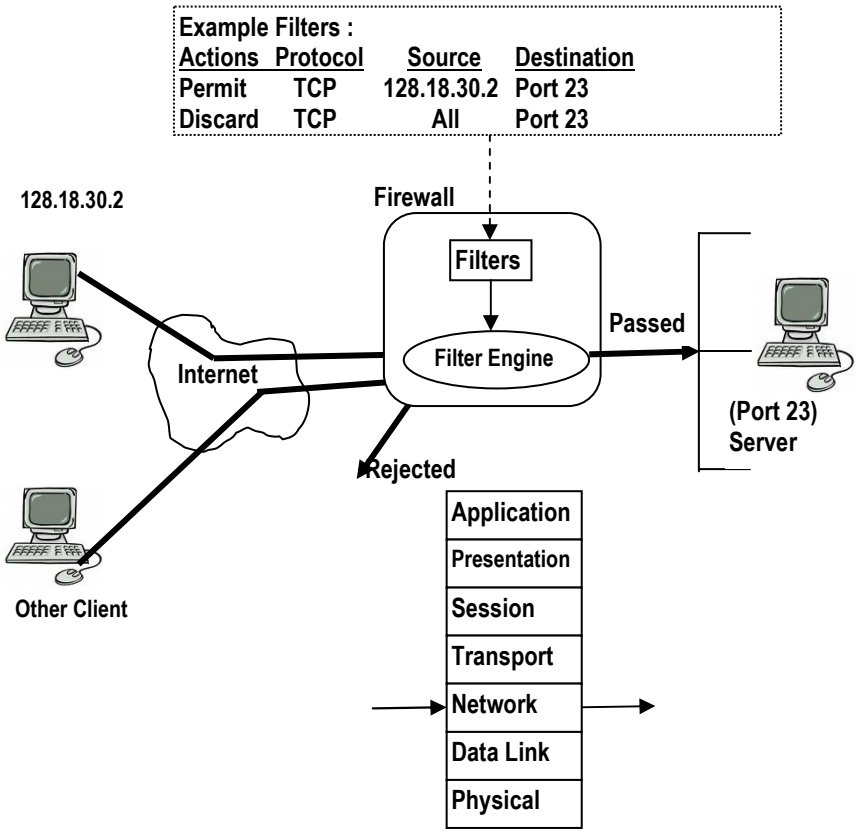


Fig. 5.3: Packet Filtering Router

iv. The firewall examines the header of each incoming packet to determine if it matches one of its packet-filtering rules. Depending on the defined rule, it takes one of the following actions: If a match is found and the rule permits, the packet is passed to the next server according to the information in the routing table.

- If a match is found but the rule disallows it, the packet is rejected.

- If matching rule is not found, a user-configurable default parameter determines if the packet can be forwarded or discarded.

Stateful Inspection Packet Filtering

Stateful Inspection Filtering is a more complex packet filtering technology that filters traffic on more than just source: destination, port number, and protocol type. It keeps track of the state of the current connection to help assure that only desired traffic passes through. This allows the creation of a one-way rule for example, traffic flowing from inside to outside the network.

Packet-filtering routers permit or deny each packet that it receives. The router examines each IP datagram to determine if it matches one of its packet-filtering rules. The filtering rules are based on packet header information that is made available to the IP forwarding process.

- The router maintains a state table for all the connections passing through the firewall.
- So the state of a connection becomes one of the criteria to specify filtering rules.
- If a packet matches an existing connection listed on the table, it will be permitted to go without further checking.
- Otherwise, it is to start a new connection and will be evaluated according to the filtering rules.

Advantages:

i. Packet filter firewalls are fairly easy and simple to implement.
ii. They are transparent to the end users, unlike some of the other firewall methods.
iii. Packet filtering often doesn't require a separate firewall because it's often included in most TCP/IP routers at no extra charge.

Disadvantages:

Its implementation is costly, as it requires a special hardware and software.

Limitations of Packet Filtering Router

i. Although packet filters are easy to implement, configuring them properly poses difficulties, particularly if a large number of rules have to be generated to handle a wide variety of application traffic and users.
ii. Packet filtering is not the best firewall security, because filters are based on IP addresses, and not on authenticated user identification. It also provides little defense against man-in-the-middle attacks and no defense against forged IP addresses.

iii. Also, packet filtering depends on IP port numbers, which isn't always a reliable indicator of the application in use; protocols like Network File System (NFS) use varying port numbers, making it difficult to create static filtering rules to handle their traffic.
iv. Many packet filtering routers lack robust logging capabilities.

Packet Filtering Routers are susceptible to some of the following attacks:
They work on a small set of data present on the Packet header. Since they have such meager information, the firewall is limited in its decision making. Because of this, these firewalls are susceptible to the following attacks:

a. Source IP Address Spoofing Attacks

In this type of attack, an intruder transmits packets that falsely contain the source IP address of a permitted system. This is done with an idea that the use of a spoofed source IP address will allow penetration of systems.

b. Source Routing Attacks

In a source routing attack, the source station specifies the route that a packet should take as it crosses the Internet. It is designed to bypass security measures and cause the packet to follow an unexpected path to its destination.

c. Tiny Fragment Attacks

Tiny Fragment Attack is a class of attack on Internet firewalls taking advantage that it is possible to impose an unusually small fragment size on outgoing packets. If the fragment size is made small enough to force some of a TCP packet's into the second fragment. If the filtering implementation does not enforce a minimum fragment size, a disallowed packet might be passed.

II. Application Level Gateways

i. Also called a “Proxy Sever”, an Application Level Gateway acts as a relay of application-level traffic. It operates in the Application layer of the OSI model. Here, a special-purpose code (a proxy service) is installed on the gateway for each desired application. If a proxy code for a particular application is not installed, the service is not supported. Users are permitted access only to the proxy services.
ii. The user contacts the gateway using a TCP/IP application such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed. When the user responds and provides a valid user-id and authentic information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two end-points. If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded

across the firewall. Further, the gateway can be configured to support only specific features of an application that the network administrator considers acceptable while denying all other features.

iii. In general terms, an application level gateway that is configured to be a web proxy will not allow any ftp, gopher, telnet or other traffic to go through. Because they examine packets at the application layer, they can filter application specific commands, such as http:post and get, etc. This cannot be accomplished with either packet filtering firewalls or circuit level neither of which knows anything about the application level information.

Fig. 5.4 explains the working of an Application-Level Gateway.

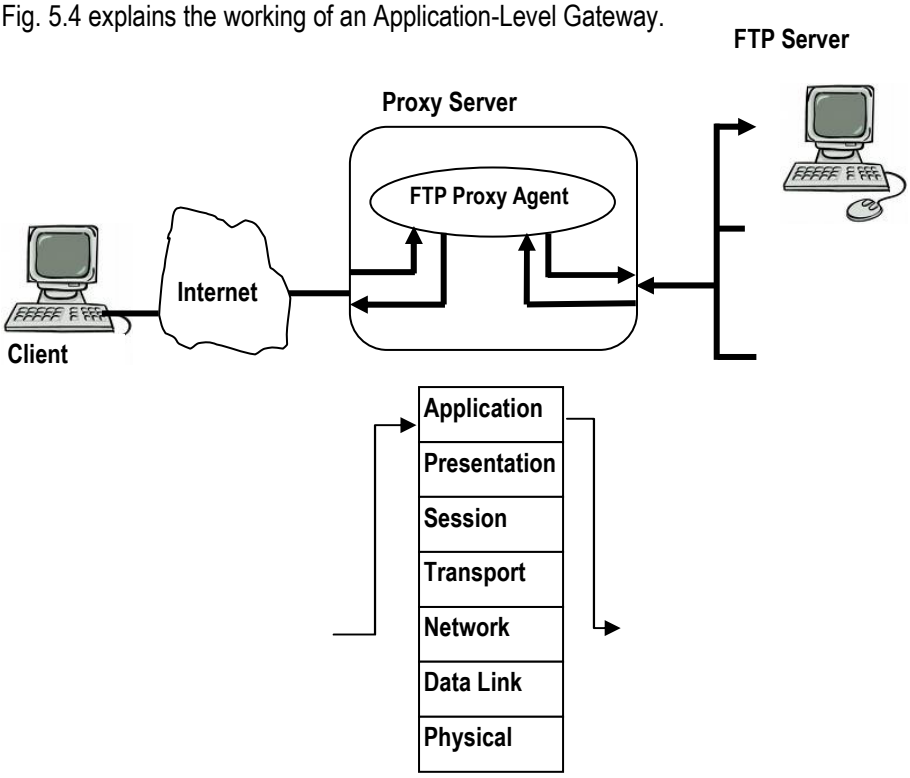


Fig. 5.4 : Application level Gateway

Advantages of Application-Level Gateways

i. Application level gateways provide complete control over each service, since the proxy application limits the command set and determines which internal hosts may be accessed by the service.

ii. There is complete control over services that are permitted. Absence of a proxy for a particular service means that the service is completely blocked.

iii. Application-level gateways have the ability to support strong user authentication.

These are secure and application specific. Since it provides detailed logging information, it is of immense use as an audit trail.

Disadvantages of Application-Level Gateways

i. Implementation of Application level gateways might cause performance issues as the complex rules set may require significant computing resources.
ii. It is vulnerable to bugs present in the Operating system and any application running on the system.

III. Circuit-Level Gateways

i. Circuit Level Filtering takes control a step further than a Packet Filter. This is a firewall approach which validates connections before allowing data to be exchanged.
ii. Circuit level gateways work at the session layer of the OSI model, or the TCP layer of TCP/IP. They can be a stand-alone system or can be a specialized function performed by an application-level gateway for certain applications.
iii. A circuit-level gateway does not permit an end-to-end connection; rather, the gateway sets up two TCP connections: one between itself and a TCP user on an inner host and another one in between itself and a TCP user on an outside host. This firewall does not merely allow or disallow packets but also determines whether or not the connection between both ends is valid according to configurable rules. Then it opens a session and permits traffic only from the allowed source and possibly only for a limited period of time. Once the two connections are established, the gateway relays TCP segments from one connection to the other without examining the contents. Whether a connection is valid or not, may be based upon :

- destination IP address and/or port
- source IP address and/or port
- time of the day
- protocol
- user
- password

Every session of data exchange is validated and monitored and all traffic is disallowed unless a session is open. Fig. 5.5 explains Circuit – level gateway.

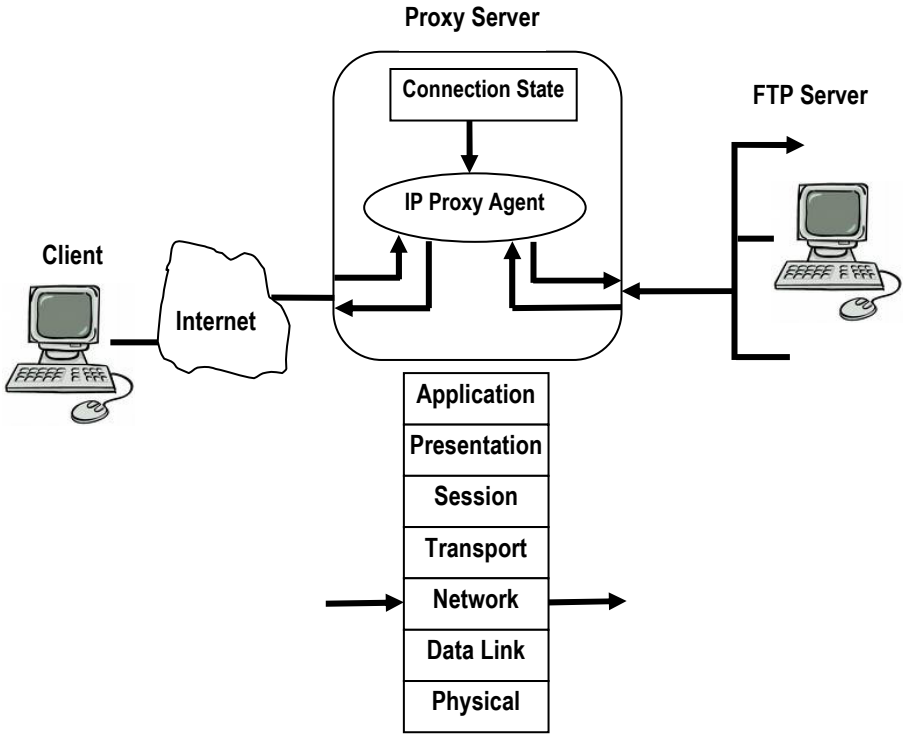


Fig. 5.5: Circuit Level Gateway

Advantages of Circuit – level Gateways

i. Circuit-level gateways are often used for outgoing connections only when the system administrator trusts the internal users. Their chief advantage is that the firewall can be configured as a hybrid gateway supporting application-level or proxy services for inbound connections and circuit-level functions for outbound connections. This makes the firewall system easier to use for internal users who want direct access to Internet services, while still providing the firewall functions needed to protect the organization from external attack.
ii. Circuit level gateways are relatively inexpensive.
iii. They have an advantage of hiding information about the private network they protect. On the other hand, they do not filter individual packets.

Disadvantages of Circuit – level Gateways

i. They cannot restrict access to protocol subsets other than the TCP.
ii. Testing the rules applied can be difficult, and may leave the network vulnerable.

Firewall Implementation

In addition to the use of a simple configuration consisting of a single system, such as a single packet filtering router or a single gateway, more complex configurations are possible and indeed quite common. The following are some of the common configurations which can be implemented:

1. Single-Homed Firewall

This is a combination of two systems: a **Packet-filtering router** and a **Bastion host**. This is more secure compared to the packet-filtering router.

Bastion Host is a special purpose computer on a network specifically designed and configured to withstand attacks and perform authentication and proxy functions. The computer generally hosts a single application. For example - a proxy server and all other services are removed or limited to reduce the threat to the computer. It is hardened in this manner primarily due to its location and purpose, which is either on the outside of the firewall and usually involves access from un-trusted networks or computers.

Here, the router is configured is such a fashion that packets from unsecure environment addressed only to the bastion host are allowed. The bastion host then takes care of the application-layer security. Fig. 5.6 shows Single - homed firewall.

Advantages

This configuration has greater security than simply a packet - filtering router or an application - level gateway alone:

i. Its implementation scheme uses both packet-level and application-level filtering, allowing for considerable flexibility in defining security policy.
ii. An intruder has to penetrate two separate systems before the security of the internal network is compromised. The security policy of the organisation determines whether inside systems are permitted direct access to the unsecured environment, or whether they are required to use the proxy services on the bastion host.
iii. This also provides flexibility in providing direct Internet access. For example, an internal network may include a public information server such as a Web server for which a high level of security is not required. In that case, the router can be configured to allow direct traffic between the information server and the Internet.

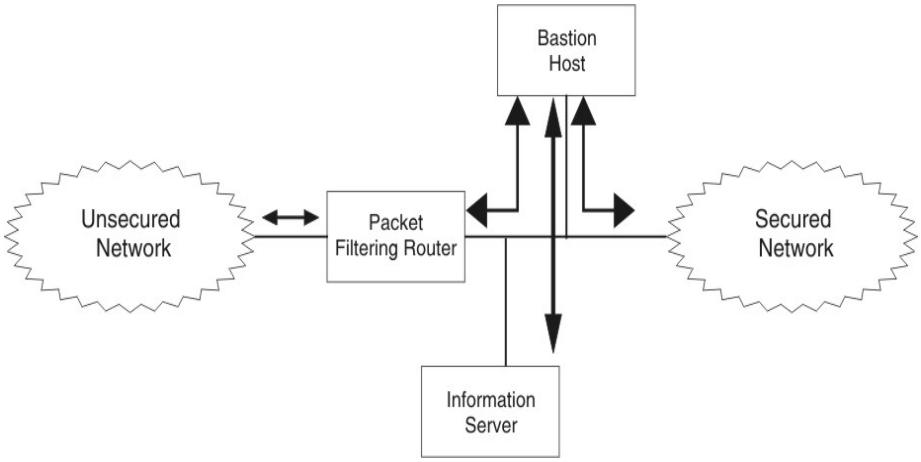


Fig. 5.6: Single Homed Firewall

Limitations

In the single - homed configuration, if the packet-filtering router is completely compromised, traffic flows directly through the router between the internet and other hosts on the private network.

2. Dual-Homed Firewall

In the single-homed firewall, there is always a possibility that if the packet-filtering router is compromised, packets will directly move into the secure environment bypassing the bastion host. This is blocked in a dual-homed firewall. The bastion host has two NICs, one for the unsecured network and another for the secured network. This facility disables bypassing the proxy services. The physical topology forces all traffic to go through the bastion host. Fig. 5.7 shows the picture of a dual-homed firewall.

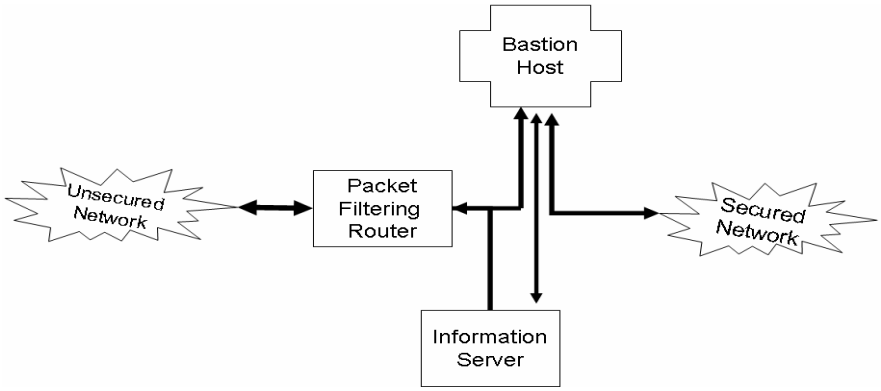


Fig. 5.7: Dual-Homed Firewall

3. "Demilitarized Zone" or Screened-Subnet Firewall

This firewall system is the most secure for it employs two packet-filtering routers or other firewall types and a bastion host. It supports both network and application layer security while defining a **Demilitarized Zone (DMZ)** network.

For all incoming traffic, the outside router (the router facing the internet) directs all the packets to the bastion host. The bastion host together with the firewall does preliminary filtering and if the packet passes the test, directs the packet to the less secure or DMZ. This usually contains the IT components that require public access such as a mail server, web server, etc. However, where the packet needs to travel into the secure network, which is configured as a separate segment, the inside router along with the second firewall provides a second line of defence, managing DMZ access to the private network by accepting only traffic originating from the bastion host as shown in Fig. 5.8.

For outgoing traffic, the inside router manages private network access to the DMZ network. It permits internal systems to access only the bastion host. The filtering rules on the outside router require use of the proxy services by accepting outgoing traffic only from the bastion host.

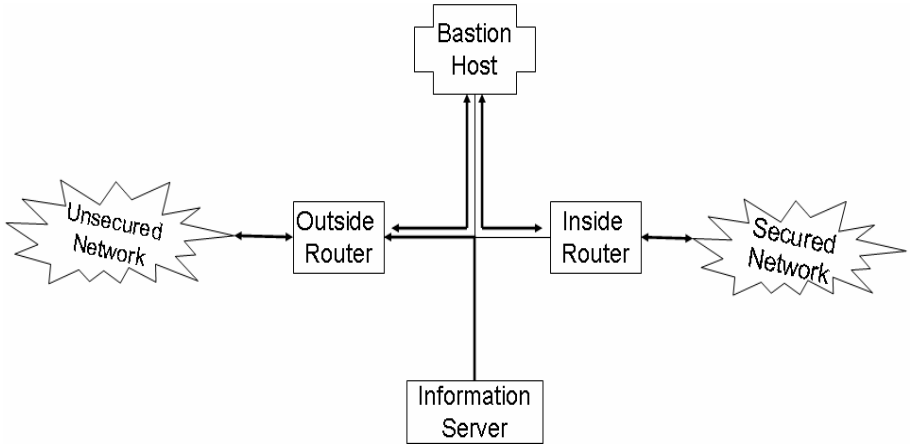


Fig. 5.8: Screened Subnet Firewall

The several key benefits of a screened subnet firewall system are:

- An intruder must penetrate three separate devices of the outside router, the bastion host, and the inside router to infiltrate the private network.
- Since the outside router advertises the DMZ network only to the Internet, systems on the Internet do not have routes to the protected private network. This ensures that the private network is "invisible."

- Since the inside router advertises the DMZ network only to the private network, its systems do not have direct routes to the Internet. This ensures that inside users access the Internet via the proxy services residing on the bastion host.
- Packet-filtering routers direct traffic to specific systems on the DMZ network, eliminating the need for the bastion host to be dual-homed.
- Since the DMZ network is a different network than the private network, a Network Address Translator (NAT) can be installed on the bastion host to eliminate the need to renumber or re-subnet the private network.

Limitations of a Firewall

The general limitation of a firewall is that it provides a false sense of security. Most management's tend to think that simply installing a firewall will provide them with the highest level of security, which is not true. Given below are me of the most common reasons for the failure of the firewall technology.

A software based firewall if installed on an improperly secured computer or network device such as a router can lead to the traffic either bypassing the firewall or the firewall itself being attacked or being ineffective in other ways. Hence there is need to ensure that the firewall is installed on a secure host called the **Bastion Host**.

1. The firewall filters the traffic passing through the firewall using the firewall rule base, which is configured with the organization's perimeter access policy. Hence if the configuration of the firewall rule base is incorrect or ineffective, naturally the firewall will fail to protect the interests thereof.
2. The network perimeter may not be properly defined; because of this any traffic from the internal network that does not pass through the firewall and connects to the external unprotected network can endanger the security of the internal network by opening up an unprotected logical access path. Hence the architecture of the network and ensuring its appropriate segmentation and closing of all other possible paths of logical access is critical to the success of fire-walling.
3. The objective of network perimeter protection can also fail if an inappropriate type of firewall is used for protecting highly sensitive assets. For example, using only a packet filter to protect a bank's internet banking data server.
4. Some other firewall limitations include:
    - **Viruses :** Not all firewalls offer full protection against computer viruses as there are many ways to encode files and transfer them over the Internet.
    - **Attacks :** Firewalls can't protect against attacks that don't go through it. For example, a firewall may restrict access from the Internet, but may not protect the equipment from dial in access to the computer systems.

- **Architecture :** Firewalls reflect the overall level of security in the network. An architecture that depends upon one method of security or one security mechanism has a single point of failure. A failure in its entirety, or through a software application bug may open the company to intruders.
- **Configuration :** A firewall can't tell users if it has been incorrectly configured. Only trained professionals can properly configure firewalls.
- **Monitoring :** Some firewalls may notify users if a perceived threat occurs, but can't notify them if someone has hacked into their network. Many organizations find that they need additional hardware, software and network monitoring tools to take care of this.
- **Encryption :** While firewalls and Virtual Private Networks (VPNs) are helpful, they don't encrypt confidential documents and E-mail messages sent within an organization or to outside business contacts. Formalized procedures and tools are needed to provide protection of confidential documents and electronic communications.
- **Management** : Firewalls stop incoming threats but organizations still require a formalized management, destruction, and archival procedure for their electronic documents. Electronic messages taken out of context can put an organization in financial jeopardy.
- **Masquerading** : Firewalls can't stop a hacker from masquerading as an employee. Hackers can use a number of ways to acquire user ids and related passwords.
- **Policies :** Firewalls are no replacement for a strong Security Policy and Procedure Manual. An organization's security structure is only as strong as its weakest link. Security professionals have the experience needed to help protect our reputation.
- **Vulnerabilities** : Like a deadbolt lock on a front door, a firewall can't tell users if there are other vulnerabilities that might allow a hacker access to an internal network. Organizations frequently rely on Security Vulnerability Assessments to help them manage their risks.

An analogy can be drawn between protecting a network and protecting a house. Firewalls are like bouncers that only allow traffic through the front door with an invitation. They may also allow certain traffic without an invitation that is looking for marketing material into the garage. However, the firewall does not know if the traffic is harmful, for example: containing a virus or benign.

General Controls associated with Firewalls

A firewall is not just a piece of hardware but is a combination of hardware, software and management access control rules. Hence controls regarding firewalls are classified into the following groups:

1. **Physical security controls:** Machines in various locations including servers, firewalls, nodes and communicating devices have to be physically secured. For this purpose, companies have to devise policies to ensure physical security of hardware.
2. **Operating System Security:** A firewall must run a secure hardened computer with an absolutely tamper-proof operating system. It should be ensured that there are no vulnerabilities at the OS level. The network administrator has to ensure that patches with regard to OS are up-to-date.
3. **Configuration of Firewall Policy:** The firewall has to be carefully configured according to the approved policy of an organisation, and provided with permitted and prohibited list of source and destination addresses and services.
4. **Change Control Procedures:** The network administrator must ensure that patching process is complete. Moreover, changes in the access rules and privileges of the users have to be properly authorised and managed.
5. **Documentation :** Documents about the network diagrams and configurations have to be thoroughly examined for vulnerabilities. It is also essential to document log analysis and other changes that are being made.
6. **Log Monitoring :** Firewalls provide features for logging every kind of traffic handled by them. These logs have to be examined at regular intervals by technical experts who are knowledgeable about internet attacks & internal vulnerabilities. It is also essential to ensure that the log files are not tampered. In case any significant intrusions are detected, forensic analysis may be required. The firewall should offer such features.

Cost related factors to Firewalls

Costs associated with firewalls include:

i. Firewall hardware purchase.
ii. Firewall hardware maintenance.
iii. Firewall software development or purchase.
iv. Firewall software update costs.
v. Firewall administrative setup and training.
vi. Ongoing administration and trouble-shooting, and

Cost of lost business or inconvenience from a broken gateway or blocked services, and the loss of some services or convenience that an open connection would supply.

Phases in the Firewall Life Cycle

Fig. 5.9 displays the phases involved in the Firewall Life Cycle, which progresses diagonally, beginning with the all important definition of security policy and arriving at implementation, review, and testing after high-level design, selection of components,

and detailed design. Even after the firewall is in use, periodic review and testing during the system's lifetime may result in an earlier phase being revisited (indicated by the upward-pointing blue arrows), as when a new, improved firewall component becomes available or when defects in an earlier phase are discovered.

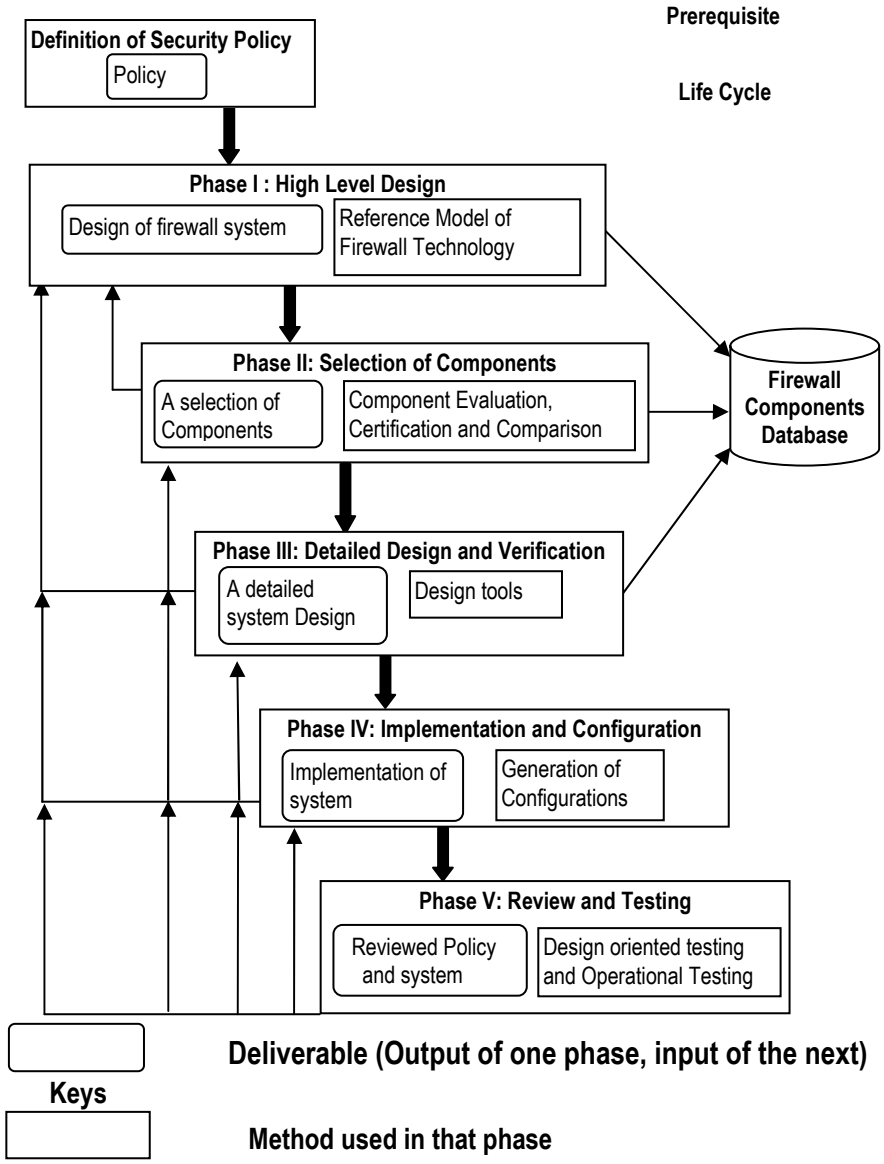


Fig. 5.9: Phases in Firewall Life Cycle

The Firewall Life Cycle is summarised in Table 5.1

PHASES	DELIVERABLES	METHODS
High-Level Design	High-level design of firewall system	Reference Model of Firewall Technology
Selection of Components	A selection of Components	Component Evaluation, Component Certification and Component Comparison
Detailed Design and Verification	A detailed system Design	Design Tool
Implementation and Configuration	Implementation of system	Generation of configuration
Review and Testing	Reviewed Policy and System	Design-oriented Testing, Operational Testing

Table 5.1: Firewall Life Cycle Summary

Summary

This chapter provides a detailed account of the concept of the firewall and its types. Firewalls are used to protect a system against any kind of intrusions and have different areas of implementation. General controls are associated with firewalls and at the same time they have certain limitations. The chapter also provides information about life-cycle of a firewall.

Questions

1. Which of the following is not a type of Firewall Implementation Scheme?
   a. Single - Homed Firewall System
   b. Dual - Homed Firewall System
   c. Screened - subnet Firewall System
   d. None of these
2. Which of the following is not a general technique that firewalls use to control access and enforce a site's security policy?
   a. Service Control
   b. Direction Control
   c. Dual Control
   d. User Control

3. DMZ stands for _____.
   a. De-military Zone
   b. Demilitarized Zone
   c. De-military Zone
   d. None of these

4. Which of the following is not a type of firewall?
   a. Application- level gateway
   b. Dual-level gateway
   c. Circuit-level Gateway
   d. Packet Filtering Router

5. Which of the following is not a step in the Firewall Life Cycle?
   a. Configuration of firewall
   b. Review and Testing
   c. Detailed Design and Verification
   d. High Level Design

6. _____ determines the types of Internet services that can be accessed, inbound or outbound.
   a. Dual Control
   b. User Control
   c. Direction Control
   d. Service Control

7. In _____ , the intruder transmits packets that falsely contain the source IP address of a permitted system.
   a. Source IP Address Spoofing Attacks
   b. Source Routing Attacks
   c. Tiny Fragment Attacks
   d. None of these

8. In _____ class of attack on Internet firewalls, it is possible to impose an nusually small fragment size on outgoing packets.
   a. Source IP Address Spoofing Attacks
   b. Tiny Fragment Attacks
   c. Source Routing Attacks
   d. None of these

9. Which phase follows the phase "Detailed Design and Verification" in the Firewall Life Cycle?
   a. Review and Testing
   b. Configuration and Verification
   c. High Level Design
   d. Implementation and Configuration

10. NAT stands for_____.
   a. Network Address Translator
   b. Network Address Transistor
   c. Network address Testing
   d. None of these

11. _____ controls how to use particular services.
   a. Service Control
   b. User Control
   c. Behavior Control
   d. Direction Control

12. _____ determines the direction in which particular service requests may be initiated and allowed to pass.
   a. Service Control
   b. User Control
   c. Behavior Control
   d. Direction Control

13. _____ is also known as a Proxy Server.
   a. Dual Level Gateway
   b. Application- Level Gateway
   c. Circuit-Level Gateway
   d. None of these

14. _____are operational at the Application Layer of the OSI Model.
   a. Application-Level Gateways
   b. Dual Level Gateways
   c. Circuit-Level Gateways
   d. None of these

15. _____ are operational at the Session Layer of the OSI Model.
    a. Application-Level Gateways
    b. Dual Level Gateways
    c. Circuit-Level Gateways
    d. None of these

Answers:

1 D	2 c	3 C	4 b	5A	6 d
7 A	8 b	9 D	10 a	11 C	12 d
13 B	14 a	15 C			

6 Cryptography

Learning Objectives

To understand

- The basic concept of Cryptography.
- Need for Cryptography and goals of cryptographic systems.
- Detailed and comparative study of Symmetric Key and Asymmetric Key Algorithms.
- Public Key Encryption and its working.
- The RSA : An Example of r Public-Key Encryption.
- Digital Signatures, Digital Envelopes, Digital Certificates.
- The concept of Cryptanalysis and its ways.

Introduction

This chapter deals with the mechanism with which a piece of information is hidden during its transmission and unfolded at the desired destination end only. The concept of hiding the information is known as Cryptography and there are several cryptographic algorithms by which information can be secured.

Cryptography

Cryptography means the practice and study of hiding information. It deals with difficult problems. A problem may be difficult because -

(i) The solution requires some secret knowledge, such as decrypting an encrypted message or signing some digital document; or

(ii) It is intrinsically difficult, such as finding a message which produces a given hash value.

In technical terms, Cryptography is the study of techniques and applications to solve difficult problems.

- **Plaintext** means an original message.
- **Ciphertext** means a coded message.

- **Enciphering/ Encryption is** the process of converting from plaintext to ciphertext.
- **Deciphering/ Decryption** is restoring plaintext from ciphertext.

Many schemes used for encryption constitute the area of **Cryptography**. These are called **Cryptographic Systems** or **Ciphers**. Techniques used for deciphering a message without any knowledge of enciphering details constitute **cryptanalysis**. Cryptography and cryptanalysis form **Cryptology**.

A **cryptanalyst** is one who analyses cryptographic mechanisms and decodes messages for military, political, or law enforcement agencies or organizations. He helps to provide privacy to people and corporations, and keeps hackers out of important data systems, as much as he possibly can.

BRIEF HISTORY OF CRYPTOGRAPHY

The following is a summary of the brief history of cryptography.

i. Julius Caesar is credited with creating one of the earliest cryptographic systems to send military messages to his generals. This technique is called Caeser cipher. It is actually shift by 3 rule, wherein alphabet A is replaced by D, B is replaced by E, and so on.
ii. After this, various cryptographic systems were invented. These encryption systems were known as Symmetric Cryptographic Techniques.
iii. One of the most influential cryptanalytic papers of the 20th century, William F. Friedman's monograph **The Index of Coincidence and Its Applications in Cryptography,** appeared as a research report of the private Riverbank Laboratories in 1918.
iv. In 1970, Horst Feistel of IBM Watson Laboratory began the preliminary work for a project which later became the US Data Data Encryption Standard (DES).
v. Whitfield Diffie and Martin Hellman proposed the idea of public-key cryptography in 1976.
vi. In 1977, DES was adopted by the National Bureau of Standards as FIPS Standard.
vii. RSA Public key Algorithm was developed in 1978.
viii. Triple DES came into existence in 1985.
ix. IDEA (International Data Encryption Algorithm), a symmetric block cipher was developed in 1991.
x. Another symmetric block cipher Blowfish was developed in 1993 by Bruce Schneier.
xi. RC5, another symmetric block cipher, was developed by Ron Rivest in 1994.

xii. In October 2000, Rijndael Algorithm was declared the Advanced Encryption Standard by the U.S. government.

NEED FOR Cryptography

Cryptography has gained prominence because people want to safeguard information from falling into wrong hands. It provides solutions to problems of protecting data and identity with regard to data in transit. Since it is virtually impossible to imagine life without the Internet, the security threats envisaged are:

i. **Interruption :** Information becomes unavailable to users when they want it because of damage to the integrity of the message.
ii. **Interception :** The message that is passed by the sender is intercepted by unauthorised persons.
iii. **Modification :** The message that has been intercepted can be tampered because of addition, deletion, or modification of its parts.
iv. **Fabrication or Identity Theft :** An unauthorised person creates a message as though it has originated from the actual sender and sends it to the receiver. Such security threats are further classified into **Active** and **Passive threats**. While passive threats include release of message contents and traffic analysis, active threats encompass masquerading, replay, modification of the message contents and denial of service.

The goals of Cryptographic Systems

The main goals of any Cryptographic System are:

- **Authentication : The process of proving one's identity.** The system provides facilities to the recipient of a message to convince him that the message he has received has originated only from the sender and not from any unauthorised person.
- **Privacy or Confidentiality : Ensuring that no one can read the message except the intended receiver.** Whatever is communicated between two parties should be known only to them. No unauthorised person should get to know about it.
- **Integrity : Assuring the receiver that the received message has not been altered in any way.** The system should provide facilities to the recipient that the message he has received has not been tampered with.
- **Non-Repudiation : A mechanism to prove that the sender really sent this message.** The system should provide facilities to the recipient to convince him t*h*at the message he has received has originated only from the sender, and the sender at a later point of time does not repudiate the same.

- **Restriction :** Nothing is communicated to other parties except that which is specifically desired to be communicated.

Messages and Encryption

The following are the basics about messages and encryption.

1. Any message that is intelligible is termed plaintext, also called as **Cleartext**.
2. Encryption is the process of converting the given plaintext into a scrambled form.
3. The encrypted message that is unintelligible is called **Ciphertext**.
4. The process of turning ciphertext back into plaintext is called **Decryption**.

Algorithms and Keys

The following are the basics of algorithms and keys:

1. A cryptographic algorithm, also called a **Cipher**, is the mathematical function used for encryption and decryption. This is done through two functions : one for encryption and the other for decryption. A cryptographic algorithm must be easy to use, but extremely difficult to crack.
2. The process of encryption and decryption is further strengthened with the help of a key. It must be long enough to break but short enough to use and transmit. For example, when we use binary numbers as the base and the length of the key is 4 bits, then the key space has 2^4 combinations (16 combinations). If the key length is 10, and then the key space should be 2^{10} combinations (1024 combinations). The more the length of the key, the more will be the key space.

Cryptography not only protects data from theft or alteration, but also ensures its authentication. There are, in general, three types of cryptographic schemes to accomplish these goals :

i. **Secret Key or Symmetric Cryptography (SKC) :** Uses a single key for both encryption and decryption.
ii. **Public Key or Asymmetric Cryptography (PKC) :** Uses one key for encryption and another for decryption.
iii. **Hash Functions :** Uses a mathematical transformation to irreversibly "encrypt" information.

I. Symmetric or Secret Key Cryptography

1. In S**ecret Key Cryptography**, the sender uses the key (or some set of rules) to encrypt the plaintext and sends the ciphertext to the receiver. The receiver applies the same key (or rule set) to decrypt the message and recover the plaintext. Because a single key is used for both functions, this cryptography is also called S**ymmetric Encryption**.

2. The key must be known to both the sender and the receiver; that, in fact, is the secret, but the difficulty with this system is the distribution of the key. Symmetric Key algorithms can be further divided into two categories.
    a. **Stream algorithms or Stream Ciphers:** These operate on a single bit (byte or computer word) at a time and implement some form of feedback mechanism so that the key is constantly changing.
    b. **Block Algorithm or Block Cipher:** A block cipher is so - called because the scheme encrypts one block of data at a time by using the same key on each groups of bits called **block**.

In general, the same plaintext block will always encrypt to the same ciphertext when using the same key in a block cipher whereas the same plaintext will encrypt to different ciphertext in a stream cipher.

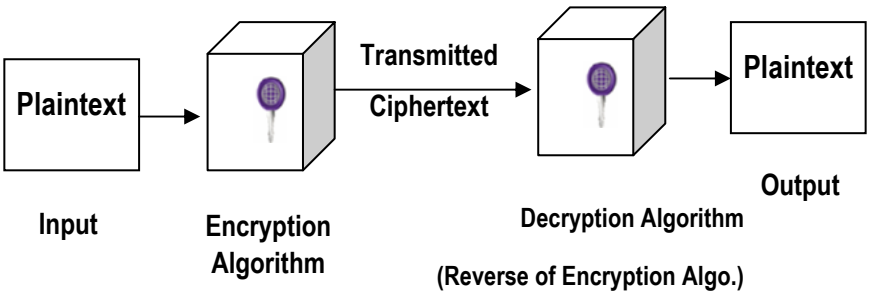


Fig. 6.1: Symmetric Cryptography by using the same key

Some Symmetric Key Encryption Algorithms

There are a number of symmetric key encryption algorithms. They are:

1. **Caeser Cipher**
   It is one of the oldest encryption methods and called "Shift by 3" method. In this, every plaintext alphabet is replaced by its third alphabet. For example, alphabet A is replaced by D, B is replaced by E, and so on.

2. **Mono - alphabetic substitution**
   Every plaintext alphabet is replaced by a unique ciphertext alphabet of the user's choice, but there are no rules to govern the choice of the alphabet.

3. **Poly - alphabetic substitution**
   The user creates a grid of size 26 by 26 English alphabets. The sender and receiver then agree on a key word and use it and the grid to encrypt and decrypt messages.

4. Transposition ciphers

The sender and receiver agree on a keyword in which no alphabets are repeated. Then using the keyword, the message is transposed either in row fashion or columnar fashion to get the ciphertext. The keyword is used in the reverse process for decryption.

5. Data Encryption Standard (DES)

The most common secret-key cryptography scheme used today is the Data Encryption Standard (DES). Originally designed by IBM in the 1970s, this was adopted by the National Bureau of Standards (NBS) (now the National Institute for Standards and Technology (NIST)) in 1977 for commercial and unclassified government applications. DES has been adopted as Federal Information Processing Standard 46 (FIPS 46-2) by the American National Standards Institute as X3.92. DES is a block-cipher employing a 56-bit key that operates on 64-bit blocks. DES has a complex set of rules and transformations.

In order to increase the power of DES, there are two important variants of DES:

i. **Triple-DES (3DES) :** This uses three independent 56-bit keys to provide 3DES with an effective key length of 168 bits. Triple-DES has been adopted by ANSI as standard X9.52 and is a proposed revision to FIPS 46 as draft FIPS 46-3.

ii. **DESX** : A variant revised by Ron Divest, which combines 64 additional key bits to the plaintext prior to encryption, effectively increasing the key length to 120 bits.

6. CAST-128

It is named after Carlisle Adams and Stafford Tavares of Nortel. Similar to DES, this is a 64-bit block cipher using 128-bit keys. A 256-bit key version is called CAST-256.

7. International Data Encryption Algorithm (IDEA)

It is another DES – like substitution – permutation crypto algorithm, employing a 128 bit key operating on a 64-bit block.

8. Rivest Ciphers

Named after its inventor Ron Rivest, it is a series of SKC algorithm.

- **RC1 :** Designed on paper but never implemented.
- **RC2 :** A 64-bit block cipher using variable-sized keys designed to replace DES. Its code has not been made public although many companies have licensed RC2 for use in their products.

- **RC4 : A** stream cipher using variable-sized keys, it is widely used in commercial cryptography products, though it can only be exported by using keys that are 40 bits or less in length.
- **RC5 :** A block-cipher supporting a variety of block sizes, key sizes, and number of encryption passes over the data.
- **RC6 :** An improvement over RC5, RC6 was one of the AES Round 2 Algorithms.

9. Blowfish

It is a symmetric 64-bit block cipher invented by Bruce Schneier. In this, key length can vary from 32 to 448 bits. Blowfish is available for free for all users.

10. Twofish

It is a 128-bit block cipher that uses 128-, 192-, or 256-bit keys. This was also invented by Schneier.

11. Advanced Encryption Standard (AES)

It is a significant advancement over the DES that uses a stronger key size of 128, 192 & 256 bits and is much faster that 3DES. In January 1997, NIST initiated a process with which to develop a new secure cryptosystem for U.S. government applications. The result, the Advanced Encryption Standard (AES), came into being as the "official" successor to DES. In October 2000, NIST announced the selection of Rijndael Algorithm (pronounced as in "rain doll" or "rhine dahl") as the algorithm.

II. Asymmetric or Public Key Cryptography

In symmetric key encryption system, a single key was used for both encryption and decryption. The main problem in this method is that the sender and receiver must agree on the symmetric key without anyone else finding out. If they are in geographically distant locations, they should have some effective means to share the key and at the same time prevent the disclosure of the secret key to anybody else. Anyone who overhears or intercepts the key in transit can later read, modify, and forge all messages encrypted or authenticated using that key. Such a process of generation, transmission and storage of keys is termed the **Key Management**. Where the number of users is more, the problem acquires gigantic proportions.

A solution to the problem of key management was found when Whitfield Diffie and Martin Hellman introduced the concept of **Public Key Cryptography** in 1976. Public key cryptography is the most significant development in cryptography in the last 300-400 years and first described by Stanford University professor Martin Hellmann and graduate student Whitfield Diffie in 1976. PKC depends upon the existence of so-called one way functions, or mathematical functions that are easy to execute whereas their inverse function is relatively difficult to compute.

It uses one key for encryption and another for decryption, such that both keys have a unique relationship (and irreversible as regards use of one key only not clear), i.e. one cannot decrypt with the key that was used for encryption. In this concept, each person gets a pair of keys, one called the public key and the other called the private key. The public key of the individual is published in all places while the private key is never revealed.

A simple analogy is the operation of a locker in a bank. As seen in Fig. 6.2, the keys for encryption and decryption are different.

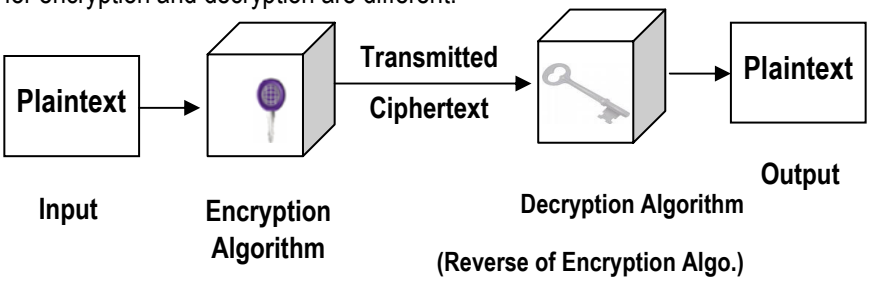


Fig. 6.2: Asymmetric Cryptography by using different keys

Another example could be of Multiplication vs. Factorization. If we have to multiply two numbers 9 and 16, then the calculation is easy and the answer is144. But if one wants to know the pair of integers that on multiplication gives the result 144, then the calculation would take longer because one has to firstly calculate 8 pair of integer factors and then to determine which one is the correct pair.

Requirements of Asymmetric Cryptography

Diffie and Hellman have given the requirements of a Public-Key Cryptographic system, thus there is a mathematical relationship between the public key portion and private key portion.

i. On account of the mathematical relationship, if a message is encrypted by one portion (public or private) of the key, it can be decrypted by the other portion (private or public) of the key.
ii. The generation of the public key and the private key for an individual should be computationally easy.
iii. The encryption and decryption process should be computationally feasible.
iv. Given the public key portion, it is computationally not feasible to arrive at the private key portion.
v. Given the public key and sufficient quantity of the ciphertext, it is computationally infeasible to arrive at the plaintext.

Working of PKC

i. Generic PKC employs two keys that are mathematically related although knowledge of one key does not allow someone to easily determine the other key.
ii. One key is used to encrypt the plaintext and the other key is used to decrypt the ciphertext. The important point here is that it does not matter which key is applied first, but that both keys are required for the process to work.
iii. Because two keys are required for the job, this approach is also called **Asymmetric Cryptography.**
iv. In PKC, one of the keys is designated the **public key** and may be advertised as widely as the owner wants. The other key is designated the **private key** and is never revealed to another party.

Example: If there are two users, say Aditya and Bhaskar who want to exchange some confidential information. Both have their own public and private keys. Since the public keys of individuals are made public and the private keys are not disclosed, all that Aditya does is as follows:

a. Looks for the Public key of Bhaskar (Receiver's Public Key);
b. Encrypts the message using the Public Key of Bhaskar;
c. Sends the message to him;
d. Bhaskar uses his private key to decrypt the message and reads it.

Even if an unauthorised person intercepts the encrypted message, he cannot decrypt the same for he does not have the private key of Bhaskar. **(non-repudiation).**

Public-key cryptography algorithms that are in use these days for **key exchange** or **digital signatures** include:

1. **Diffie-Hellman :** After the RSA algorithm was published, Diffie and Hellman came up with their own algorithm. But D-H is used for secret-key key exchange only, and not for authentication or digital signatures.
2. **Digital Signature Algorithm (DSA) :** The algorithm specified in NIST's Digital Signature Standard (DSS) provides digital signature capability for the authentication of messages.
3. **ElGamal :** Designed by Taher Elgamal, a PKC system is similar to Diffie-Hellman and used for key exchange.
4. **Elliptic Curve Cryptography (ECC) :** A PKC algorithm based upon elliptic curves. ECC can offer levels of security with small keys comparable to RSA and other PKC methods. It was designed for devices with limited compute power and/or memory, such as smartcards and PDAs.

5. **Public-Key Cryptography Standards (PKCS) :** A set of interoperable standards and guidelines for public-key cryptography, designed by RSA Data Security Inc. are:
    - PKCS #1 : RSA Cryptography Standard
    - PKCS #2 : Incorporated into PKCS #1.
    - PKCS #3 : Diffie-Hellman Key-Agreement Standard
    - PKCS #4 : Incorporated into PKCS #1.
    - PKCS #5 : Password-Based Cryptography Standard (also referred as RFC 2898)
    - PKCS #6 : Extended-Certificate Syntax Standard (
    - PKCS #7 : Cryptographic Message Syntax Standard (Also RFC 2315)
    - PKCS #8 : Private-Key Information Syntax Standard (Also RFC 5208)
    - PKCS #9 : Selected Attribute Types (Also RFC 2985)
    - PKCS #10 : Certification Request Syntax Standard (Also RFC 2986)
    - PKCS #11 : Cryptographic Token Interface Standard
    - PKCS #12 : Personal Information Exchange Syntax Standard
    - PKCS #13 : Elliptic Curve Cryptography Standard
    - PKCS #14 : Pseudorandom Number Generation Standard is no longer available
    - PKCS #15 : Cryptographic Token Information Format Standard
6. **Cramer-Shoup:** A public-key cryptosystem proposed by R. Cramer and V. Shoup of IBM in 1998.
7. **Key Exchange Algorithm (KEA):** A variation on Diffie-Hellman, it was proposed as the key exchange method for Capstone.
8. **LUC:** A public-key cryptosystem designed by P.J. Smith and based on Lucas sequences, it can be used for encryption and signatures, using integer factoring.
9. **RSA:** The first, and still the most common PKC implementation, named after the three MIT mathematicians who developed it - **Ronald Rivest, Adi Shamir, and Leonard Adleman** in 1977. RSA today is used in hundreds of software products and can be used for key exchange, digital signatures, or encryption of small blocks of data.

How the public key encryption method works?

Fig. 6.3 shows the working of a Public-Key encryption system

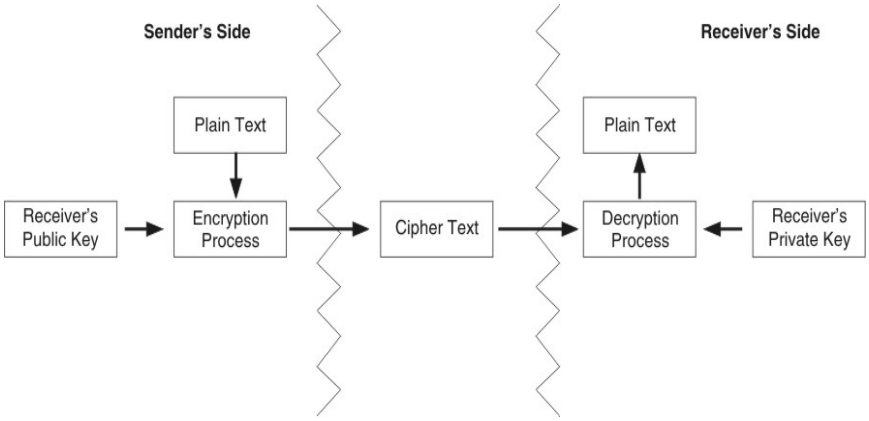


Fig. 6.3: Working of a Public-Key Encryption System

RSA : An Example of Public-Key Encryption

In RSA cryptography, a public and private key is generated by using the following steps:

1. Generate two large prime numbers, **p** and **q**.
2. Compute **n = pq**.
3. Compute the totient **m = (p-1)(q-1).**
4. Choose a small integer **e** such that **1< e < m**, and **e** and **m** share no divisors other than 1 (ie., e and m are coprime).
5. Determine **d using modular arithmetic** (1<d<m) which satisfies the congruence relation **de ≡ 1mod (m),** ie… **de % m = 1**
   - stated differently, **ed – 1** can be evenly divided by the totient **(p-1)(q-1).**
   - This is often computed using the Extended Euclidean algorithm.
   - **d** is kept as the private key exponent.

Publish **e** and **n** as the public key. Keep **d** and **n** as the secret key.

Encryption : $C = P^e \% n$ Decryption : $P = C^d \% n$

(x % y means the remainder of x divided by y) and P is the original message.

RSA Example

Step	Action	Result
1	Take two prime numbers, p and q	p = 7; q = 19
2	Calculate n = pq	n = 7 * 19 = 133
3	Compute m = (p - 1)(q - 1)	m = (7 - 1)(19 - 1) = = 6 * 18 = 108
4	Choose a small number, e coprime to m. e coprime to m refers to the largest number that can exactly divide both e and m (their greatest common divisor, or gcd) is 1. Euclid's algorithm is used to find the gcd of two numbers.	e = 2 => gcd(e, 108) = 2 (no) e = 3 => gcd(e, 108) = 3 (no) e = 4 => gcd(e, 108) = 4 (no) **e = 5 => gcd(e, 108) = 1 (yes)**
5	Find d, such that de % m = 1 This is equivalent to finding d which satisfies de = 1 + nm, where n is any integer. We can rewrite this as d = (1 + nm) / e. Now we work through values of n until an integer solution for e is found.	**With e = 5, m =108 :** n = 0 => d = 1 / 5 (no) n = 1 => d = 109 / 5 (no) n = 2 => d = 217 / 5 (no) **n = 3 => d = 325 / 5 = 65 (yes)**
6	Public Key now is	n = 133, e = 5
7	Private Key now is	n =133, d = 65

The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the modulus n and the private (or decryption) exponent d which must be kept secret.

Encryption by using RSA

The message must be a number less than the smaller of p and q. However, at this point we don't know p or q, so in practice a lower bound on p and q must be published. This can be somewhat below their true value and so isn't a major security concern. For this example, let's use the message P as "6".

Aditi transmits her public key (n,e) to Bharat and keeps the private key secret. Bharat then wishes to send a message P = 6 to Aditi.

With e = 5 and n = 133, Bharat calculates the ciphertext C as -

C = P^e%n
= 6^5%133
= 7776%133
= 62

Now C = 62 is an encrypted form of original message P = 6. Bharat then transmits C to Aditi.

Decryption by using RSA

This works very much like encryption, but since this involves a larger exponent, this is broken down into several steps. The ciphertext C = 62 reaches Aditi. Aditi can recover the original message P from C by using her private key exponent d = 65 by the following computation:

P = C^d % n
= 62^{65} % 133
= 62 * 62^{64} % 133
= 62 * $(62^2)^{32}$ % 133
= 62 * 3844^{32} % 133
= 62 * $(3844 \% 133)^{32}$ % 133
= 62 * 120^{32} % 133

We now repeat the sequence of the operation that reduced 62^{65} to 120^{32} to reduce the exponent down to 1.

= 62 * 36^{16} % 133
= 62 * 99^8 % 133
= 62 * 92^4 % 133
= 62 * 85^2 % 133
= 62 * 43 % 133
= 2666 % 133
= 6

And that matches the plaintext put in at the beginning, so the algorithm worked!

Constraints on RSA

Some of the constraints suggested by researchers on RSA are:

a. p and q should differ in length by only a few digits.
b. Both (p-1) and (q-1) should contain a large prime factor.
c. gcd (p-1, q-1) should be small.

RSA Security

Four possible approaches to attacking the RSA Algorithm are:

i. **Brute Force :** This involves trying all possible private keys.
ii. **Mathematical attacks :** There are several approaches, all factoring the product of two primes.
iii. **Timing attacks :** These depend on the running time of the decryption algorithm.
iv. **Chosen ciphertext attacks :** This type of attack exploits properties of the RSA algorithm.

It has been proved by researchers that there are so many available prime numbers that asymmetric cryptoystems using prime numbers (RSA here) will never run out of them. It has also been proved mathematically that the number of prime numbers of length 512 bits or less is around 10^{150} , which is really a very large number.

Comparison between Symmetric and Asymmetric Key Encryption Algorithms

SYMMETRIC KEY / PRIVATE KEY ENCRYPTION	ASYMMETRIC KEY / PUBLIC KEY ENCRYPTION
Key Size is generally small as compared to public key encryption	Key Size is generally large when compared to private key encryption
Works fast	Slow compared to symmetric key
Consumes less computer resources	Uses more computer resources compared to symmetric key method
Less secure when compared to asymmetric method because of key management problem	Increased security because private keys are never revealed to anyone
Authentication is a cumbersome process and not reliable.	Provides facilities for authentication through digital signatures.
Efficient for encryption	Provides facilities for efficient digital signatures and key management

III. Hash Functions

Hash functions, also called **Message Digests** and **One-way Encryption**, are algorithms that**,** in some sense, use no key. Instead, a fixed length hash value is computed based upon the plaintext that makes it impossible to recover either the contents or length of the plaintext. Hash algorithms are typically used to provide a digital fingerprint or a file's contents, often used to ensure that the file has not been

altered by an intruder or virus. Hash functions are commonly employed by many operating systems to encrypt passwords. Hash functions then provide a measure of the integrity of a file.

Some of the commonly used Hash algorithms used these days are:

1. **Message Digest (MD) Algorithms:** A series of byte-oriented algorithms that produce a 128 – bit hash value from an arbitrary-length message.
2. **Secure Hash Algorithm (SHA) :** Algorithm for NIST's Secure Hash Standard (SHS). SHA-1 produces a 160-bit hash value, SHA-224 produces 224, SHA- 256 produces 256 bit, SHA- 384 produces 384 and SHA-512 produces 512 bits in length.
3. **RIPEMD** : a series of message digests that is optimized for 32-bit processors to replace the then-current 128-bit hash functions.
4. **HAVAL** : Creates hash values that are 128,160,192,224 or 256 bits in length.
5. **Whirlpool :** Operates on a message less that 2^{256} bits in length and produces a message digest of 512 bits.
6. **Tiger :** Runs efficiently on 64-bit processors and produces 192-bit output.

Why do we use three Cryptographic Techniques?

Each of the cryptographic schemes is optimized for some specific applications.

1. **Secret Key cryptography** is ideally suited to encrypting messages, thus providing privacy and confidentiality. The sender can generate a session key on a per-message basis to encrypt the message; the receiver needs the same session key to decrypt the message.
2. **Public key cryptography** is mainly used in Key exchange and Digital Signatures.
3. **Hash Functions** are ideally suited for ensuring data integrity because any change made in the contents of a message will result in the receiver calculating a different hash value than the one placed in the transmission by the sender. Since it is highly unlikely that two different messages will yield the same hash value, data integrity is ensured to a high degree.

6.6 Digital Signatures

In the world of hardcopies, signing a cheque or any message binds the owner of the message to the message, but not so in the digital world. E-mails can be spoofed, manual signatures on scanned documents can be copied, so the problem is how to achieve the properties of a signature in the digital world. Message authentication protects two parties who exchange messages from a third party. However, it does not protect the two parties against each other and several kinds of disputes may arise.

If Aditya sends an authenticated message to Bhaskar, the following disputes may occur between the two:

a. Bhaskar may forge a different message and claim that it came from Aditya.
b. Aditya may deny sending the message.

The signature basically serves three purposes.

1. **Authentication** (establishing the identity of the person who has signed it);
2. **Integrity** (that the document that has been signed is unchanged); and
3. **Non-repudiation** (that the person who has signed it can't deny it later).

On similar lines, a digital signature too serves the above-mentioned three purposes. A **Digital Signature** may be defined as a data string dependent on some secret known only to the signer, and additionally, on the content of the message. Digital signatures are not simply a typed name or image of a handwritten signature. It is based on public-key encryption and is associated with a digital document. Digital signatures must be verifiable, ie., if a dispute arises, an unbiased third party should be able to settle the dispute fairly without accessing the signer's secret.

How do digital signatures help in Authentication?

Let us take the earlier example of Aditya trying to send a document to Bhaskar. Aditya does the following:

1. Encrypts the document using his private key (digitally signs the document)
2. Sends the encrypted (signed) message to Bhaskar
3. Bhaskar decrypts the message using Aditya's public key.
4. The result is that Bhaskar gets the confidence that the Document has been sent by Aditya.

In the case of asymmetric encryption systems, there are two keys, one public and the other private. The private key is kept secret and not revealed under any circumstances. If any message is encrypted using this private key of the sender, it can now be decrypted only by the public key of the sender. The receiver by his ability to decrypt the message using the sender's public key, knows beyond any doubt that the message could have been signed only by the sender. This is equivalent to affixing a signature on a document.

ENCRYPTING ANY DOCUMENT USING THE PRIVATE KEY OF AN INDIVIDUAL (TERMED AS DIGITAL SIGNATURE) IS TREATED EQUIVALENT TO THE INDIVIDUAL SIGNING THE DOCUMENT

How a digital signature achieves Non-Repudiation?

Digital signatures also help in non-repudiation. Since the private key of the individual is only known to them, encryption of a message cannot be done without the concurrence of the sender. Hence at a later point of time, the sender cannot repudiate having sent the message.

However, this process achieves authentication and also Non-Repudiation but does not guarantee message integrity which is provided by a signed document. Hence there is need to protect the integrity of the message.

How digital signatures help in maintaining Integrity ?

In order to maintain integrity of the message, an additional task is being performed on the document. This is called **hashing,** and the result of hashing is a **message digest**. A hash function **H** is one that takes a variable-size input **m** and returns a fixed-size string, called the **Message Digest**.

The basic properties of any hash function are:

- the input string can be of any length.
- the output always has a fixed length.
- The Hash value **H(x)** is relatively easy to compute for any given **x**.
- **H(x)** is one-way, (which implies that given a hash value **h**, it is computationally infeasible to find some input **x** such that **H(x) = h**.)
- **H(x)** is collision-free. (what is meant here is that it is computationally infeasible to find any two messages **x** and **y** such that **H(x) = H(y)**.)

Among the most common hash functions in commercial cryptographic applications are a family of Message Digest (MD) algorithms, all of which produce a 128-bit hash value from an arbitrary-length message. MD2, MD4, MD5 were developed by Rivest. The Secure Hash Algorithm (SHA), proposed by NIST, is another example. SHA produces a 160-bit hash value.

For maintaining the integrity of the message, we go back to the example of Aditya and Bhaskar. To send a message to Bhaskar, Aditya does the following:

1. Subjects the message to a hash function to obtain a message digest.
2. Encrypts the message digest by using his private key (digitally signs the message digest).
3. Attaches the encrypted message digest to the message and sends it to Bhaskar.
4. Upon receiving the message and the encrypted message digest, Bhaskar decrypts the encrypted message digest using Aditya's public key and obtains the message digest.

5. Bhaskar subjects the message to the hash function and obtains a message digest.
6. He then compares the computed message digest with the message digest he has received from Aditya. If both of them are the same, the integrity of the message is established. If the integrity is suspect, then Bhaskar can ask Aditya to resend the message.

Fig. 6.4 shows how the system works.

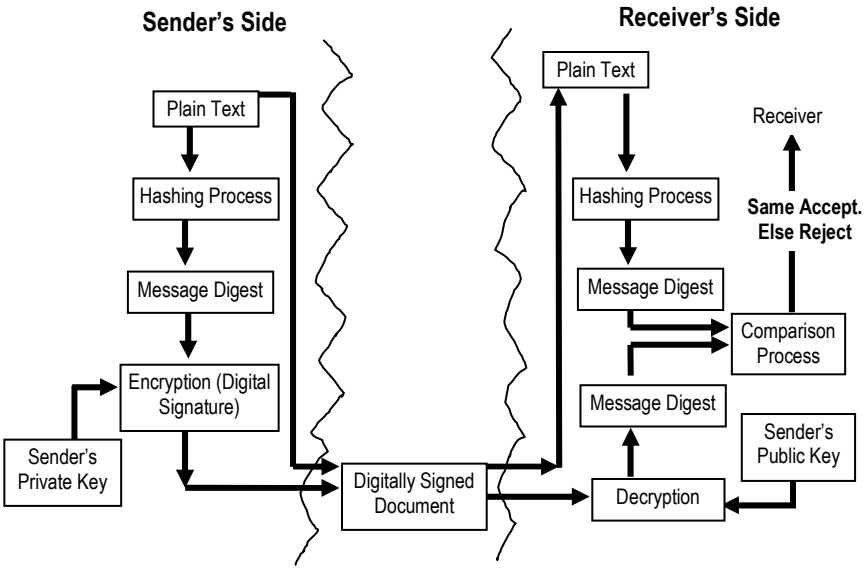


Fig. 6.4: Working of the Digital Signature by Using Hashing

On the basis of above properties, we can formulate the following requirements for a digital signature.

- The signature must be a bit pattern that depends on the message being signed.
- The signature must use some information unique to the sender to prevent both forgery and denial.
- It must be easy to produce the digital signature.
- It must be easy to recognize and verify the digital signature.
- It must be computationally infeasible to forge a digital signature, either by constructing a new message from an existing digital signature or by constructing a fraudulent digital signature for a given message.
- It must be practical to retain a copy of the digital signature in storage.

6.7 Digital Envelopes

The Digital enveloping technology achieves the advantages of confidentiality in addition to identity authentication, message integrity and non repudiation. This is achieved by using both the symmetric and asymmetric encryption techniques. The main advantages of symmetric encryption are faster encryption, efficiency in operation, and consumption of less resource. The main advantage of asymmetric encryption is efficient key management and security.

To explain this method, let us go back to the example of Aditya trying to send a confidential message to Bhaskar. The steps in the creation of a digital envelope are outlined below.

i. Aditya subjects the message to a hashing function. The result is the message digest.
ii. The message digest is then encrypted (signed) using the private key of Aditya (the result is the digital signature)
iii. The plain-text message is encrypted using a randomly chosen symmetric key (encryption of the message). This key is independent of the private and public keys.
iv. The symmetric key used for encryption is encrypted using public key algorithm (Asymmetric Algorithm) and the public key of Bhaskar. (Encryption of the key)
v. All the three, encrypted message, encrypted key and the encrypted message digest are sent to the receiver (Bhaskar).
vi. On receipt of the above, Bhaskar first decrypts the encrypted key using his private key (decryption of the key) and obtains the symmetric key.
vii. He uses the symmetric key so obtained to decrypt the message (decryption of the message).
viii. He uses Aditya's public key to decrypt the message digest (decryption of the message digest).
ix. The plain-text so obtained is subjected to the hash function and the message digests are compared. If they are the same, the message is accepted. .
x. Hence he achieves message integrity, identity authentication, non repudiation and confidentiality.

Fig. 6.5 shows how a digital envelope is created and used.

A **digital envelope** (encryption) is the electronic equivalent of putting a message into a sealed envelope to provide privacy and resistance to tampering. A digital signature is the electronic equivalent of a signet ring and sealing wax: one can seal the message so that the receiver has a high degree of confidence that the message really came from the purported sender and that no one has altered it.

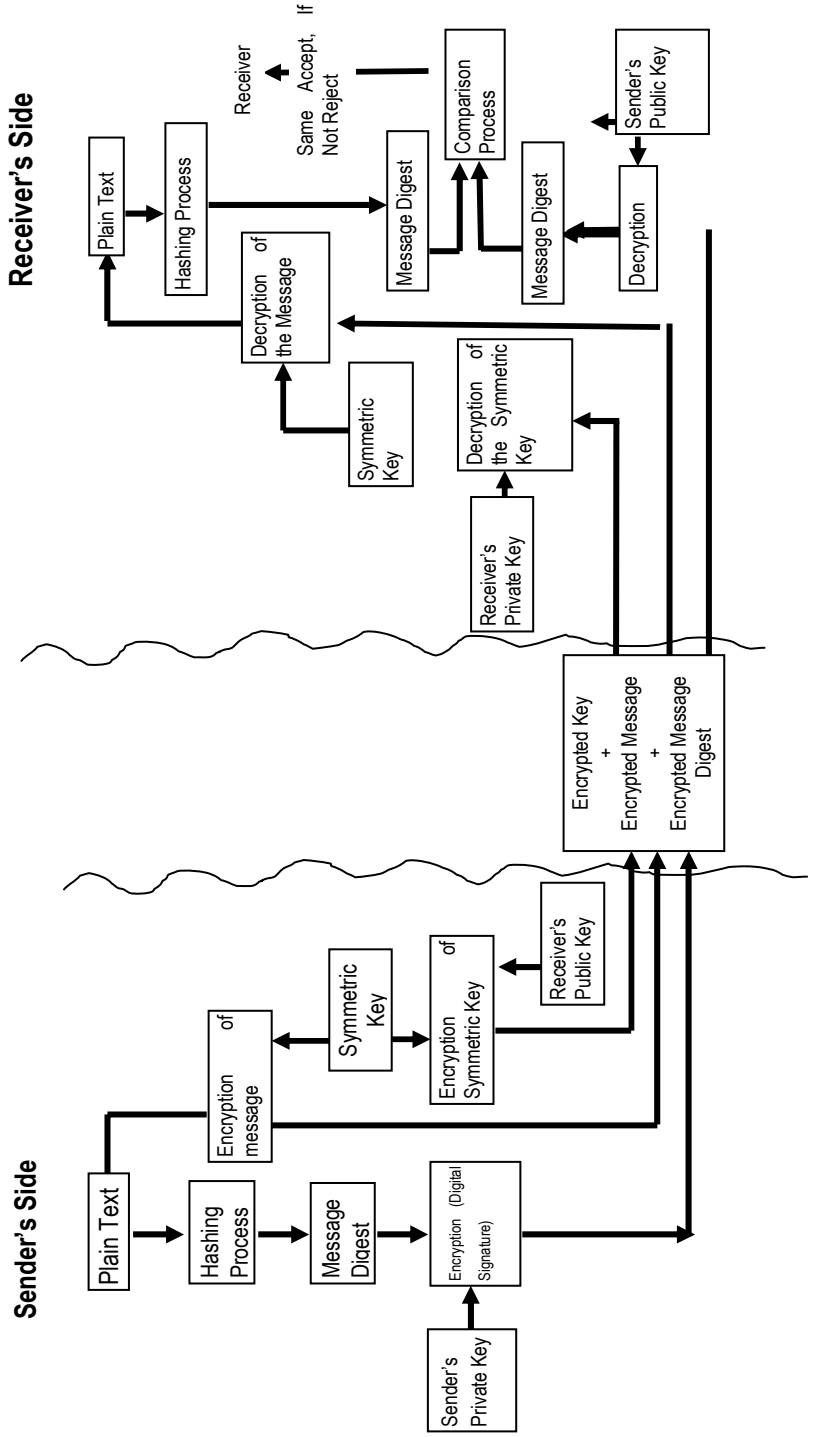


Fig. 6.5: How Digital Envelopes are Created

6.8 Digital Certificates

In cryptography, a **public key certificate** (also known as a **digital certificate** or **identity certificate**) is an electronic document which uses a digital signature to bind together a public key with an identity - information such as the name of a person or an organization, their address, and so forth. The certificate can be used to verify that a public key belongs to an individual.

In a typical public key infrastructure (PKI) scheme, the signature will be of a certificate authority (CA). In a web of trust scheme, the signature is either of the user (a self-signed certificate) or of other users ("endorsements"). In either case, the signatures on a certificate are attestations by the certificate signer that the identity information and the public key belong together.

Contents of a Typical Digital Certificate

- **Serial Number:** Used to uniquely identify the certificate.
- **Subject:** The person or entity identified.
- **Signature:** The algorithm used to create the signature.
- **Issuer:** The entity that verified the information and issued the certificate.
- **Valid-From:** The date from which the certificate is valid.
- **Valid-To:** The expiration date.
- **Public Key:** The public key to encrypt a message to the named subject.
- **Thumbprint Algorithm:** The algorithm used to hash the certificate.
- **Thumbprin**t: The hash itself to ensure that the certificate has not been tampered with.

The importance of digital certificates and Certifying Authorities (CA) is to authenticate that the public key indeed belongs to the subscriber. The reason is that there must be some mechanism to address the trust issues between the buyer and seller in an e-commerce environment. In any e-commerce environment, one has to answer the following:

i. How, for example, does one site obtain another party's public key?
ii. How does a recipient determine if the public key really belongs to the sender?
iii. How does the recipient know that the sender is using their public key for a legitimate purpose for which they are authorised?
iv. When does the public key expire?
v. How can the key be revoked in case of compromise or loss?

The basic concept of a certificate draws analogy from the authentication document such as a passport or driving licence. These attestations provide the bona fides of the certificate holder. Along similar lines, digital certificates, which are digital

documents, provide bona fides for the signor. The specific functions of a digital certificate include:

- **Establishing identity :** Associate, or **bind**, a public key to an individual, organization, corporate position, or other entity.
- **Assign authority :** Establishes what actions the holder may or may not take based upon this certificate.
- **Secure confidential information** (e.g., encrypting the session's symmetric key for data confidentiality).

Typically, a digital certificate contains a public key, a name, an expiration date, the name of the authority that issued the certificate (and, therefore, is vouching for the identity of the user), a serial number, any pertinent policies describing how the certificate was issued and/or how the certificate may be used, the digital signature of the certificate issuer, and any other information.

The most widely accepted certificate format is the one defined in International Telecommunication Union Telecommunication Standardisation Sector (ITU-T) Recommendation X.509. X.509, a specification used around the world. The contents of X.509 certificate are listed below.

Contents of an X.509 V3 Certificate
1. version number
2. certificate serial number
3. signature algorithm identifier
4. issuer's name and unique identifier
5. validity (or operational) period
6. subject's name and unique identifier
7. subject public key information
8. standard extensions certificate appropriate use definition eye usage limitation definition certificate policy information
9. other extensions Application-specific CA-specific

Table 6.1: Contents of X.509 certificate

If we are using the browser of Internet Explorer, we can view the digital certificates if we execute the following steps.

Fig. 6.6: To view digital Certificates using Internet Explorer

a. Launch Internet Explorer.
b. Go to **Tools → Internet Options → Content → Certificates**. Fig. 6.6 can be viewed.
c. Click on **Certificates → Intermediate Certification Authorities**.
d. Choose any one Intermediate Certification Authorities. To illustrate, if we choose Thawte Premium CA (as shown below in Fig. 6.7) and click **View → Details**, we can view the digital certificate.

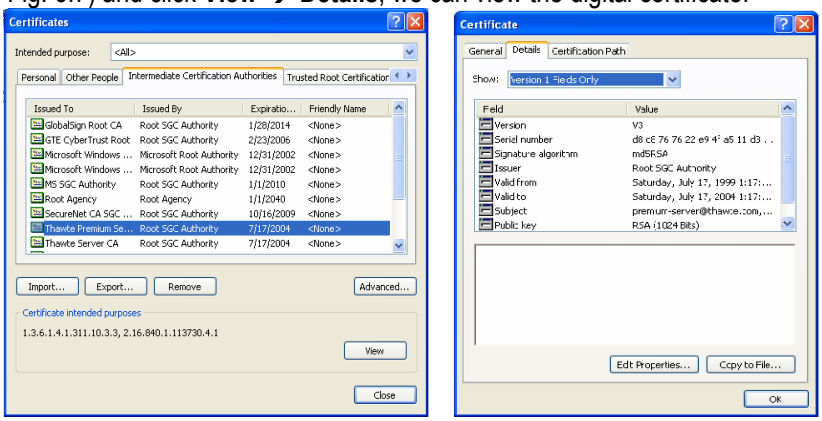


Fig. 6.7: How to view a Digital Certificate?

e. One can also export the digital certificate by clicking the **Copy to file** option.

How are Digital Certificates issued and used?

Let us assume Aditya wants to get a digital certificate. The following are the steps to be adopted.

i. Aditya approaches a certifying authority (CA).
ii. Aditya provides all information that a CA asks for and thereby issues a PUBLIC KEY only.

iii. The CA, after satisfying himself, encrypts the public key of Aditya using their private key and issues a digital certificate.
iv. Aditya uses the digital certificate for all his commercial transactions.
v. When any person wants to verify the digital signature, he needs to click on the digital certificate, which provides verification of the identity of the individual as issued by CA.

6.9 Cryptanalysis

Cryptanalysis mainly deals with methods of recovering the plaintext from ciphertext without using the key. In other words, it is defined as the study of methods for obtaining the meaning of encrypted information, without access to the secret information which is normally required to do so. It also deals with identifying weakness in a cryptosystem. **Cryptanalysis** is also used to refer to any attempt to circumvent the security of other types of cryptographic algorithms and protocols, and not just encryption. There are many types of cryptanalytic attacks. The basic assumption is that the cryptanalyst has complete knowledge of the encryption algorithm used.

1. **Ciphertext-only attack:** The cryptanalyst has the ciphertext of several messages, all of which have been encrypted using the same encryption algorithm. Using the ciphertext, he deduces the key for encryption.
2. **Known-plaintext attack:** Here, the cryptanalyst has access not only to the ciphertext of several messages, but also to the plaintext of those messages. Using the ciphertext and its corresponding plaintext, he deduces the key used to encrypt the messages.
3. **Chosen-plaintext attack:** This is a modified form of plaintext attack. Here, the cryptanalyst not only has access to the ciphertext and associated plaintext for several messages, but he also chooses the plaintext that gets encrypted. Because of this, the cryptanalyst chooses a specific plaintext block to encrypt which in all probability yields more information about the key.
4. **Adaptive-chosen-plaintext attack:** This is the special case of a chosen-plaintext attack. Not only can the cryptanalyst choose the plaintext that is encrypted, but he can also modify his choice based on the results of previous encryption. In a chosen-plaintext attack, a cryptanalyst might just be able to choose one large block of plaintext to be encrypted; in an adaptive-chosen-plaintext attack he can choose a smaller block of plaintext and then choose another based on the results of the first, and so forth.
5. **Chosen-ciphertext attack:** The cryptanalyst can choose different ciphertexts to be decrypted and has access to the decrypted plaintext. For example, the cryptanalyst has access to a tamperproof box that does automatic decryption.

His job is to deduce the key. This attack is primarily applicable to public-key algorithms and sometimes effective against symmetric algorithm as well.

6. **Chosen-key attack:** This attack doesn't mean that the cryptanalyst can choose the key; it means that he has some knowledge about the relationship between different keys.
7. **Rubber-hose cryptanalysis:** The cryptanalyst threatens, blackmails, or tortures someone until they give him the key. Bribery is also adopted sometimes. This is called a **Purchase-key Attack**. These are all very powerful attacks and often the best way to break an algorithm.

Differential and Linear Cryptanalysis

Differential Cryptanalysis: Differential cryptanalysis is a general form of cryptanalysis applicable primarily to block ciphers, but may also apply to stream ciphers and cryptographic hash functions. In a broad sense, it is the study of how differences in input can affect the resultant difference at the output. In the case of a block cipher, it refers to a set of techniques for tracing differences through the network of transformations, discovering where the cipher exhibits non-random behavior, and exploiting such properties to recover the secret key.

Linear Cryptanalysis: Linear cryptanalysis is a general form of cryptanalysis based on finding affine approximations to the action of a cipher. Attacks have been developed for block ciphers and stream ciphers. Linear cryptanalysis is one of the two most widely used attacks on block ciphers; the other being differential cryptanalysis.

There are two parts to linear cryptanalysis. The first is to construct linear equations relating to plaintext, ciphertext and key bits that have a high bias; that is, whose probabilities of holding (over the space of all possible values of their variables) are as close as possible to 0 or 1. The second is to use these linear equations in conjunction with known plaintext-ciphertext pairs to derive key bits.

Summary

This chapter gives detailed knowledge about the concept of Cryptography, its need and the goals achieved by any cryptographic system during data transmission over the network. The chapter discusses various cryptographic algorithms and their advantages and disadvantages with examples. The reader gets a brief idea about the Digital signatures, digital envelopes and digital certificates.

Questions

1. The coded message is known as _____.
   a. Plaintext
   b. Ciphertext

c. Encryption
d. Decrytpion

2. With _____, a single key is used for both encryption and decryption.
   a. Secret key cryptography
   b. Public key cryptography
   c. Asymmetric key cryptography
   d. None of these

3. DES stands for_____.
   a. Digital Encryption Standard
   b. Digital Encryption Symmetry
   c. Data Encryption Standard
   d. Digital Encryption Symmetry

4. _____ is the mathematical function used for encryption and decryption.
   a. Bits
   b. Stream
   c. Block
   d. Cipher

5. PKCS stands for _____.
   a. Public-Key Cryptography Standards
   b. Private-Key Cryptography Standards
   c. Public-Key Cipher Standards
   d. Private-Key Cipher Standards

6. _____ is a mechanism to prove that it is actually the sender who has sent the message.
   a. Integrity
   b. Privacy
   c. Non-repudiation
   d. Authentication

7. _____ mainly deals with methods of recovering the plaintext from ciphertext without using the required key.
   a. Cryptography
   b. Cryptanalysis
   c. Encryption
   d. Decryption

8. IDEA stands for _____.
   a. International Digital Encryption Algorithm
   b. International Data Encryption Algorithm
   c. International Data Encoded Algorithm
   d. International Digital Encoded Algorithm

9. Hash functions are also called _____.
   a. One-way Encryption
   b. Public Key Encryption
   c. Symmetric Key Encryption
   d. Asymmetric Key Encryption

10. ____ is a study of how differences in input can affect the resultant difference in output.
    a. One-way Encryption
    b. Public Key Encryption
    c. Differential Cryptanalysis
    d. Linear Cryptanalysis

11. SHA stands for _____.
    a. Secure Hash Algorithm
    b. Simple Hash Algorithm
    c. Stream Hash Algorithm
    d. None of these

12. ____ is a property that does not permit any person who signed any document to deny it later.
    a. Integrity
    b. Validation
    c. Maintenance
    d. Non-Repudiation

13. Caeser Cipher is also known as _____.
    a. Shift by 2 method
    b. Shift by 3 method
    c. Shift by 4 method
    d. Polyalphabetic Substitution

14. Which of the following is not a hash function?
    a. MD Algorithm
    b. Secure Hash Algorithm
    c. Quantum Cryptography
    d. HAVAL

15. Which one of the following purposes is not served by Digital Certificates?
    a. Authentication
    b. Integrity
    c. Non-Repudiation
    d. Selection

16. _____ is a general form of cryptanalysis based on finding affine approximations to the action of a cipher.
    a. One-way Encryption
    b. Linear Cryptanalysis
    c. Differential Cryptanalysis
    d. Public Key Encryption

17. FIPS stands for _____.
    a. Federal Information Processing Standard
    b. Federal Information Processing Symmetry
    c. Foundation of Information Processing Standard
    d. None of these

18. ____ ensures that no one except the intended receiver can read a message.
    a. Authentication
    b. Integrity
    c. Privacy
    d. Selection

19. A ____ is a person who analyses cryptographic mechanisms.
    a. System analyst
    b. Cryptanalyst
    c. Accountant
    d. Mechanic

20. Adaptive-chosen-plaintext attack is a special case of _____.
    a. Ciphertext-only attack
    b. Known-plaintext attack
    c. Rubber-hose cryptanalysis
    d. Chosen-plaintext attack

Answers:

1 b	2 a	3 c	4 d	5 a	6 c
7 b	8 b	9 a	10 c	11 a	12 d
13 b	14 c	15 d	16 b	17 a	18 c
19 b	20 d				

Sources:

1. William Stallings, Cryptography and Network Security, Principles and Practices, Pearson Prentice Hall, Fourth Edition, 2006.
2. Andrew S. Tanenbaum, Computer Networks, PHI, Fourth Edition, 2003.
3. Saunders D H, Computers Today, McGraw Hill International Edition, New York.
4. Leon A and Leon M, Introduction to Information Systems, Vijay Nicole Private Ltd. Chennai, 2004.
5. Bahrami A, Object Oriented Systems Development -An unified approach, McGraw Hill International edition, New York, 1999.
6. Beck, Leland L, Systems Software, Pearson Education India, New Delhi, 2002.
7. Berson A and Anderson, Sybase and Client-Server Computing, McGraw Hill International edition, New York, 1999.
8. Laudon and Laudon, Management Information Systems, Prentice Hall of India, New Delhi, 1999.
9. Pressman R, Software Engineering, a Practioner's approach, Tata McGraw Hill, New Delhi, 1999.
10. Rumbaugh J, et al, Object Oriented Modeling and Design, Prentice Hall of India, New Delhi, 2002.
11. Sommerville I, Software Engineering, Addison Wesely Publishing.
12. Stallings W, Operating Systems, 4th edition, Prentice Hall of India, New Delhi, 2003.
13. Weber R, Information Systems Control and Audit, Pearson Education India, New Delhi, 2002.
14. Abbey M and Corey M J, Oracle8, A beginners guide, Tata McGraw Hill, New Delhi, 1999.
15. Leon A and Leon M, Database Management Systems, Vikas Publishing, Chennai, 2002.
16. Ramakrishnan R, Database Management Systems, McGraw Hill International Edition, New York, 2001.
17. Ullman J, Widom, A first course on database systems, Pearson Education India, New Delhi, 2002.

18. Weber R, Information Systems Control and Audit, Pearson Education India, New Delhi, 2002.
19. http://www.firstsql.com
20. C.J.Date, An Introduction to Database systems, Third Edition Vol. 1, Narosa Publishing House, New Delhi, Madras, Bombay, Calcutta.
21. Elmsari, Navathe, Somayajulu, Gupta, Fundamentals of Database Systems IV Edition, Pearson Education.
22. Silberschatz, Galvin, Gagne, Operating System concepts VI Edition,Wiley.

Module – II

Protection of Information assets

1 Securing Physical Access

Learning Objectives

- The need for information systems controls
- Physical access to assets, threats and exposures
- Physical access controls and techniques of control
- Audit and evaluation of physical access controls
- Environmental threats and exposures
- Administrative and technical controls for the computing environment
- Audit and evaluation of environmental controls

Introduction to Information Systems Controls

The increasing use of information technology in a large number of organizations has made it imperative that appropriate information systems are put in place in an organization. Information technology covers all key aspects of business processes of an enterprise and has an impact on its strategic and competitive advantage and for its success. The enterprise strategy outlines the approach it wishes to formulate with relevant policies and procedures on harnessing the resources to achieve business objectives.

Control is defined as policies, procedures, practices and enterprise structure that are designed to ensure that business objectives are achieved and undesirable events are either prevented or detected and corrected.

Thus Information Systems (IS) auditing includes reviewing the implemented system or providing consultation and evaluating the reliability of operational effectiveness of controls.

Need for control and Audit of Information systems

Technology has impacted business in terms of information. It has increased the ability of businessmen to capture, store, analyze and process considerable amounts of data and information. Today's dynamic global enterprises need information integrity, reliability and validity for timely flow of accurate information throughout the organization. Making that possible is the key function of the control process.

Factors influencing an organization's control and audit of computers and the impact of the information systems audit function on it are depicted in Figure 1.1

Fig 1.1: Need for control and audit of Information Systems

i. *Organizational Costs of Data Loss*: Data is a critical resource of an organisation for its present and future process and its ability to adapt and survive in a changing environment.
ii. *Incorrect Decision Making*: Management and operational controls taken by managers involve detection, investigations and correction of out-of-control processes. These high level decisions require accurate data.
iii. *Costs of Computer Abuse*: Unauthorised access to computer systems, computer viruses, unauthorised physical access to computer facilities and unauthorised copies of sensitive data can lead to the destruction of assets (hardware, software, and documentation etc of an organization.).
iv. *Value of Computer Hardware, Software and Personnel*: These are critical resources of an organisation which have an impact its infrastructure and

business competitiveness.

v. *High Cost of Computer Error*: In a computerised environment where many critical business processes are performed a data error during entry or process can cause great damage.
vi. *Maintenance of Privacy*: Today data collected in a business process contains details about individuals: about their education, employment, residence, health, etc. Because these are now available on computers, there is a need to ensure their privacy.
vii. *Controlled evolution of computer Use*: Technology use and reliability of complex computer systems cannot be guaranteed and the consequences of using unreliable systems can be destructive.
viii. *Information Systems Auditing*: is the process of attesting objectives (those of the external auditor) that focus on asset safeguarding and data integrity, and management objectives (those of the internal auditor) that include not only to assess objectives but also its effectiveness and efficiency objectives.
ix. *Asset Safeguarding Objectives*: The information system assets (hardware, software, data files etc.) must be protected by a system of internal controls from unauthorised access.
x. *Data Integrity Objectives*: is a fundamental attribute of IS Auditing. The importance to maintain integrity of data of an organisation depends on the value of information, the extent of its access, and the value of data to business from the perspective of the decision maker, levels of competition and market environment.
xi. *System Effectiveness Objectives*: Effectiveness of a system is evaluated by auditing the characteristics and objectives of the system to meet substantial user requirements.
xii. *System Efficiency Objectives*: To optimize the use of various information system resources (machine time, peripherals, system software and man hours) along with their impact on the computing environment.

Objective of Control

Control objective is defined as “A statement of the desired result or purpose to be achieved by implementing control procedures in a particular IT process or activity”. It describes what is sought to be accomplished by implementing control, and serves two main purposes:

i. Outlines the policies of the organization as laid down by the management, and
ii. a benchmark for evaluating if the control objectives are met.

The objective of controls is to reduce or, if possible, eradicate the causes of the exposure to probable loss. All exposures have causes and are potential losses due to threats. Some categories of exposures are:

- Errors or omissions in data, procedure, processing, judgment and comparison.
- Improper authorizations and accountability with regard to procedures, processing, judgment and comparison.
- Inefficieny in procedures, processing and comparison.

Some of the critical control considerations in a computerized environment are:

- Lack of understanding of IS risks and lack of necessary IS and related controls.
- Absence or inadequate IS control framework.
- Absence of or weak general and IS controls.
- Lack of awareness and knowledge of IS risks and controls amongst business users and even IT staff.
- Complexity of implementation of controls in a distributed computing environments and extended enterprises.
- Lack of control features or their implementation in a highly technology driven environment.

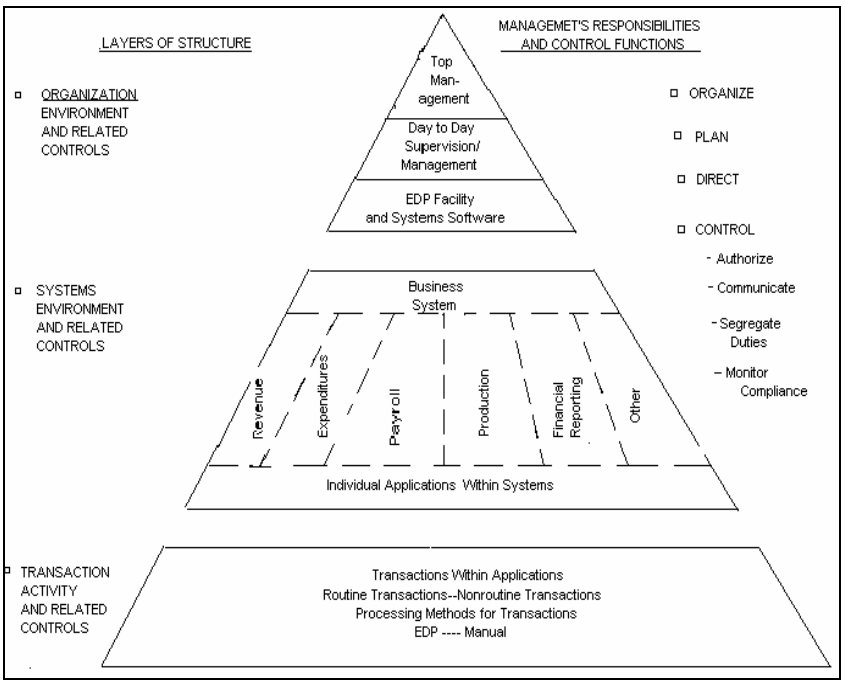


Fig.1.2: Structure of a Control environment.

Internal Controls

The basic purpose of internal control in an organization is to ensure that business objectives are achieved and undesired risk events are prevented or detected and corrected. This is achieved by designing an effective internal control framework, which comprises policies, procedures, practices, and organizational structure. Eventually, all these policies, procedures etc. are broken into discrete activities and supporting processes, which are managed manually or automatically. Control is not solely a policy or a procedure which is performed at a certain point of time; rather it is an ongoing activity, based on the risk assessment of the organization.

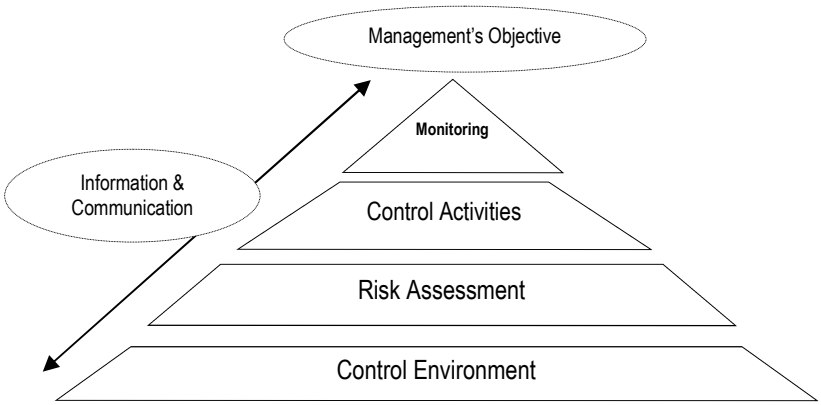


Fig 1.3: Elements of the Internal Control Environment

Types of Internal Controls

Controls can be preventive, detective, or corrective (reactive) and are implemented administratively, technically or physically. Examples of administrative implementations are items such as policies and processes. Technical implementations are the tools and software that logically enforce control (such as passwords) and physical implementation includes controls such as guards and locked rooms.

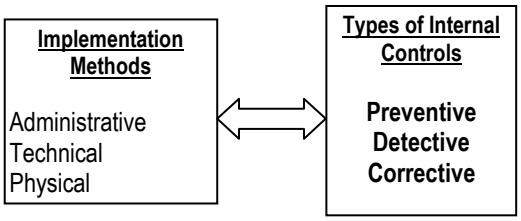


Fig 1.4: Internal Controls and Implementations

i. Preventive Controls

These controls are those inputs, which are designed to protect the organization from unauthorized activities. This attempts to predict the potential problems before they occur and make necessary adjustments. The broad classification of preventive controls is:

- A clear cut understanding about the vulnerabilities of the asset.
- Understanding possible threats.
- Provision of necessary controls for coping with probable threats. Examples of preventive controls include employing qualified personnel, segregation of duties, access control, documentation, etc.

Purpose	Manual Control	Computerized Control
Restrict unauthorized entry into the premises	Build a gate and post a security guard.	Use access control software, smartcard, biometrics, etc.
Restrict unauthorized entry into the software applications	Keep the computer in a secured location and allow only authorized person to use the applications	Use access control, viz. User ID, password, smart card, etc.

Table 1: Preventive Controls

ii. Detective Controls

These controls are designed to detect errors and malicious acts.. An example of detective control is the use of automatic expenditure profiling where management gets regular reports of actual and intended spending. The main characteristics of such controls are:

- A clear understanding of legal activities so that anything that deviates from these is reported as unlawful, malicious, etc.
- An established mechanism to refer the reported unlawful activity to the appropriate person/group.
- Interaction with the preventive control to prevent such acts from occurring.
- Surprise checks by administrators.

Examples of detective controls include - hash totals, check points in production jobs, error message over tape labels, duplicate checking of calculations, past-due accounts report, etc.

iii. Corrective Controls

These controls are designed to correct an error when it is detected. These include the use of default dates on invoices where an operator tries to enter an incorrect date. A business continuity plan is considered a significant corrective control. The main characteristics of the corrective controls are to:

- Minimize the impact of the threat.
- Identify the cause of the problem.
- Remedy the problems spotted by detective controls.
- Get feedback from preventive and detective controls.
- Correct errors arising from problems.
- Modify the processing system to minimize future occurrences of a problem.

Examples of Corrective Controls are contingency planning, backup procedure, rerun procedure, treatment procedures for a crucial error occurrence etc.

For an auditor to effectively evaluate the efficiency of the objectives of these controls, **Table 2** is used for keeping in mind the minimum redundancy in the levels of control. The controls rating by an auditor are:

- ***H**igh:* Controls implemented to deal with exposure/error should be highly effective.
- ***M**oderate:* Controls implemented to deal with exposure/error is moderately effective.
- ***L**ow:* Controls implemented to deal with exposure/error have a low effectiveness.
- ***B**lank*: Controls not implemented or does not exist for that cause or exposure or error type.

Methods of Control Implementation	**Type of Internal control**		
	Preventive	**Detective**	
		With Corresponding Corrective	**Without Corrective**
Manual Control	***B**lank or **L**ow*	***M**oderate*	***B**lank*
	Least effective, generally manual controls applied at front-end of processing;	Moderately effective manual controls; probably least efficient	Least effective and possibly dangerous since users rely on them improperly; very

	moderately efficient		inefficient
Computerized Control	***L**ow or **M**oderate*	***H**igh*	***B**lank*
	Moderately effective, generally Application controls, applied at front-end of processing; probably most efficient	Most effective, generally controls that are computerized and applied before processing can take place; moderately efficient	May have some effectiveness but probably little; highly inefficient

Table 2: Effectiveness and Efficiency of Internal Controls

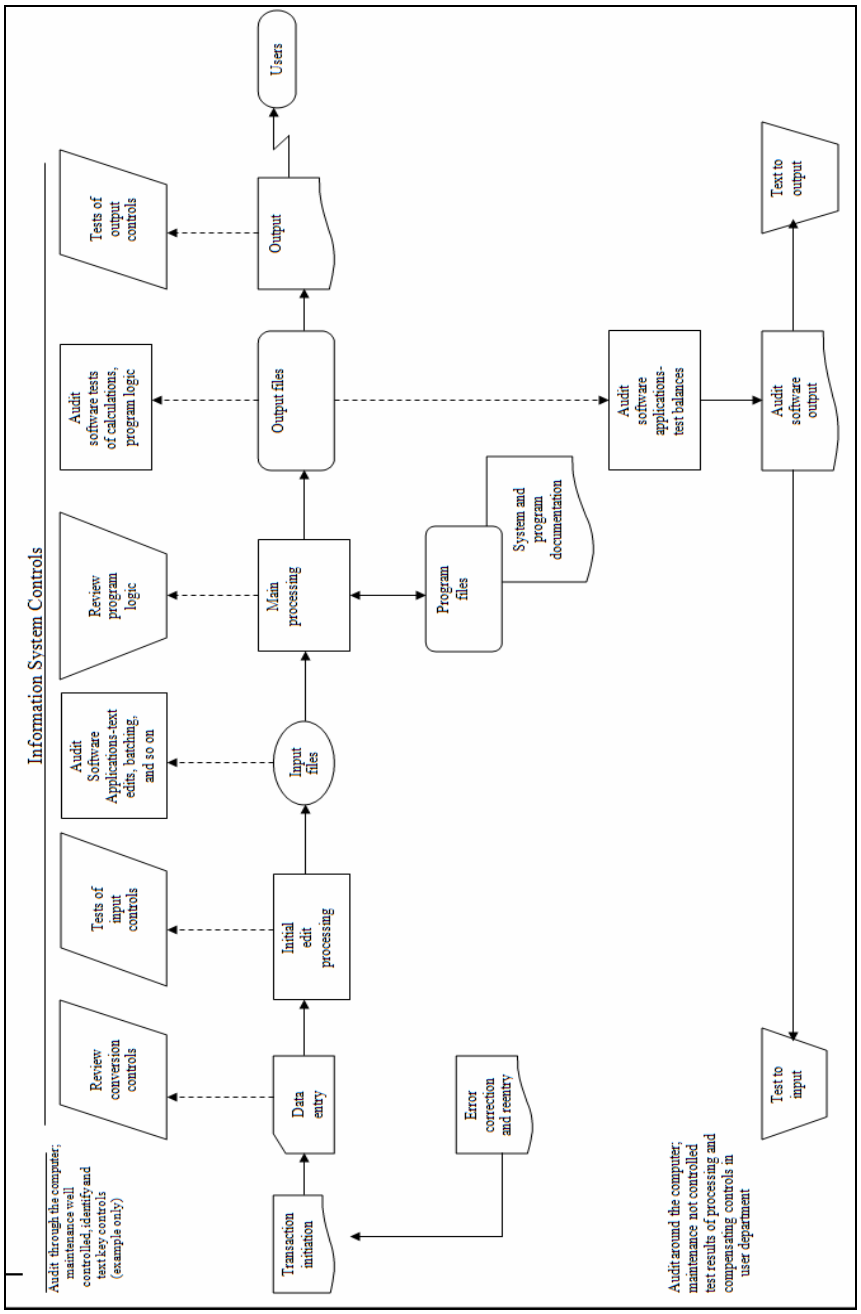


Fig.1.5: Information Systems Controls

IS Assets

A typical computing environment would consist of computers, numerous supporting computing equipment, communications equipment, and the like; facilities which house these computing equipment and infrastructure are computer rooms, power sources, and offsite storage. Further, one would find storage media, documents, computer supplies and documentation related to Information Systems resources. Each of these information system resources may need differing approaches to security both in terms of the techniques of securing them and appropriate investment to secure them, hence the need to categorize such assets. From the perspective of physical access and control, Information System resources may be categorized as follows:

Table 3 lists some of the commonly found Information systems assets found in organizations of various types.

Asset Class	Overall IT Environment	Asset Name
Tangible	Physical Infrastructure	- Data Centers and Servers. - Desktop computers, Mobile computers, Cell Phones, PDAs - Server application and End- user application software - Development Tools - Routers - Network Switches - Fax machines, PBXs, Removable Media, Power supplies. - Fire suppression systems, Air conditioning and other environmental control systems.
Tangible	Intranet Data	- Source code, Human Resources data, Financial and Marketing data. - Employee password, employee and computer system cryptographic keys. - Smart cards and other intellectual property. - Employee biometric identifiers,

	Extranet Data Internet Data	business and personal contact data - Network infrastructure design - Partner financial and contact data, Partner cryptographic keys, credit reports and purchase order data. - Supplier contact data, financial data, contract data, cryptographic keys and purchase order data - Website sales application and marketing data. - Customer contact and credit card data - Press releases, White papers, product documentation
Intangible		- Reputation - Goodwill - Employee moral - Employee productivity
IT Services	Messaging Core Infrastructure	- Instant Messaging - Email/Scheduling - Domain Name System - Dynamic Host Configuration Protocol (DHCP) - Enterprise Management tools - File Sharing, Storage and Dial-up remote access - Telephony, Virtual Private Network (VPN) access

Table 3: Information System Assets

Physical Access

Physical access is a term used in computer security that refers to the ability of people to physically gain access to a computer system. These can be enforced by border guards, doormen, ticket checkers, etc., or with a device such as a turnstile (a gate which ensures one-way traffic of people). There may be fences to avoid circumventing this access control. An alternative to this in the strict sense (physically

controlling access itself) is a system of checking authorized presence. For example, a ticket, controller (Transportation).

Physical access control can be achieved by a human (a guard, bouncer, or receptionist), or through mechanical means such as locks and keys, or through technological means such as access control systems like the Access control vestibule. Within these environments, physical key management may also be employed as a means of further managing and monitoring access to mechanically keyed areas or access to certain small assets.

Access control is the ability to permit or deny the use of a particular resource by a particular entity. Access control mechanisms can be used for managing physical resources (such as a movie theater, to which only ticketholders are admitted), logical resources (a bank account, with a limited number of people authorized to make a withdrawal), or digital resources (for example, a private text document on a computer, which only certain users are able to read).

Objectives of Physical Access Controls

Physical access control is a matter of who, where, and when. The system determines who is allowed, where they are allowed, and when they are allowed to enter or exit. It seeks to safeguard IS resources from physical access exposures. However designing, acquiring and implementing these controls is an expensive proposition.

i. Physical access controls encompass securing physical access to computing equipment as well as facilities housing the IS computing equipment and supplies. The choice of safeguards should be such that they prevent unauthorized physical access and at the same time cause very little inconvenience to authorized users.
ii. Physical access controls restrict physical access to resources and protect them from intentional and unintentional loss or impairment. Assets to be protected include:
    - primary computer facilities,
    - cooling system facilities,
    - microcomputers, and
    - telecommunications equipment and lines, including wiring closets and sensitive areas such as buildings, individual rooms or equipment.
iii. Physical access controls include manual door or cipher key locks, photo Ids and security guards, entry logs, perimeter intrusion locks, etc. The controls are meant to:
    - grant/discontinue access authorizations.
    - control passkeys and entry during and after normal business hours.
    - handle emergencies.

- control the deposit and withdrawals of tapes and other storage media to and from the library.

iv. Physical controls also include:
- pre-planned appointments,
- identification checks,
- controlling the reception area,
- logging in visitors,
- escorting visitors while in sensitive areas, etc.

Physical Access Threats and Exposures

One of the most important steps in any risk management procedure is to identify the threats to any organization. A threat is defined as an event (for example, a theft, fire accident, etc.), which can adversely impact the well being of an asset.

Physical Access Threats to Assets

Physical threats to information system assets comprise threats to computing equipment, facilities which house the equipment, media and people. This section deals with the security of the physical equipment and infrastructure and the environment in which they operate. The focus of the IS Auditor is to examine all factors that adversely affect confidentiality, integrity and availability of information, due to improper physical access. Confidentiality, Integrity and Availability (CIA Triad) are the core principles of information safety.

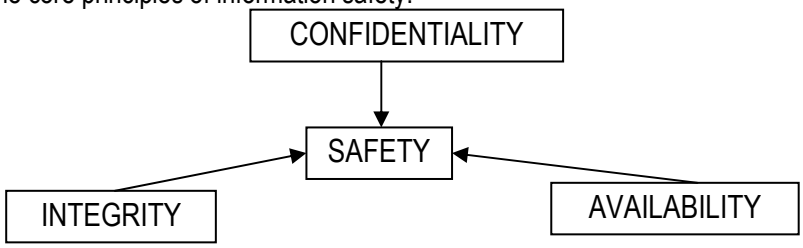


Fig 1.5: Principles of Information Safety

- **Confidentiality:** Preventing disclosure of information to unauthorized individuals or systems.
- **Integrity:** Preventing modification of data by unauthorized personnel.
- **Availability:** Keeping information available for any need.

Physical access threats are of four kinds:

i. **Electrical**: Electrical vulnerabilities are seen in things such as spikes in voltage to different devices and hardware systems, or brownouts due to an insufficient

voltage supply. Electrical threats also come from the noise of unconditioned power or of total power loss.

ii. **Environmental**: These include natural disasters such as fires, hurricanes, tornados, and flooding. Extreme temperature and humidity are also environmental threats. .

iii. **Hardware**: It means a threat of physical damage to corporate hardware or its theft.

iv. **Maintenance**: These threats arise from the poor handling of electronic components, which cause ESD (electrostatic discharge), or because of the lack of spare parts, poor cabling, poor device labeling, etc.

Sources of Physical Access Threats

The sources of physical access threats can be broadly divided into the following:

- Physical access to IS resources by unauthorized personnel.
- Authorized personnel having pre-determined rights of access, misusing their rights in a manner prejudicial to the interests of the organization.
- Authorized personnel gaining access to Information Systems resources in respect of which they have no authorized access. Interested or informed outsiders such as competitors, thieves, organized criminals and hackers.
- Former employees.
- Ignorant people who unknowingly perpetrate a violation.
- Discontented or disgruntled employees.
- Employees on strike.
- Employees under termination or suspended and pending termination.
- Those addicted to drugs or gambling.
- People experiencing financial or emotional problems.

Threats from improper physical access usually are from humans. Some illustrative examples are:

- Unauthorized persons gaining access to restricted areas. Examples are prospective suppliers gaining access to computer terminal of purchase department, viewing the list of authorized suppliers and their rates.
- Employees gaining access to areas not authorized, e.g. sales executives gaining access to the server room.
- Damage, or theft of equipments or other IS resources.
- Abuse of data processing resources, e.g. employees using internet for personal use.
- Damage due to civil disturbances and war.
- Theft of computer supplies, e.g. floppies, cartridges, printers and consumables.

- Public disclosure of sensitive information, e.g. Information regarding location of servers, confidential or embarrassing information.

Physical Access Exposures to Assets

i. **Unintentional or Accidental**: When authorized or unauthorized personnel gain accidental physical access to IS resources, it can cause loss or damage to the organization.
ii. **Deliberate**: Unauthorized r or authorized personnel may gain access to IS resources, to which they have no right of access. This may result in the perpetrator achieving his objective of causing loss or damage to the organization.
iii. **Losses**: Improper physical access to IS resources may result in losses to organization which can result in compromising any one of the following:

- **Confidentiality:** of organizational information or knowledge of protected organizational resources. Example: unauthorized access to systems containing sensitive information may be viewed or copied by visitors gaining access to such systems.
- **Integrity of information:** by improper manipulation of information or data contained on systems or media. Example; Unauthorized access to record rooms or databases may result in modification or deletion of file content.
- **Availability of information:** Improper access to IS resources may be used to adversely impact availability of IS resources' ultimately preventing or delaying access to organizational information and business applications. Example: A disgruntled bank employee may switch off power to information servers, thus sabotaging operations.

Physical Access Control Techniques

Physical access controls are broadly classified into administrative and technical control techniques.

Administrative Controls

i. Choosing and Designing a Secure Site

In the choice of the location during initial planning for a facility the following concerns are to be addressed.

- **Local considerations:** What is the local rate of crime (such as forced entry and burglary).
- **External services:** The relative proximity of local emergency services, such as police, fire, and hospitals or medical facilities.

With respect to designing the site the following considerations apply:

- **Visibility**: Facilities such as data centers should not be visible or identifiable from the outside, that is, there are no windows or directional signs.
- **Windows:** Windows are normally not provided in a structure for data centers. If they are, they must be translucent and shatterproof.

  Doors: Doors in the computer centre must be resistant to forcible entry and have a fire rating equal to that of walls. Emergency exits must be clearly marked and monitored or alarmed. Electric door locks on emergency exits should revert to a disabled state if power outages occur. While this may be considered a security issue, personnel safety always takes precedence, and these doors should have manual operational options in an emergency.

ii. Security Management

- **Controlled user registration procedure**: The right of physical access is given only to persons entitled thereto and, to the extent necessary, based on the principles of least privileges.
- **Audit Trails**: With respect to physical security, audit trails and access control logs are vital because management needs to know where access attempts occurred and who attempted them. The audit trails or access logs must record the following:
  - The date-and time of the access attempt
  - Whether the attempt was successful or not
  - Where from the access was granted (which door, for example)
  - Who attempted the access?
  - Who modified the access privileges at the supervisory level?
- **Reporting and incident handling procedure**: Once an unauthorized event is detected, appropriate procedures should be in place to enable reporting of such incidents and effectively handling them to mitigate losses. The security administrator should be told about such incidents. He may use such history to effect modifications to the security policy.

iii. Emergency Procedures

The implementation of emergency procedures and employee training and knowledge of these procedures is an important part of administrative physical controls. These procedures should be clearly documented, readily accessible (including copies stored of-site in the event of a disaster), and updated periodically.

iv. Administrative Personnel Controls

Administrative personnel controls encompass those administrative processes that are commonly implemented by the Human Resources department during employee hiring and firing. These include pre-employment screening, ongoing employee checks, and post-employment procedures.

Technical Controls

These controls are technical solutions, which have administrative aspects. Given below are various tools and techniques to achieve physical security.

i. **Guards:** Guards are commonly deployed in the perimeter control, depending on the cost and sensitivity of the resources to be secured. While guards are capable of applying subjective intelligence, they are also subject to the risks of social engineering. They are useful whenever immediate, discriminating judgment is required.
ii. **Dogs:** Dogs are used in perimeter security. They are loyal, reliable, and have a keen sense of smell and hearing. However, they cannot make judgment calls the way humans can.
iii. **Compound walls and perimeter fencing:** A common method of securing against unauthorized boundary access to the facility. These help in deterring casual intruders but are ineffective against determined intruders.
iv. **Lighting:** Extensive outside lighting of entrances or parking areas can discourage casual intruders.
v. **Deadman Doors:** These are also called mantrap systems. These are used to secure entrance to sensitive computing facilities or storage areas. This technique involves a pair of doors which function in such a way that only one person can get in at a time. Such doors reduce the risk of piggybacking, in which an unauthorized person enters the secured facility by closely following an authorized person.
vi. **Bolting door locks:** This is the most commonly used means to secure against unauthorized access to rooms, cabins, closets. These use metal locks and keys and access can be gained by any person having physical possession of the key. This is cheap yet a reasonably effective technique, but requires physical custody and inventory of keys. .
vii. **Combination or Cipher locks:** The most common kind of cipher lock consists of a push button panel that is mounted near the door outside of the secured area. There are ten numbered buttons on the panel. To gain entry, a person presses a four digit number in a particular pre-determined sequence which disengages the levers for a preset interval of time. Cipher locks are used

where large number of entrances and exits are used frequently. Such locks (both cipher and combination) enable resetting the unlocking sequence periodically. .

viii. **Electronic Door Locks:** Such locks may use electronic card readers, smart cards readers or optical scanners. The readers or scanners read the cards and upon the information stored on the card matching with the information pre-stored internally in the reader device, the device disengages the levers securing the door, thus enabling physical access. The advantages of such locks are:

- These provide a higher level of security over the previous discussed devices.
- The same device can be used to distinguish between various categories of users.
- Individual access needs can be restricted through the special internal code and sensor devices.
- Restrictions can be assigned to particular doors or at particular hours.
- Duplication of such cards is difficult.
- Card entry can be easily deactivated from a central electronic control mechanism. This is useful in case of cards being lost or for disabling access to terminated employees, etc. This also enables easy implementation of changes to security policy.
- The devices may also include various features such as “card swallow” after preset number of failed attempts, activating audible alarms, engaging other access areas thus securing sensitive areas or trapping the unauthorized entrant.

However, administrative control over card issue, access monitoring, securing electronic communication medium of the mechanism, access to device’s programming mechanism must be ensured for effective security.

ix. **Biometric Door Locks:** These are some of the most secure locks since they enable access based on individual features such as voice, fingerprint, hand geometry, retina or iris. These are similar to the electronic door locks but more sophisticated since in this case the mechanical component securing the physical door is controlled by an electronic device. In this case the device has a scanner/reader which reads the fingerprint or such other biometric and matching individual features with the one that has been internally stored.

While these devices are considered highly secure, they suffer from the following disadvantages:

- Their cost of acquisition, implementation and maintenance is quite high. Because of this, they are used mainly to secure sensitive installations.

- These involve time consuming process of user registration.
- Also involved are privacy issues relating to use of devices like retina and fingerprint scanners.
- Compared to other devices, they have a higher error rate, and may cause false rejection or, more critically. a false acceptance.

x. **Video Cameras: Cameras** provide preventive and detective control. Closed-Circuit Television (CCTV) cameras have to be supplemented by security monitoring and guards for taking corrective action. The location of such cameras and recording/retention of tapes/images for future playback should be decided on the basis of security strategies.

xi. **Identification badges:** Special identification badges such as employee cards, privileged access pass, visitor passes etc. enable tracking the movement of personnel. These can also be signature and/or photo identity cards. These are physically examined by security staff to permit/deny access and detect unauthorized access.

xii. **Manual Logging:** All visitors to the premises are prompted to sign a visitor's log recording the date and time of entry/exit, name of entrant, organization, purpose etc. The visitor may also be required to authenticate his identity by a business card, photo identification card, driver's license, etc.

xiii. **Electronic Logging:** Electronic cards may record the date and time of entry/exit of the card holders by making the card-holders to swipe the card at the time of entry and exit. This is a faster and more reliable method for restricting access to employees and pre-authorized personnel. These devices may use electronic/biometric security mechanisms.

xiv. **Controlled single point access:** Physical access to the facility is granted though a single guarded entry point. Multiple entry points may dilute administration of effective security. This involves identifying and eliminating or disabling entry from all entry points except one.

xv. **Controlled Visitor access**: A pre-designated responsible employee or security staff escorts all visitors such as maintenance personnel, contract workers, vendors, consultants for a specified time period. This is useful in cases where physical access to sensitive facilities is to be given to employees or outsiders such as contract employees.

xvi. **Wireless Proximity Readers.** A proximity reader does not require physical contact between the access card and the reader. The card reader senses the card in possession of a user in the within its scope area or proximity and enables timely access to arise an alarm control.

xvii. **Alarm Systems/Motion detectors.** Alarm systems provide detective controls and highlight security breaches to prohibited areas, access to areas

beyond restricted hours, violation of direction of movement e.g. where entry only/exit only doors are used. Motion detectors are used to sense unusual movement within a predefined interior security area and thus detect physical breaches of perimeter security, and may sound an alarm.

xviii. **Secured Distribution Carts:** One of the issues in batch output control is to get printed hardcopy reports (which may include confidential materials) securely across to the intended recipients. In such cases distribution trolleys with fixed containers secured by locks are used, and the keys to the relevant container are held by the respective user team.

xix. **Cable locks:** A cable lock consists of a plastic-covered steel cable that chains a PC, laptop or peripherals to the desk or other immovable objects.

xx. **Port controls:** Port controls are devices that secure data ports (such as a floppy drive or a serial or parallel port) and prevents their misuse.

xxi. **Switch controls:** A switch control is a cover for the on/of switch, which prevents a user from switching of the file server's power.

xxii. **Peripheral switch controls:** These types of controls are lockable switches that prevent the use of a keyboard.

xxiii. **Biometric Mouse:** The input to the system uses a specially designed mouse, which is usable only by pre-determined/pre-registered person based on the fingerprint of the user.

xxiv. **Laptops Security:** Securing laptops and portables represent a significant challenge, especially since, loss of laptops create loss of confidentiality, integrity and availability. Cable locks, biometric mice/fingerprint/iris recognition and encryption of the file system are some of the means available to protect laptops and their data.

Auditing Physical Access Controls

Auditing physical access requires the auditor to review the physical access risks and controls to form an opinion on their effectiveness. This involves risk assessment, review of documentation and testing of controls.

i. Risk Assessment

The auditor should satisfy himself that the risk assessment procedure adequately covers periodic and timely assessment of all assets, physical access threats, vulnerabilities of safeguards and exposures.

ii. Controls Assessment

Based on the risk profile, the auditor evaluates if physical access controls are in place and adequate to protect the IS assets against risks.

iii. Review of Documentation

Planning for review of physical access controls requires an examination of relevant documentation, such as security policy and procedures, premises plans, building plans, inventory list, cabling diagrams.

iv. Testing of Controls

The auditor should review physical access controls for their effectiveness. This involves:

- A tour of organizational facilities including outsourced and offsite facilities.
- Preparing physical inventory of computing equipment and supporting infrastructure.
- Inspecting/Interviewing personnel for information on the awareness and knowledge of procedures.
- Observing safeguards and physical access procedures. This would also involve inspection of:
  - Core computing facilities.
  - Computer storage rooms.
  - Communication closets.
  - Backup and Off-site facilities.
  - Printer rooms.
  - Disposal yards and bins.
  - Inventory of supplies and consumables.

Some special control considerations also involve the following:

- All points of entry/exit
- Glass windows and walls
- Moveable and modular cubicles
- Ventilation/Air-conditioning ducts
- False Ceiling and flooring panels.

Did you know?

Companies are now allocating surveillance technology in the wrong places, and not where intruders are more likely to gain access, such as the cargo landing where smokers take their breaks, or on the cafeteria patio.

Leaving physical access to chance in these areas makes it much easier for an attacker to simply walk in resulting in a security breach.

Review of Physical access procedures includes user registration and authorization, special access authorization, logging, periodic review, supervision, etc. Employee termination procedures should provide withdrawal of rights such as retrieval of physical devices such as smart cards, access tokens, deactivation of access rights and its appropriate communication to relevant constituents in the organization.

Examination of physical access logs and reports includes examination of incident reporting logs and problem resolution reports.

Few examples of Physical Control Techniques and their Suggested Audit Procedures

Control Activities	Control Techniques	Audit Procedures
Physical safeguards to commensurate with the risks of physical damage or access.	Identify facilities housing sensitive and critical resources. Identify all threats to physical well-being of sensitive and critical resources by adequately secured keys, alarm systems, security devices and other access control devices, including- - the badging system. - display and output devices. - data transmission lines. - power equipment and poser cabling. - Mobile or portable systems. All deposits and withdrawals of tapes and other storage media from the library are authorized and logged. Emergency exit and reentry procedures ensure that only authorized personnel are allowed to reenter after fire drills, etc.	Review the physical layout diagram of computer, telecommunications and cooling system facilities. Walk through facilities. Review risk analysis. Review procedures for the removal and return of storage media from and to the library. Review of written emergency procedures. Observe a fire drill. Review the knowledge and awareness of emergency procedures among employees with respect to facilities using interviews, questionnaires etc.

Establish adequate security at entrance and exits based on risk	All employee access is authorized and credentials (badges, ID cards, smart cards) are issued to allow access. Management conducts regular review of individuals with physical access to sensitive facilities. Visitors to the sensitive areas, such as the main computer room and tape/ media library, are formally signed in and escorted. Entry codes are changed periodically.	Review procedures and logs of employee entry and exits during and after normal business hours. Review Procedures used by management to ensure that individuals having access to sensitive facilities are adequately restricted and possess physical access authorization. Review visitor entry logs. Interview guards at the entry point. Review documentation on logs of entry, code changes and system maintenance.
Perimeter Security	Control/restrict vehicle and pedestrian traffic with measures like fences, gates, locks, guard posts and inspections. Installation of closed circuit system with recording and warning alarms round the clock.	Assess vehicle and pedestrian traffic around high risk facility. Inspect guard procedures and practices for controlling access to facility grounds. Inspect the facility surveillance system to assess its capability for protecting the facility.
Security control policies and procedures are documented, approved and implemented by management.	Security control policies and procedures at all levels- -Are document -Address purpose, scope, roles, responsibilities and compliance. -Ensure users can be held	Review security policies and procedures at the enterprise level, system level and process level are aligned with business/enterprise stated objectives.

	accountable for their actions. -are approved by management and - Periodically reviewed and updated.	

Environmental Access Controls

This section examines the risks to IS resources arising from undesired changes in the environment. Environmental threats to information assets include threats to facilities and supporting infrastructure, which house and support the computing equipment, media and people. IS Auditor should review all factors that adversely bear on the confidentiality, integrity and availability of information, due to undesired changes in the environment or ineffective environmental controls.

Objectives of Environmental Controls

The objects to be protected from environmental threats are almost the same as discussed in the section on physical access controls. However, from the perspective of environmental exposures and controls, information systems resources may be categorized as follows (with focus primarily on the housing of facilities:

i. **Hardware and Media:** These include Computing Equipment, Communication equipment, and Storage Media
ii. **Information Systems Supporting Infrastructure or Facilities:** These include the following:
    1. Physical Premises, like Computer Rooms, Cabins, Server Rooms/Farms, Data Centre premises, Printer Rooms, Remote facilities and Storage Areas
    2. Communication Closets
    3. Cabling ducts
    4. Power Source
    5. Heating, Ventilation and Air Conditioning (HVAC)
iii. **Documentation:** Physical and geographical documentation of computing facilities with emergency excavation plans and incident planning procedures.
iv. **Supplies:** Third party maintenance procedures for say air-conditioning, fire safety, and civil contractors whose entry and access with respect to their scope of work assigned are to be monitored and logged.
v. **People***:* Employees, contract employees, visitors, supervisors and third party maintenance personnel are to be made responsible and accountable for environmental controls in their respective information processing facility(IPF).

Training of employees and other stake holders on control procedures is a critical component.1.8.2 Environmental Threats and Exposures

Undesired or unintentional or intentional alteration in the environment in which computing resources function can result in threats to the availability of information systems and integrity of information. Exposures from environmental threats include total or partial loss of computing facilities, equipment, documentation and supplies causing loss or damage to organizational data and information and more importantly people. The threats can be broadly classified as Natural and Man-made.

i. Natural Threats

Threats to facilities and environment from natural causes include:

- Natural disasters such as earthquakes, foods, volcanoes, hurricanes and tornadoes
- Extreme variations in temperature such as heat or cold, snow, sunlight, etc.
- Static electricity
- Humidity, vapors, smoke and suspended particles
- Insects and organisms such as rodents, termites and fungi
- Structural damages due to disasters

ii. Man-made Threats

These can be unintentional or intentional. Some examples are:

- Fire due to negligence and human action
- War and nomb threats
- Uncontrolled/(unconditioned?) power, spikes, surges, low voltage
- Equipment failure
- Failure of Air-conditioning, Humidifiers, Heaters
- Food particles and residues, undesired activities like smoking in computer facilities. Structural damages due to human action/inaction and negligence
- Electrical and Electromagnetic Interference (EMI) from generators and motors.
- Radiation
- Chemical/liquid spills or gas leaks due to human carelessness or negligence

iii. Exposures

Some examples of exposures from violation of environmental controls:

- A fire could destroy valuable computer equipment and supporting infrastructure and invaluable organizational data. Usually the use/ storage of thermocole or Styrofoam (technically called Expanded Polystyrene) material, inflammable material used for construction of the server cabin, false ceiling aggravate the probability of fire and loss due to fire.

- Magnetic tapes use materials that are inflammable.
- Poor quality of power cables can over-heat and cause fire.
- Lightening may burn up communication devices and computing equipment due to improper earthing or grounding.
- Continuous process systems bear the risk of internal component damage due to improper air conditioning or high humidity.
- Damage of keyboards and other computing devices can be caused by accidental dropping of beverages, liquid, etc.
- The organizational policies do not check the consumption of food, tobacco products near computer equipments resulting in food particles leftover in computer facilities that attract rodents and insects, which can damage cabling and hard disks.
- Chemical or liquid spills from a nearby unit may seep into the IPF (Information Processing Facility) thereby damaging equipment.
- Sudden surges in power or other voltage fluctuations can irreversibly damage computer equipment.
- Fungi formation on tapes can lead to tapes and disks being not readable.
- EMI (Electromagnetic Interference) from generators can damage integrity of contents on magnetic media.
- Water leakages can induce shocks and short circuits.

Techniques of Environmental Controls

The IS supporting infrastructure and facilities not only provide a conducive environment for the effective and efficient functioning of the information processing facility (IPF) but also protect the contents of such facilities from undesirable variations in the environment.

Based on the risk profile, computing equipment, supporting equipment, supplies, documentation and facilities should be appropriately situated, protected to reduce risks from environmental threats and hazards or exposures.

These control techniques are broadly classified into Administrative and Technical. .

Administrative Controls

i. Choosing and designing a safe site

The considerations during choosing a location for the facility are (as discussed in the section on Physical Access Controls).

- **Natural disasters.** Probability of natural disasters as compared to other locations? Natural disasters can include weather-related problems (wind,

snow, flooding, and so forth) and earthquake faults.

- **Transportation.** Does the site have a problem due to excessive air, highway, or road traffic?
- **External services**. Relative proximity of the local emergency services, such as police, fire, and hospitals or medical facilities is to be factored while choosing a site.

Considerations during designing a site are:

- **Walls:** Entire walls, from the floor to the ceiling, must have an acceptable fire rating. Closets or rooms that store media must have a high fire rating.
- **Ceilings:** Issues of concern regarding ceilings are weight-bearing and fire rating.
- **Floors:** If the floor is a concrete slab, the concerns are the physical weight it can bear and its fire rating. If it is a raised flooring, the fire rating, its electrical conductivity (grounding against static build-up), and that it employs a non-conducting surface material are major concerns. Electrical cables must be enclosed in metal conduits, and data cables must be enclosed in raceways, with all abandoned cables removed. Openings in the raised floor must be smooth and nonabrasive, and they should be protected to minimize the entrance of debris or other combustibles. Ideally, an IPF should not be located between floors and not at or near the ground floor, nor should it be located at or near the top floor.
- **Windows:** Windows are normally not acceptable in a data centre. But if they are there, they must be translucent and shatterproof.
- **Doors:** Doors in the computer centre must resist forcible entry and have a fire rating equal to that of the walls. Emergency exits must be clearly marked and monitored or alarmed. Electric door locks on emergency exits should revert to a disabled state if power outages occur to enable safe evacuation. While this may be considered a security issue, personnel safety always takes precedence, and these doors should be manned or manually operational in case of an emergency.
- **Media Protection:** Location of media libraries, fire proof cabinets, and different kind of media used are to be protected in a fungi resistant and heat resistant environment.
- **Sprinkler system and fire resistance:** The fire-resistance rating of construction material is a major factor in determining the fire safety of a computer operations room. Generally, the computer room must be separated from other occupancy areas by a basic constructional plan with a fire-resistant rating of not less than two hours.

- **Water or gas lines:** Water drains should be "positive;" that is, they should flow outward, away from the building, so they do not carry contaminants into the facility.
- **Air conditioning:** AC units should have dedicated power circuits. Similar to water drains, the AC system should provide outward, positive air pressure and have protected intake vents to prevent air carrying toxins from entering the facility.
- **Electrical requirements:** The facility should have established backup and alternate power sources.

ii. Facilities Planning

As part of planning facilities, the security policy should provide for specific procedures for analysis and approval of building and refurbishment plan. Depending on the size and nature of computing facilities, a separate function should exist for facilities planning and management. The following aspects need to be considered in this context:

- As part of environmental security clearance, procedures should be prescribed to ensure that all aspects relating to facilities planning are adequately considered.
- Approved list of materials to be used for construction of facilities, based on the class of computing facilities, the specification of equipment to be housed in such facility need consideration.
- The organization chart should provide for designated personnel assigned with the responsibility of risk assessment procedures as a dynamic function.
- The risk profile of the organization should take into consideration newer environmental threats. A few examples of threats to be considered are given below:
  - Installation of a generator by a neighbor.
  - Sudden changes in climate leading to extreme changes in humidity levels.
  - Building construction in the vicinity of IPF leading to increase in suspended dust particles in the environment.
  - Raising of foundation and flooring by a neighbor causing change in the flow of rainwater.
  - Installation of high power consumption equipment adversely affecting the quality of power.

iii. Documentation

The documentation of physical and geographical location and arrangement of computing facilities and environmental security procedures should be modified promptly for any changes. Access to such documentation should be strictly

controlled. For example, knowledge of location and scheme of ventilation ducts can be used by a perpetrator to gain unauthorized entry to sensitive facilities which otherwise may be secured by physical access controls.

iv. People Responsibility and Training

Responsibility and accountability for environmental controls planning and management should be well documented and communicated as part of basic job description.

Awareness and training initiatives should encompass educating employees and stakeholders on environmental exposures and controls and their responsibilities in this regard. New employee induction programs should include informing and educating employees on environmental control procedures, prohibited activities (eating, smoking, drinking inside IPF), and maintaining secrecy and confidentiality. Care should also be taken to ensure that sharing such information should not result in risks, where unauthorized persons gain knowledge of sensitive environmental control vulnerabilities.

v. Emergency Plan

Disasters result in increased environmental threats e.g. smoke from a fire in the neighborhood or in some other facility of the organization would require appropriate control action, evacuation plan should be in place and evacuation paths should be prominently displayed at strategic places in the organization.

- *Reporting procedures* should be in place to enable and support reporting of any environmental threats to a specified controlling authority.
- *Periodic inspection, testing and supervision* of environmental controls should form a part of the administrative procedures. The tests of such inspection, tests and drills should be escalated to appropriate levels in the organization.
- *Documented and tested emergency evacuation plans* should consider the physical outlay of the premises and orderly evacuation of people, shutting down of power and computer equipment and activation of fire suppression systems.
- *Administrative procedures* should also provide for Incident Handling procedures and protocols due to environmental exposures.

vi. Vendors/Suppliers (Third Party)

In most cases installation and maintenance of environmental controls involve the services of third parties (such as air conditioning, fire safety equipment, these are not parties like carpenters, so rethink) civil contractors, and carpenters. By virtue of their scope of work, knowledge of and access to sensitive computing facilities and environmental control vulnerabilities are available to such agencies. Procedures

should include detailed analysis of considerations such as, whether to outsource, choice of such an agency, background verification, security bonding, controlled access of maintenance staff and performance appraisal.

vii. Maintenance Plans

A comprehensive maintenance and inspection plan is critical to the success of environmental security and controls. Preventive maintenance plan and management procedures should be in place. This is a critical aspect of environmental control procedures, negligence in respect of which can lead to exposing the IPF to risks, e.g. prolonged ineffectiveness/failure of air conditioning facility can lead to risks of damage to servers and thereby loss of organizational data; a fire extinguisher not working at the time of disaster due to negligence in refilling and maintenance. Maintenance plans should also include evaluation of effectiveness and efficiency of environmental facilities such as electric power distribution, heating plants, water, sewage, and other utilities required for system operation or staff comfort. Environmental controls should be documented and a suitable preventive maintenance should be put in place and administered through schedules and logs.

viii. MTBF and MTTR

Failure modes of each utility and risks of utility failure should be identified, parameterized and documented. This includes estimating the MTBF (Mean Time Between Failures) and MTTR (Mean Time To Repair/recover/respond/ restore). Planning for Environmental controls would require evaluating alternatives with low MTBF or installing redundant units. Stocking spare parts on site and training maintenance personnel can reduce MTTR. Each of these strategies can be evaluated by comparing the reduction in risk with the cost to achieve it.

Technical Controls

Some of the techniques for implementing control to protect against environmental risks are:

i. **Fire-resistant Walls, Floors and Ceilings:** The construction of IPF should use fire-resistant materials for walls, floors and ceilings. Depending on application and investment, manufacturers offer materials with varied fire ratings. Fire rating resistance of at least 2 hours is generally recommended.
ii. **Concealed Protective Wiring:** Power and Communication cables should be laid in separate fire resistant panels and ducts. The quality rating of power cables should match the load and manufacturer specifications
iii. **Ventilation and Air Conditioning**: The temperature in the IPF should be controlled depending on the type of equipment and processing. Improper

maintenance of temperature leads to overheating of internal components. It should be examined if uninterruptible powers supply systems should supply power HVAC equipment that supports critical IPF units.

iv. **Power Supplies:** Computing equipment can be subject to risks of power failure and other power anomalies. Power supply should conform to computing equipment manufacturer specifications. Many elements can threaten power systems, the most common being noise and voltage fluctuations. Noise in power systems refers to the presence of electrical radiation in the system that is unintentional and interferes with the transmission of clean power. There are several types of noise, the most common being electromagnetic interference (EMI) and radio frequency interference (RFI). Voltage fluctuations are classified as Sag (momentary low voltage), Brownout (prolonged low voltage), and Spike (momentary high voltage), Surge (prolonged high voltage) and Blackouts (complete loss of power).

Some of the controls to ensure uninterrupted delivery of clean power are:

i. **Uninterruptible Power Supply (UPS)/ Generator:** UPS usually consist of battery backup or kerosene powered generator that interfaces with the external power supplied to the equipment. On interruption in external power supply, the control is immediately switched to the battery back-up. Depending on the application, UPS are available with battery backup of a few minutes to a number of days. UPS generally is a good solution in case of applications which require proper closure of processing and systems.

   In respect of continuous process equipments, UPS may fail to meet the purpose if regular power supply is not available for a prolonged period of time. Diesel or kerosene generators can also be used, but they require some time to be switched on and the power from generators has to be cleansed before delivery to computer systems.

ii. **Electrical Surge Protectors/Line Conditioners:** Power supply from external sources such a grid and generators are subject to many quality problems such as spikes, surges, sag and brown outs, noise, etc. Surge protectors, spike busters and line conditioners cleanse the incoming power supply and deliver clean power fit for the equipment.

iii. **Power leads from two sub-stations:** Failure of continued power supply to some high consumption continuous processing could even result in concerns regarding public safety such as refineries, nuclear reactors and hospitals. Electric power lines may be exposed to many environmental and physical threats such as foods, fire, lightning, careless digging, etc. To protect against such exposures,

redundant power lines from a different grid supply should be provided for. Interruption of one power supply should result in the system immediately switching over to the stand-by line.

iv. **Smoke Detectors and Fire Detectors:** Smoke and fire detectors activate audible alarms or fire suppression systems on sensing a particular degree of smoke or fire. Such detectors should be placed at appropriate places, above and below the false ceiling, in ventilation and cabling ducts. In case of critical facilities, such devices must be linked to a monitoring station (such as a fire station). Smoke detector should supplement and not replace fire suppression systems.

v. **Fire Alarms:** Manually activated fire alarm switches should be located at appropriate locations that are prominently visible and easily accessible in case of fire (but should not be easily capable of misuse during other times). By manual operation of switch or levers, these devices activate an audible alarm and may be linked to monitoring stations both within and/or outside the organization.

vi. **Emergency Power Of: To take care of** the necessity of an immediate power shutdown during situations such as computer facility fire or emergency evacuation, emergency power-of switches should be provided. There should be one within the computer facility and another just outside it. Such switches should be easily accessible and also properly shielded to prevent accidental use.

vii. **Water detectors:** Risks to IPF equipment from flooding and water logging can be controlled by the use of water detectors placed under false flooring or near drain holes. Water detectors should be placed on all unattended or unmanned facilities. Water detectors on detecting water activate an audible alarm

viii. **Centralized Disaster monitoring and control Systems:** Such systems provide for an organization-wide network control wherein all detection devices, alarms and corrective/suppression devices are controlled from a central monitoring command and control. It is necessary that such systems are powered by a secure and reliable/uninterrupted power supply. Such systems should be failure tolerant and should involve low maintenance

ix. **Fire Suppression Systems**Combustibles are rated as either Class A, B, or C based upon their material composition, thus determining which type of extinguishing system or agent is used. Fires caused by common combustibles (like wood, cloth, paper, rubber, most plastics) are classed as Class A and are suppressed by water or soda acid (or sodium bicarbonate). Fires caused by flammable liquids and gases are classed as Class B and are suppressed by Carbon Dioxide (CO), soda acid, or Halon. Electrical fires are classified as Class C fires and are suppressed by Carbon Dioxide (CO), or Halon. Fire caused by

flammable chemicals and metals (such as magnesium and sodium) are classed as Class D and are suppressed by Dry Powder (a special smothering and coating agent). Class D fires usually occur in places like chemical laboratories and rarely in office environments. Using the wrong type of extinguisher while suppressing a fire can be life-threatening.

Broadly, Fire Suppression systems for facilities are classed into a. Water based systems and b. Gas based systems

a. Water Based Systems

- **Wet Pipe Sprinklers:** In this case, sprinklers are provided at various places in the ceiling or on the walls and water is charged in the pipes. As generally implemented a fusible link in the nozzle melts in the event of a heat rise, causing a valve to open and allowing water to flow. These are considered the most reliable. However, they suffer from the disadvantage of leakage, breakage of pipes exposing the IPF to the risks of dampness and equipment to water damage.
- **Dry-Pipe Sprinklers**: These are similar to the wet pipe sprinklers except that in this the water is not kept charged in pipes but pipes remain dry and upon detection of heat by a sensor, water is pumped into the pipes. This overcomes the disadvantage with wet pipe systems of water leakages etc.
  Pre-action: At the present time, this is the most recommended water-based fire suppression system for a computer room. It combines both the dry and wet pipe systems by first releasing water into the pipes when heat is detected (dry pipe) and then release the water flow when the link in the nozzle melts (wet pipe). This feature enables manual intervention before a full discharge of water on the equipment occurs.

b. Gas Based Systems

- **Carbon-dioxide:** Such systems discharge CO and effectively cut off oxygen supply from the air, which is a critical component for combustion. However, CO being potentially lethal for human life, such systems are recommended only in unmanned computer facilities or in portable or hand-held fire extinguishers. Portable fire extinguishers commonly contain CO or soda acid and should be commonly located at exits, clearly marked with their fire types and checked regularly by licensed personnel.
- **Halon:** was once considered the most suitable agent for fire suppression. It is an inert gas, does not damage equipment as water systems do and does not leave any liquid or solid residues. However, Halon is not considered safe for humans beyond certain levels of concentration and is an ozone-depleting agent and is therefore environmentally unfriendly. Under an international agreement, the Montreal Protocol, the production of Halon was suspended in 1994.

Integration and Fine Tuning of Environmental Controls

As part of environmental risk assessment, facilities planning and facilities management, it is critical to consider the overall effectiveness and efficiency of controls. Planning for Environmental Controls should consider interdependencies in respect of nature of IS assets being secured, vulnerabilities in respect of such assets and nature of other controls such as logical and physical controls. The Security Policy should orchestrate the overall design, effectiveness and efficiency of controls to ensure that investment in environmental controls is optimum, and there is no compromise on security.

Audit and Evaluation of Environmental Controls

Audit of environmental controls should form a critical part of every IS audit plan. The IS Auditor should satisfy himself or herself not only as regards the effectiveness of various technical controls but by assuring himself or herself that the overall controls assure safeguarding the business against environmental risks. Some of the critical audit considerations that an IS Auditor should take into account while conducting his audit is given below:

Audit Planning and Assessment

- As part of Risk assessment, the risk profile should include all kinds of environmental risks that the organization is exposed to, which include taking stock of both natural and man-made threats.
- The profile should be periodically reviewed to ensure updating the profile with new risks that may have arisen.
- The Controls assessment should include examining that controls are in place to safeguard the organization against all acceptable risks and should include newer risks.
- The Security Policy of the organization should be reviewed to assess that policy and procedures for safeguarding the organization against environmental risks are adequately covered.
- The building plans, wiring plans, surroundings, power and cable wiring etc. should be reviewed to determine their appropriateness of location of IPF.
- The IS auditor should interview relevant personnel to satisfy himself regarding employee awareness of environmental threats and controls, role of the interviewee in environmental control procedures such as prohibited activities in IPF, incident handling, evacuation procedures, and to assess if adequate incident reporting procedures exist.
- Review of administrative procedures such as preventive maintenance plans and their implementation, incident reporting and handling procedures, inspection and

testing plan and procedures.

Table : Environmental Exposure Assessment Matrix

Environmental Exposure	Area of management's environmental Responsibility				Area of managements control function			
	Organize	Plan	Direct	Control	Authorize	Segregate Duties	Communicate	Monitor Compliance
Erroneous recordkeeping	*H*	*M*	*L*	*M*	*H*	*L*	*H*	*M*
Unacceptable accounting	*H*	*L*	*M*	*L*	*H*	*M*	*M*	*M*
Loss or asset destruction	*L*	*H*	*L*	*M*	*M*	*H*	*L*	*M*
Business interruptions	*H*	*L*	*M*	*M*	*M*	*L*	*L*	*M*
Erroneous management decisions	*M*	*H*	*H*	*L*	*H*	*L*	*H*	*M*
Excess cost or deficient revenues	*M*	*L*	*L*	*L*	*M*	*L*	*L*	*L*
Standards compliance	*L*	*L*	*L*	*M*	*H*	*L*	*M*	*M*
Fraud and misuse	*L*	*H*	*L*	*L*	*H*	*H*	*M*	*H*
Unachieved process objectives	*M*	*L*	*M*	*M*	*H*	*L*	*M*	*M*

The ranking of the matrix is as follows:

- **High (H):** Very critical controls required to prevent this exposure.
- **Medium M):** Controls are required to prevent this exposure along with accompanied by personnel responsibility.

- **Low (L):** Controls are useful but not fully effective in preventing exposure.

Audit of technical controls

As part of audit procedures, the audit of environmental controls requires the IS auditor to conduct physical inspections and observe practices, which may include the following activities:

- Inspect the IPF and examine the construction with regard to the type of materials used for construction by referring to appropriate documentation.
- Visually examine the presence of water and smoke detectors, examine power supply arrangements to such devices, testing logs, etc.
- Examine location of fire extinguishers, fire fighting equipment and refilling date of fire extinguishers and ensure their adequate and appropriate maintenance.
- Examine emergency procedures, evacuation plan and marking of fire exits. If considered necessary, the IS Auditor can also require a mock drill to test the preparedness with respect to disaster.
- Examine documents for compliance with legal and regulatory requirements as regards fire safety equipment, external inspection certificate, and shortcomings pointed out by other inspectors/auditors.
- Examine power sources and conduct tests to assure quality of power, effectiveness of power conditioning equipment, generators, simulate power supply interruptions to test effectiveness of back-up power.
- Examine environmental control equipment such as air-conditioners, dehumidifiers, heaters, ionizers, etc.
- Examine complaint logs and maintenance logs to assess if MTBF and MTTR are within acceptable levels.
- Observe activities in the IPF for any undesirable d activities such as smoking, consumption of eatables, etc.

A. Documentation of findings

As part of the audit procedures, the IS auditor should also document all findings as part of working papers. The working papers could include audit assessment, audit plan, audit procedure, questionnaires, and interview sheets, inspection charts, etc.

Few Examples of Environmental Controls and their Audit Procedures.

Control Activities	Control Techniques	Audit Procedures
Safeguards against the risks of heating, ventilation and air-conditioning systems.	Identify systems that provide constant temperature and humidity levels within the organization.	Review a heating, ventilation and air-conditioning design to verify proper functioning within an

		organization.
Control of radio emissions affect on computer systems.	Evaluate electronic shielding to control radio emissions that affect the computer systems.	Review any shielding strategies against interference or unauthorized access through emissions.
Establish adequate interior security based on risk	Critical systems have emergency power supplies for alarm systems; monitoring devices, exit lighting, communication systems.	Verify critical systems (alarm systems, monitoring devices, entry control systems) have emergency power supplies. Identify back -up systems and procedures and determine the frequency of testing. Review testing results.
Adequately protect against emerging threats, based on risk.	Appropriate plans and controls such as shelter in place or for a potential CBR attack(chemical, biological and radioactive attack) Restricting public access and protect critical entry points-air intake vents, protective grills and roofs.	Interview officials, review planning documents and related test results. Observe and document the controls in place to mitigate emerging threats. Observe location of these devices and identify security measures implemented. Verify the controls existence and intrusion detection sensors.
Adequate environmental controls have been implemented	Fire detection and suppression devices are installed and working.(smoke detectors, fire extinguishers and sprinkle systems) Controls are implemented to mitigate disasters, such as floods, earthquakes.	Interview managers and scrutinize that operations staff are aware of the locations of fire alarms, extinguishers, shut-off power switches, air - ventilation apparatus and other emergency devices.

	Redundancy exists in critical systems like, uninterrupted power supply, air cooling system, and backup generators Humidity, temperature, and voltage control are maintained and acceptable levels Emergency lighting, power outages and evacuation routes are appropriately located.	Determine that humidity, temperature and voltage are controlled within the accepted levels. Check cabling, plumbing, room ceiling smoke detectors, water detectors on the floor are installed and in proper working order.
Staff have been trained to react to emergencies	Operational and support personnel are trained and understand emergency procedures. Emergency procedures are documented and periodically tested- incident plan, inspection plan and maintenance plan.	Interview security personnel to ensure their awareness and responsibilities. Review training records and documentation. Determine the scope and adequacy of training. Review test policies, documentation and know-how of operational staff. Review incident handling procedures and maintenance and inspection plan.

Summary

The chapter deals with the physical and environmental threats and their control and audit procedures on information system assets. The first step in providing a secured physical environment for the information system assets is listing the various assets in the computing environment. These assets could range from hardware, software, facilities and people.. The next step is to identify the various threats and exposures the assets are exposed to. These could include unauthorized access to the resources, vandalism, public disclosure of confidential information and the like. The main source of threat is from outsiders and the employees of the organization. However, the information assets are exposed to various other threats like natural damage due to environmental factors like food, earthquake, fire, rain, etc.

Master Checklist for Physical and Environmental Security

To ensure IS assets are maintained in a secured manner within a controlled environment.

Sr. No.	Check points
Secured Physical Access	
1.	That physical Access Control Policy is documented and approved
2.	Whether the policy on the following is appropriate and covers: - Layout of facilities - Physical Security of the assets - Access to the assets - Maintenance of the assets - Signage on the facilities - Labels for assets - Visitors' authorization and recording - Entrance and exit procedures - Legal & regulatory requirements
3.	Whether critical IS facilities (like data center) are located appropriately (Verify the location for the following:- - Protection against natural disasters like earthquakes, flooding, extreme weather, etc. - Not in congested places - Not being on ground or top floor - Not being below ground level to avoid water leakage etc. - Not having a showcase window - Not having a direct access from the outside or through a public hallway - Place which is not visible externally).
4.	Whether the access to IS facilities is controlled through a secured mechanism (Verify the access control mechanism - e.g. access card, lock and key or manned reception).
5.	Whether the access to the IS facilities is limited to approved persons only (Approved persons may include employees, vendors and customers).

Sr. No.	Check points
6.	Whether the physical access control procedures are adequate and appropriate for approved persons (Access should be provided on need to do and need to know basis).
7.	Whether the visitor to critical IS facilities are escorted by employees (Records for visitors' access should be maintained).
8.	Whether a periodical review of access rights is carried out
9.	Whether the physical security is continually addressed
10.	Whether all access routes are identified and controls are in place
11.	Whether the security awareness is created not only in IS function but also across the organization
12.	Whether the physical security is ensured at suppliers' facilities also in cases where organization's' assets (either physical or data) are processed at supplier's facilities
13.	Whether the usage of any equipment outside the business premises for information processing is authorized by the management
14.	Is the security provided to equipment used outside business premises similar to / same as that offered to equipment used inside the business premises?
15.	Whether adequate monitoring equipment is available to monitor the movement of the personnel inside the facility
16.	In case of outsourced software, whether all maintenance work is carried out only in the presence of/ with the knowledge of appropriate IS staff
17.	Whether appropriate access controls like password, swipe card, bio-metric devices etc. are in place and adequate controls exist for storing data/ information on them Are there controls to ensure that the issue and re-collection of such access devices are authorized and recorded?
18.	Whether access violations are recorded, escalated to higher authorities and appropriate action taken
19.	Whether employees are required to keep the critical / sensitive documents in secured places
20.	Check if facility IS related risks with respect to lighting, building orientation, signage and neighborhood characteristics are identified?

Sr. No.	Check points
21.	Do the network, operating system and application monitoring procedures provide ample information to identify associated risks?
22.	Verify that surveillance systems are designed and operating properly.
23.	Ensure that physical access control procedures are comprehensive and being followed by the security staff.
24.	Verify if the security controls in place are appropriate to prevent intrusion into sensitive IS facilities: data centre, communication hubs, emergency power services facilities.
25.	Review facility monitoring measures to ensure that alarm conditions are addressed promptly.
Environmental Controls: To check	
1.	Whether the Environmental Control policy is documented and approved
2.	Whether IS facilities are situated in a place that is fire resistant (Verify for wall, floor, false ceiling, furniture and cabling being noncombustible / fire resistant / fire retardant).
3.	Whether smoking restrictions in IS facilities are in place
4.	Whether adequate smoke / temperature detectors are installed, connected to the fire alarm system and tested
5.	Whether fire instructions are clearly posted and fire alarm buttons clearly visible
6.	Whether emergency power-off procedures are laid down and evacuation plan with clear responsibilities in place
7.	Whether fire prevention and control measures implemented are adequate and tested periodically
8.	Whether fire drill and training are conducted periodically
9.	Whether air-conditioning, ventilation and humidity control procedures are in place, tested periodically and monitored on an ongoing basis
10.	Whether an adequate alternate power arrangement is available If so, is it covered under maintenance?
11.	Whether alternative water, fuel, air-conditioning and humidity control resources are available

Sr. No.	Check points
12.	Check if heating, ventilation, and air-conditioning systems maintain constant temperatures within a data center and other IS facilities.
13.	Evaluate the data center's use of electronic shielding to verify that radio emissions do not affect computer systems or that system emissions cannot be used to gain unauthorized access to sensitive information.
14.	Verify if there are sufficient battery backup systems providing continuous power during momentary black-outs and brown-outs along with generators that protect against prolonged power loss and are in good working.
15.	Ensure that a fire alarm is protecting a critical IS facility like data center from the risk of fire, a water system is configured to detect water in high-risk areas of the data center and a humidity alarm is configured to notify data center personnel of either high or low-humidity conditions.
16.	Check logs and reports on the alarm monitoring console(s) and alarm systems which are to be monitored continually by data center/IS facility personnel.
17.	Verify that fire extinguishers are placed every 50ft within data center isles and are maintained properly with fire suppression systems are protecting the data center from fire.
18.	Whether there are emergency plans that address various disaster scenarios, such as backup data promptly from off-site storage facilities.
19.	Ensure that there exists a comprehensive disaster recovery plan that key employees are aware of their roles in the event of a disaster and are updated and tested regularly.
20.	Ensure that detail part inventories and vendor agreements are accurate and current and maintained as critical assets.

Questions

1. Which of the following is not a type of Internal Control?
   a. Preventive
   b. Additive
   c. Detective
   d. Corrective

2. Which of the following is not a Water-based system?
   a. Post implementation
   b. Pre-action
   c. Dry Pipe system
   d. Wet Pipe system

3. Which of the activities is not included in auditing Physical Access Control?
   a. Risk assessment
   b. Controls Assessment
   c. Review of Documentation
   d. Corrective assessment

4. Which of the following is not an implementation method of Physical Access Control?
   a. Administrative
   b. Technical
   c. Logical
   d. Physical

5. Which of the following is not a principle of Information Safety?
   a. Redundancy
   b. Confidentiality
   c. Integrity
   d. Availability

6. MTBF stands for _____.
   a. Mean Time Between Falls.
   b. Median Time Between Failures.
   c. Mean Time Between Failures.
   d. Median Time Between Falls.

7. MTTR stands for _____.
   a. Mean Time To Ready/recover/respond/ restore.
   b. Mean Time To Repair/recover/respond/ restore.
   c. Mean Time To Repair/recover/respond/ relieve.
   d. Mean Time To Repair/restart/respond/ restore.

8. Which of the following is not a category of physical access threat?
   a. Electrical
   b. Hardware
   c. Environmental
   d. Mathematical

9. UPS stands for _____
   a. Uninterrupted Power Supply.
   b. Uninterrupted Power Supplier.
   c. Uniform Power Supply.
   d. None of these.

10. Deadman doors are also called _____.
   a. Biometric door locks.
   b. Mantrap systems.
   c. Bolting door locks.
   d. None of these.

11. _____controls encompass securing physical access to computing equipment as well as t to facilities housing the IS computing equipment and supplies.
   a. Environmental access
   b. Logical access
   c. Physical access
   d. Computer system

12. IPF stands for _____.
   a. Information Product Facility.
   b. Information Processing Feature.
   c. Input Processing Facility.
   d. Information Processing Facility.

13. War and bomb threats are an example of _____.
   a. Environmental Threat.
   b. Man Made Threat.
   c. Physical Threat.
   d. Metaphysical Threat.

14. Humidity, vapors, smoke and suspended particles are an example of _____.
   a. Natural Threat.
   b. Man Made Threat.
   c. Physical Threat.
   d. None of these.

15. Data_____ prevents modification of data by unauthorized personnel.
   a. Integrity
   b. Confidentiality
   c. Availability
   d. Marketability.

16. Preventing disclosure of information to unauthorized individuals or systems is defined as _____.
    a. Integrity.
    b. Confidentiality.
    c. Availability.
    d. Utility.

17. Physical Infrastructure is a/an _____ Asset Class.
    a. Intangible
    b. Tangible and Intangible
    c. Tangible
    d. Development

18. HVAC stands for _____
    a. Heating, Ventilation and Air Conditioning.
    b. Heating, Vending and Air Conditioning.
    c. Hiring, Vending and Air Conditioning.
    d. Heating, Vending and Authoring.

19. ____ is defined as policies, procedures, practices and enterprise structure that are designed to provide reasonable assurance that business objectives will be achieved and undesirable events are either prevented or detected and corrected.
    a. Audit
    b. Access
    c. Prevention
    d. Control

20. CCTV stands for _____.
    a. Closed-Circuit Television.
    b. Clear Circuit Television.
    c. Closed-Circuit Tariff.
    d. None of these.

Answers:

1 b	2 a	3 d	4 c	5 a	6 c
7 b	8 d	9 a	10 b	11 c	12 d
13 b	14 a	15 a	16 b	17 c	18 a
19 d	20 a				

2 Logical Access Controls

Learning Objectives

- The Objectives of Logical Access
- The Paths of Logical Access
- Logical Access Exposures
- Authentication Techniques
- Operating System Security
- Database Security

Introduction

Today IT systems store and process a wide variety of data centrally and provide access to a large number of users. Storing data centrally on a system is cost effective and contributes to efficient information sharing and processing. In such an environment it is not unusual to expect that:

- some information is accessible to all users,
- some is for several groups or departments, and
- some for only a few individuals.

Information on a system that is accessed by many users has the associated risk of unauthorized access. Therefore, a significant concern is to ensure that users have access to information they need but do not have inappropriate access to data that may be sensitive and not required by them. It is also important to ensure that certain items, though readable by many users, are changed only by a few.

Logical access controls are a means of addressing these concerns. These are protection mechanisms that limit users' access to data to what is appropriate for them. Such controls are often built into the operating system, or form part of the "logic" of applications programs or major utilities, such as Database Management Systems. They may also be implemented in add-on security packages that are installed into the operating system.

In this chapter, we look at the ways in which t data is accessed and how logical access controls help to ensure that only the right persons access the right data.

Objectives of Logical Access Controls

Information is the primary commodity in the world of E-Commerce. As technology advances and access to markets expands, the need to protect information to ensure confidentiality, integrity, and its availability to those who need it for making critical personal, business, or government decisions becomes very important.

Logical access controls are the means of information security. Their purpose is to restrict access to information assets / resources. They are expected to provide access to information resources on a need to know and need to have basis using the principle of least privileges. It means that access should not be so restrictive that it makes the performance of business functions difficult but, at the same time, it should not be so liberal that it is misused. The data, an information asset, can be

- Resident on a machine (for use by an application)
- Stored in some medium (Back up)
- Or it may be in transit. (being transferred from one location to another)

Logical access controls is all about protection of these assets wherever they reside.

Paths of Logical Access

Access to an organization's information systems is possible through various means / routes. For example, access to an information resource on a network is possible through one of following:

- A machine connected to the network
  - A terminal
  - A client machine
  - An administrator console
- A network device that is part of the network and with a free port to which a personal computer can be attached
  - Hub
  - Switch
  - Bridge
  - L3 Switch
  - Router
- Dialup device connected to network
  - A computer with a modem (This is useful only if the network to which logical access is required also responds to modem calls)
- A machine having access to the network through wireless mode

Each of these routes has to be subjected to appropriate means of security in order to secure it from possible logical access exposures.

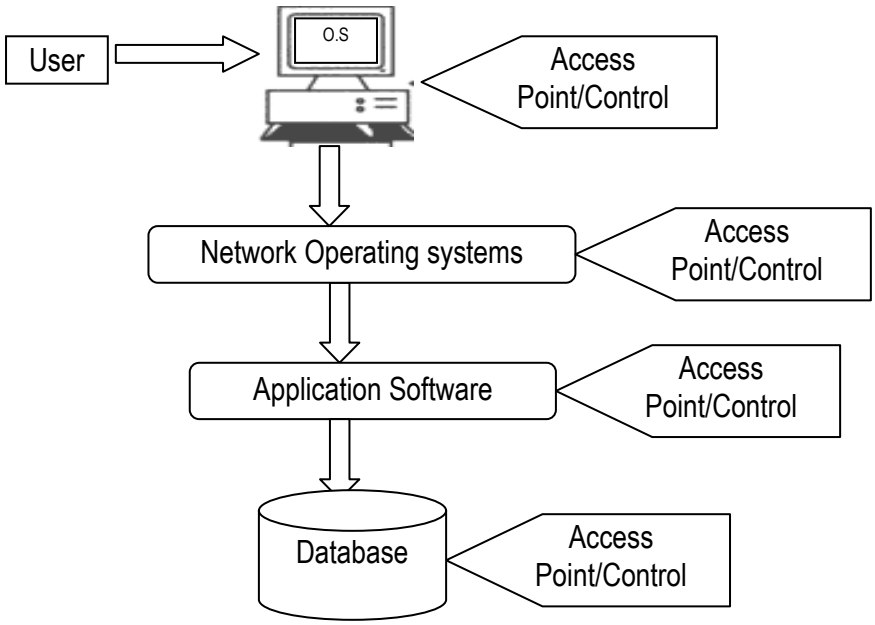


Fig 2.1: Logical Access Paths in an Enterprise Information System.

Logical Access Exposures

Improper logical access can result in loss or damage to information and resources leading to undesirable consequences for an organization. It can also result in a violation of the confidentiality or integrity or availability of information. Some of the technical exposures relating to logical access are discussed as follows:

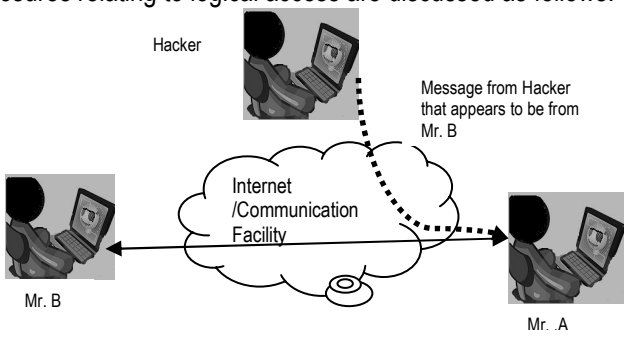


Fig.2.2: Masquerade

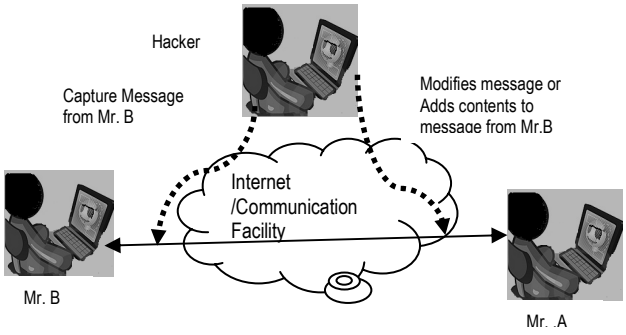


Fig.2.3: Piggybacking

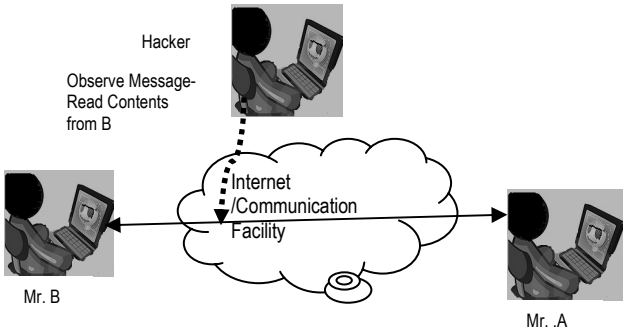


Fig.2.4: Wire Tapping

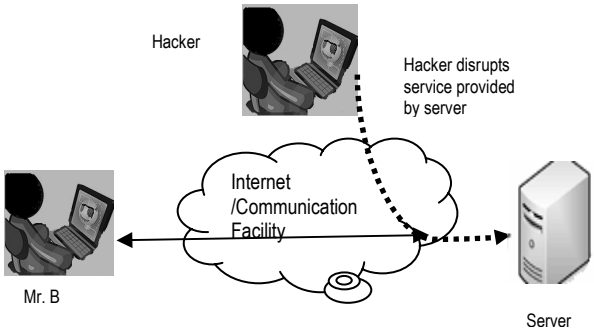


Fig.2.5: Denial of Service

Data Leakage:	Information in electronic form is highly susceptible to leakage and stealing. Sensitive information contained in computer files may be copied on to disks or other media, electronically transferred across networks to other locations, transmitted across the Internet or simply viewed or printed. Accidentally, users may be able to lay hands on sensitive information contained in the systems, due to improper logical access configurations.
Wire Tapping:	This is similar to tapping of telephone lines. Networking cables or telecommunication lines may be tapped by any being for eavesdropping. This can also be done by the use of special probes without any damage to the communication media.
Scavenging or Dumpster Diving:	Discarded listings, tapes, or other information storage media from trash are filtered to determine useful information, such as access codes, passwords, or sensitive data. The items may have been discarded by their owners, but become useful to the Dumpster diver.This can be used by the perpetrator to put together critical information about an organization or subject it to attack.
Emanation Interception	Many of the devices used for computing radiate signals or waves, such as electromagnetic radiation emanated by dot matrix printers and monitors, which can be intercepted by using simple devices such as magnetic field effect detector, spectrum analyzer and recording with transmitters.
Data Diddling	This is possibly the least technical of the various logical issues and exposures. This involves unauthorized modification of input being submitted for processing. This could simply involve altering the information on the input documents manually, such as vouchers or invoices waiting to be entered or as they are entered. This also includes modification of information in electronic form such as input files or transaction files, waiting processing.
Piggybacking:	Unauthorized access to information by using a terminal that is already logged on with an authorized ID (identification) and left unattended.
Masquerading:	It means disguising or Impersonating. The attacker pretends to be an authorized user of a system either r to gain access or

	get greater privileges than he is authorized for. Masquerading may be attempted by using a stolen logon IDs and passwords, through security gaps in programs, or through bypassing the authentication mechanism. The attempt may come from within an organization, say from an employee; or from an outsider through some connection to the public network. Weak authentication provides one of the easiest points of entry for a masquerader. Once attackers have been authorized for entry, they may get full access to the organization's critical data, and (depending on the privilege level they pretend to have) may be able to modify and delete software and data or make changes.
Spoofing:	Is means to deceive or to play hoax on a network by one of the following: *IP spoofing:* To deceive for the purpose of gaining access to someone else's resources (for example, to fake an Internet address so that one looks like a certain kind of trusted Internet user). *E-mail spoofing:* To change an e-mail header so that the message appears to have originated from someone or somewhere other than the actual source. *Web spoofing:* To playfully satirize a Web site. In all the cases of spoofing something is being forged. It may, for example, be an email ID or an IP address, etc.
Asynchronous Attacks:	Such attacks are specific to devices, which exchange data asynchronously. In asynchronous transmission, the sender and receiver do not operate at the same speed, hence the data is often stored in temporary buffers awaiting transmission or processing, and susceptible to attacks. These attacks are quite complex and often difficult to detect.
Keystroke monitoring (also called Key Logging):	is a process whereby a system administrator views or records both the keystrokes entered by a computer user and the computer's response during a user-to- computer session. Keystroke monitoring means viewing or capturing characters as they are typed by users, reading users' electronic mail, and viewing other recorded information typed by users. It is also used as a diagnostic in software development to determine sources of error in computer systems. Such systems are also highly useful for law enforcement, eavesdropping, and

	espionage. For instance, providing a means to obtain passwords or encryption keys and thus bypassing other security measures. Keystroke monitoring can be achieved by using both hardware and software. Commercially available systems include devices which are attached to the keyboard cable (and thus visible if the user makes a thorough inspection) and also devices which can be installed (thus invisible to users), for example, key logger programs.
Buffer Overflow	A program or process may receive more data than it can handle. If the developer has not written instructions on how the process should behave when handling excessive data, the program may malfunction. This malfunctioning can be exploited by a malicious attacker.
Rounding Down	The perpetrator rounds down the amounts in various transactions to the nearest desired decimal place. This involves small modifications to the transaction values whereby the rounded difference is routed to a pre-designated account. A perpetrator can induce the bank transaction processing system to alter paisa value of clearing credits to various customer accounts from say Rs.5,45,859.37 rounded down to Rs.5,45,859.35 and transfer Rs.0.02 to an unauthorized account.
Salami Technique:	This is similar to the Rounding Down technique except that in this case the perpetrator slices off a fixed value from each amount. For example, the perpetrator may modify a program that calculates interest to account holders in such a way that a small fixed amount, say Rs. 0.03, is deducted from all payments and transferred to an unauthorized account.
Trap Doors	These are application system vulnerabilities which allow the perpetrator to bypass security protections and allow him to insert unauthorized logic or modify the behaviour of application systems in undesirable ways. Usually, a trapdoor is created by the original programmer to allow him unauthorized access to the system.
Remote Shut Down:	The perpetrator uses various techniques to assess the vulnerabilities and loopholes of a system and attempts to gain access to it. He/she then remotely and maliciously shuts down the system causing a loss of system availability and perhaps

	loss of integrity as the data may be corrupted during the unplanned shut down.
Denial of Service (DoS)	The perpetrator attempts to flood memory buffers and communication ports to prevent delivery of normal services. This and the more aggressive Distributed Denial of Service attack are dealt with in a subsequent chapter.
Social Engineering	The perpetrator uses social and tactical skills to obtain information such as passwords, PIN etc. exploiting the human weaknesses of greed, appeasement and fear.
Phishing Attacks	The most popular attacks on banking systems in recent times have been attackers who target gullible victims, using a combination of social engineering, e-mail and fake websites to con the victim to click on a link embedded in an apparent authentic mail from a reputed bank. The link takes the victim (generally a customer of the bank) to a lookalike Bank website that gets the personal details of the victim including details such as PIN and internet banking password, which is then exploited by the hacker.

Did you know?

As per one of the prominent surveys, US consumers lost over $3.2 billion during 2007 due to phishing frauds alone.

Malicious Code

Malicious code (also called "Malware") is the name used for any program that adds to, deletes or modifies legitimate software for the purpose of intentionally causing disruption and harm or to circumvent or subvert the existing system. . Examples of malicious code include viruses, worms, Trojan Horses, and logic bombs. Some malicious codes are based on Active X and Java applets.

Viruses:	Are malicious codes that are attached to a host program and propagate when an infected program is executed? The perpetrator's objective is to multiply and spread the code. However, they are dependent on another program or human action to replicate or to activate their payload.
Worms:	Are malicious programs that attack a network by moving from device to device and create undesirable traffic. They differ from viruses in

	that they replicate or get activated on their own, and do not depend on any program or human action, and spread rapidly than viruses.
Trojan Horses:	These are malicious codes which hide inside a host program that does something useful. Once these programs are executed, the hidden malicious code is released to attack the workstation, server, or network or to allow unauthorized access to those devices. Some Trojans are programmed to open specific ports to allow access for exploitation. Then the open Trojan port could be scanned and located, enabling an attacker to compromise the system. These are also used as tools to create backdoors into the network for later exploitation by crackers.
Logic Bombs:	These are legitimate programs, to which malicious code has been added. Their destructive action is programmed to "blow up" on occurrence of a logical event such as time or a logical event as number of users, memory/disk space usage, etc. Every time the infected program is run, the logic bomb checks external environment to see whether the condition to trigger the bomb has been met. If not, the control is passed back to the main application and the logic bomb waits. If the condition is satisfactory, the rest of the logic bomb's code is executed, and it attacks the system. Logic bombs are very difficult to detect since they reside in the system, and their destructive instruction set is known only after it blows up.
Macro Viruses:	A macro is an instruction that carries out program commands automatically. Many common applications (e.g. word processing, spreadsheet, and slide presentation applications) make use of macros. A macro virus is a computer virus that "infects" a Microsoft Word or similar application and causes a sequence of actions to be performed automatically when the application is started or as an event trigger. If a user accesses a document containing a viral macro and unwittingly executes this macro virus, it can then copy itself into that application's startup files. The computer is now infected and a copy of the macro virus resides on the machine. Any document on the machine that uses the same application can become infected. If the infected computer is on a network, the infection is likely to spread rapidly to other machines on the network. Because these types of files are commonly used and sent through e-mail, a computer network can be quickly infected by these viruses. Macro viruses tend to be surprising, but relatively harmless.

Polymorphic Viruses	Polymorphic viruses are difficult to detect because they hide themselves from antivirus software by altering their appearance after each infection. Some polymorphic viruses can assume over two billion different identities.
Stealth Viruses	Stealth viruses attempt to hide their presence from both the operating system and the antivirus software by encrypting themselves. They are similar to polymorphic viruses and are very hard to detect.
Adware and Spyware	They are software that tracks the Internet activities of the user usually for the purpose of sending targeted advertisements. Besides the loss of privacy and waste of bandwidth (loss of availability), they do not pose other security related risks.. However, it is quite likely that Trojans could be embedded in such software. Adware and Spyware often come with some commercial software, both packaged as well as shareware software. There is often a reference to the Adware and Spyware software in the license agreement.

Logical Access Controls

Security policy is a framework within which an organization establishes needed levels of information security to achieve the desired confidentiality goals. It is a statement of information values, protection responsibilities, and organization's commitment to its system. It is a set of rules, principles, and practices that determine how security is implemented in an organization. Access control policy is a part of security policy and must address the following:

User access management

i. **User registration**
Information about every user is documented. The following questions are to be answered:

- Why is the user granted access?
- Has the data owner approved it?
- Has the user accepted responsibility?
- Is the de-registration process is also equally important?

ii. **Privilege management**
Access privileges are to be aligned with job requirements and responsibilities. For example, an operator at the order counter shall have direct access to order processing activity of the application system. He/she will be provided higher

access privileges than others. However, misuse of such privileges could endanger the organization's information security.

iii. **User password management**
Passwords are usually the default screening point for access to systems. Allocations, storage, revocation, and reissue of passwords are password management functions. Educating users and make them responsible for their password is a critical component of password management.

iv. **Review of user access rights**
A user's need for accessing information changes with time and requires a periodic review of access rights to check anomalies in the user's current job profile, and the privileges granted earlier.

User responsibilities

User awareness and his responsibility is also an important factor:

i. **Password use**
Mandatory use of strong passwords to maintain confidentiality.

ii. **Unattended user equipment**
Users should ensure that no equipment under their care is ever left unprotected. They should also secure their PCs with a password, and prevent its access by others.

Network access control

An Internet connection exposes an organization to the entire world. This brings up the issue of benefits the organization derives along with the precaution against harmful elements. This can be achieved through the following means:

i. **Policy on use of network services**
An enterprise wide applicable internet service requirements aligned with the business needs to meet their service scope is the first step. A clear service definition for the appropriate services and approval to access required network services will be a part of this policy.

ii. **Enforced path**
Based on risk assessment, it is necessary to specify the exact path or route connecting the networks; say, for example, internet access by employees will be routed through a firewall. It is also necessary to maintain a hierarchical access levels for both internal and external user logging.

iii. **Segregation of networks**
Based on the sensitive information handling function, say a Virtual Private Network (VPN) connection between a branch office and the head office, this

network is to be isolated from the basic internet usage service availability for employees.

iv. **Network connection and routing control**
The traffic between networks should be restricted, based on identification of source and authentication access policies implemented across the enterprise network facility.

v. **Security of network services**
Are implemented with the help of techniques of authentication and authorization policy implemented across the organization's network.

Operating system access control

Operating system provides the platform for an application to use various IS resources and perform a specific business function. If an intruder is able to bypass the network perimeter security controls, the operating system is the last barrier to be conquered for unlimited access to all the resources. Hence, protecting the operating system access is extremely crucial.

i. **Automated terminal identification**
This will ensure that a particular session can only be initiated from a particular location or computer terminal.

ii. **Terminal log-on procedures**
A basic log-on procedure that with hardly any help or hint, it prevents misuse by an intruder.

iii. **User identification and authentication**
The users must be identified and authenticated in a foolproof manner. Depending on risk assessment, more stringent methods like Biometric Authentication or Cryptographic means like Digital Certificates should be employed.

iv. **Password management system**
An operating system could enforce the use of strong passwords. Internal storage of password should use one-way encryption algorithms and the password file should not be accessible to users.

v. **Use of system utilities**
System utilities are the programs that help to manage critical functions of the operating system. For example, addition or deletion of users. Use and access to these utilities should be strictly controlled and logged.

vi. **Duress alarm to safeguard users**
If users are forced to execute some instruction under threat, the system should

provide a means to alert the authorities. An example could be forcing a person to withdraw money from the ATM. Many banks provide a secret code to alert the bank about such transactions.

vii. **Terminal time out**
Log out the user if the terminal is inactive for a defined period. This will prevent misuse in the absence of a legitimate user.

viii. **Limitation of connection time**
Define the available time slot, and do not allow any transaction beyond this time period. For example, no computer access after 8.00 p.m. and before 8.00 a.m.— or on a Saturday or Sunday.

Application and monitoring system access control

i. **Information access restriction**
Access to information is prevented by the application of specific menu interfaces, which limit access to system function. A user is allowed to access only to those items that he is authorized to access. Controls are implemented on the access rights of users, For example, read, write, delete, and execute. Ensure that sensitive output is sent only to authorized terminals and locations.

ii. **Sensitive system isolation**
Based on the critical constitution of a system in an enterprise, it may even be necessary to run the system in an isolated environment. Monitoring system access and use is a detective control, to check if preventive controls (discussed so far) are working. If they are not, this control will detect and report any unauthorized activities.

iii. **Event logging**
In Computer systems it is easy and viable to maintain extensive logs for all types of events. These should be logged and archived properly.

iv. **Monitor system use**
Based on the risk assessment, constant monitoring of some critical systems is essential. Define the details of types of accesses, operations, events and alerts that are to be monitored. The extent of detail and the frequency of review would be based on the criticality of operation and risk factors. The log files are to be reviewed periodically and attention should be given to any gaps in these logs.

v. **Clock synchronization**
Event logs maintained across an enterprise network play a significant role in correlating an event and generating a report on it. Hence the need for

synchronizing clock time across the enterprise/organization network is mandatory.

Mobile computing

In today's organizations computing facility is not restricted to a particular data centre alone. Ease of access on the move provides efficiency and results in additional responsibility on users and the need to maintain information security on the management. Theft of data carried on the disk drives of portable computers is a high risk factor. Both physical and logical access to these systems is critical. Information is to be encrypted and access identifications like fingerprint, eye-iris, and smart cards are necessary security features.

Identification and Authentication

The primary function of access control is to allow authorized access and prevent unauthorized access. Access control mechanism is actually a three step process, as depicted in the figure below:

a. Identification: Identification is a process by which a user provides a claimed identity to the system such as an account number.

b. Authentication: Authentication is a mechanism through which the user's claim is verified.

c. Authorization: The authenticated user is allowed to perform a pre-determined set of actions on eligible resources.

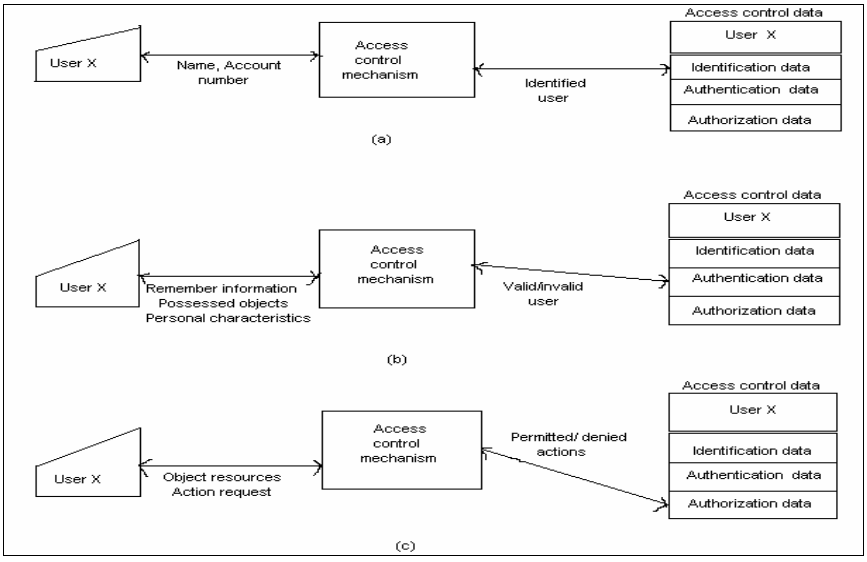


Fig.2.6: Identification, Authentication and Authorization

The primary function of access control is to allow authorized access and prevent unauthorized access to information resources in an organization. Therefore, it may become necessary to apply access control at each security architectural layer of an organization's information system architecture to control and monitor access in and around the controlled area. This includes operating system, network, database and application systems. In each of these layers attributes may include some form of identification, authentication and authorization and logging and reporting of user activities.

Interfaces exist between operating system access control software and other system software access control programs such as those of routers, and firewalls that manage and control access from outside or within an organization's networks. On the other side operating system access control software may interface with databases and / or application system access controls to application data.

Identification Techniques

Authentication is the process of verifying that the identity claimed by the user is valid. Users are authenticated by using any one of the three personal authentication techniques. These are:

- something the user knows (e.g., a password.,
- something the user has (e.g., a token or smart card., and
- something the user is (a physical / biometric comparison).

Single-factor authentication uses any one of these. Two-factor or dual factor authentication uses two and the three-factor authentication uses all the three. . Individual authentication assurance increases when multiple authentication technologies and techniques are used in a combined way. .

Once the user is authenticated, the system must be configured to validate that the user is authorized (has a valid need-to-know) for the resource being protected and can be held accountable for any actions taken. Authorized access to logical assets can be implemented as a combination of manual, automated, and/or administrative methods. A deny-by-default policy, where access to the information resource is denied unless explicitly permitted, should be mandated. The decision to grant or deny access to an information resource is the responsibility of the resource owner.

Authentication Techniques

As stated above, authentication may be through remembered information, possessed tokens, or physical characteristics. We shall examine each class of authentication techniques below.

Fig.2.7 : What you have (Token), what you know (password/PIN) and who you are (Biometric.

i. Passwords :This is the most common authentication technique that depends on remembered information. The user identifies himself by using his login-id to the system and then provides the password information. Once the system is able to locate the match, it authenticates the user and enables access to resources based on the authorization matrix. However, if a match is not successful, the system returns a message (such as "Invalid User or password") thus preventing access to resources.

ii. Personal Identification Numbers (PINs): It is a type of password, usually a 4-digit numeric value, that is used in certain systems to gain access.. The PIN should be such that a person or a computer cannot guess it in sufficient time by using a guess and check method, i.e. where it guesses the PIN, and checks for correctness by testing it on the system that the person is attempting to gain access to and the process is repeated with a different guess till access is obtained. PINs are commonly used for gaining access to Automatic Teller Machines (ATMs).

iii. Weaknesses of Logon mechanism Logon/password access security is based on information to be remembered by the user (what the user knows). This results in the following weaknesses:

Passwords are easily shared.

- Users often advertently or inadvertently reveal passwords, and thus compromise security. Repeated use of the same password could help people to guess it easily.

- If a password is too short or too easy, the chances of it being guessed are quite high.
- If a password is too long or too complex, the user may forget or may write it down.
- If many applications are to be accessed by one user, many passwords have to be remembered.
- Passwords can be shared, guessed, spoofed or captured.

i. Recommended practices for strong passwords

- The user should not share his password.
- The password should be easy for the user to remember but hard for the perpetrator to guess.
- On the creation of a new user, the first password is allotted by the security administrator and a change of password is forced on the first login.
- Users should be encouraged or forced to change passwords periodically. For example, once in 60 days.
- Concurrent logins by the same user should not be permitted.
- Passwords should not be too short and should not be the name user pet-name of a user, or common words found in a dictionary.
- Password combination should be random and use alphabetic, numeric and special characters (such as "!", "#" , "^", etc.).
- The number of wrong login tries should be restricted to three, after which the system should lockout the user. Further access can be granted only through the intervention of the security administrator.
- The logon ids active in the system should not exceed the number of users actually authorized to access systems resources.
- Passwords should be stored in an encrypted form using one-way encryption.
- In case the user remains inactive at a terminal, for a length of time (say 20 minutes), the terminal should lock out the user and require the user to login again.

ii. Attacks on logon/password systems

Due to their inherent weaknesses, logon-id/password access control technique is vulnerable to various kinds of malicious attacks. Some of the common attacks on such systems are:

- **Brute force:** In this crude form of attack, the attacker tries out every possible combination to hit on the successful match. The attacker may also use various password cracking softwares that assist in this effort.

- **Dictionary attack:** On similar lines, this type of attack is based on the assumption that users tend to use common words as passwords, which can be found in a dictionary, hence the name. The "dictionary" simply consists of a list of words, including proper names (Raju, Ramesh, Ibrahim, etc. and also that of mythological or religious names (Krishna, Jesus, Osiris, Buddha, etc.).
- **Trojan**: AA malicious software, which the attacker can use to steal access control lists, passwords or other information.
- **Spoofing attacks:** In this technique, the attacker plants a Trojan program, which masquerades as the system's logon screen, gets the logon and password information and returns control to the genuine access control mechanism. Once the information is obtained, the attacker uses the information to gain access to the system resources.
- **Piggybacking:** As stated earlier, an unauthorized user may wait for an authorized user to log in and leave a terminal unattended. The logical techniques that are used to secure against this attack are to automatically log out the session after a pre-determined period of inactivity or by using password-protected screen savers.

Token Based Authentication

Objects that a user is required to possess for identification and authentication are known as tokens.

Fig.2.8: Smart Tokens

Fig.2.9: Memory Tokens

i. **Plastic Cards:** Plastic cards contain information about the user and primarily provide a means of identification of the user and enable authentication. Plastic cards are of the following types:

- **Memory tokens:** In its most common form, the cards contain visible information such as name, identification number, photograph and such other information about the user and also a magnetic strip. This strip stores static information about the user. In order to gain access to a system, the user is

required to swipe his card through a card reader, which reads the information on the magnetic strip and passes onto the computer for verification. Where two-factor authentication is adopted, the user is not only required to have his card read by a card reading device but also required to key in remembered information (passwords, PIN)

- **Smart Tokens: In** this case, the card or device contains a small processor chip which enables storing dynamic information on the card. Besides static information about the user, the smart tokens can store dynamic information such as bank balance, credit limits etc. However, the loss of smart cards can have serious implications.

ii. **Proximity Readers:** In this case when a person in possession of the card reaches the restricted access area, the card data is read by the proximity readers (or sensors) and transmits information to the authentication computer, which enables access to the restricted area or system. Because the user is not required to insert the card into the device, access is faster. Proximity tokens can be either static or processor based. In static tokens, the card contains a bar code, which has to be brought in proximity of the reader device. In case of processor based tokens, the token device, once in the range of the reader, senses the reader and transmits the password to authenticate the user. Other token based systems include challenge response systems and one time passwords.

iii. **Single Sign-on:** In many situations, a user, because of his job responsibilities in an organization is often required to log into more than one application. The user has to remember multiple logons and passwords. This can be solved by a single sign-on, which provides user access to various applications. It is a session/user authentication process that permits a user to enter one name and password in order to access multiple applications. The single sign-on, which is requested at the beginning of the session, authenticates the user to access all the applications they have been given the rights to on the server, and eliminates future authentication prompts when the user switches applications during that particular session. The concern in a decentralized processing or database environment is that the passwords travel over communication lines. Also if the single username and password used for single sign on is compromised, unauthorized access to all related application is possible.

Biometric Security

Compared to log on and token based authentication, Biometrics offers a very high level of authentication based on "what the user is". Biometrics, as the name suggests, is based on certain physical characteristics or behavioral patterns identified with the individual, which are measurable. The International Biometric Group defines

biometrics as automated mechanism which uses physiological and behavioral characteristics to determine or verify identities. Behavioral biometrics is based on measurements and data derived from an action and indirectly measure characteristics of the human body. Based on some feature unique to every user, biometrics seeks to minimize the weaknesses in other mechanisms of authentication. Some biometric characteristics are:

- Fingerprints
- Facial Scans
- Hand Geometry
- Signatures
- Voice
- Keystroke Dynamics
- Iris Scanners
- Retina Scanners

Implementation of biometric authentication is often expensive and involves the following phases:

- Identification of IS assets which require biometric security
- Based on the above, identification of appropriate biometric application
- Acquisition and testing of appropriate hardware, calibration of error rates for effectiveness and efficiency of enrolment and readability
- Implementation of administrative procedures for exception reporting and adjustment for false positives
- Enrolment of Users
- Implementation of related physical and logical controls

Identification and authentication is based on a match with items in the database containing data captured during user enrolment. Registration or enrolment of an individual's physical or behavioral characteristics involves capturing information and digitizing and storing of biometric data. Based on the data read by the sensor, the image or digitized data is compared to the stored data to obtain a match. If matching ends successfully, authentication is successful.

However, due to the complexity of data, biometrics can cause two kinds of errors: False Rejection Rate (FRR) which is wrongfully rejecting a rightful user and False Acceptance Rate (FAR) which involves an unauthorized user being wrongfully authenticated as a right user. Ideally a system should have a low false rejection and low false acceptance rate. Most biometric systems have sensitivity levels which can be tuned. With increase in the sensitivity of a system, FAR drops while FRR

increases. Thus, FRR and FAR tends to get inversely related. An overall metric used is the Crossover Error Rate (CER), which is the point at which FRR equals FAR.

Due to their high cost of implementation, biometric access controls were initially used only for high value critical information resources such as defence, banking etc. However, with rapid decrease in the cost of biometric hardware, biometric controls are being increasingly preferred for commercial applications. Finger print based biometric controls are quite popular and widely deployed in data centres.

Authorization Techniques –Operating Systems

Operating systems provide security to computing systems. The operating system manages resources and execution of applications with security constraints defined at operating systems level. The system must also protect itself because compromise would give access to all the user accounts and all the data in its files. A weak operating system would allow attackers access not only to data in it but also in database systems. The system isolates processes from each other, protects the permanent data stored in its files, and provides controlled access to shared resources. Most systems use the access matrix as a security model. An access matrix defines which processes have what types of access to specific resources.

General operating systems access control functions include:

- Authentication of the user
- User Management
- Restrict Logon IDs to specific workstations and / or specific times
- Manage account policies
  - Password Policy
  - Account Lockout Policy
- Manage audit policy
- Log events
- Report capabilities

Pluggable Authentication Modules

The pluggable authentication module (PAM) framework provides system administrators with the ability to incorporate multiple authentication mechanisms into an existing system through the use of pluggable modules.

Applications enabled to make use of PAM can be plugged-in to new technologies without modifying the existing applications. This flexibility allows administrators to do the following:

- Select any authentication service on the system for an application

- Use multiple authentication mechanisms for a given service
- Add new authentication service modules without modifying existing applications
- Use a previously entered password for authentication with multiple modules
- Use a general authentication scheme independent of the authentication mechanism.

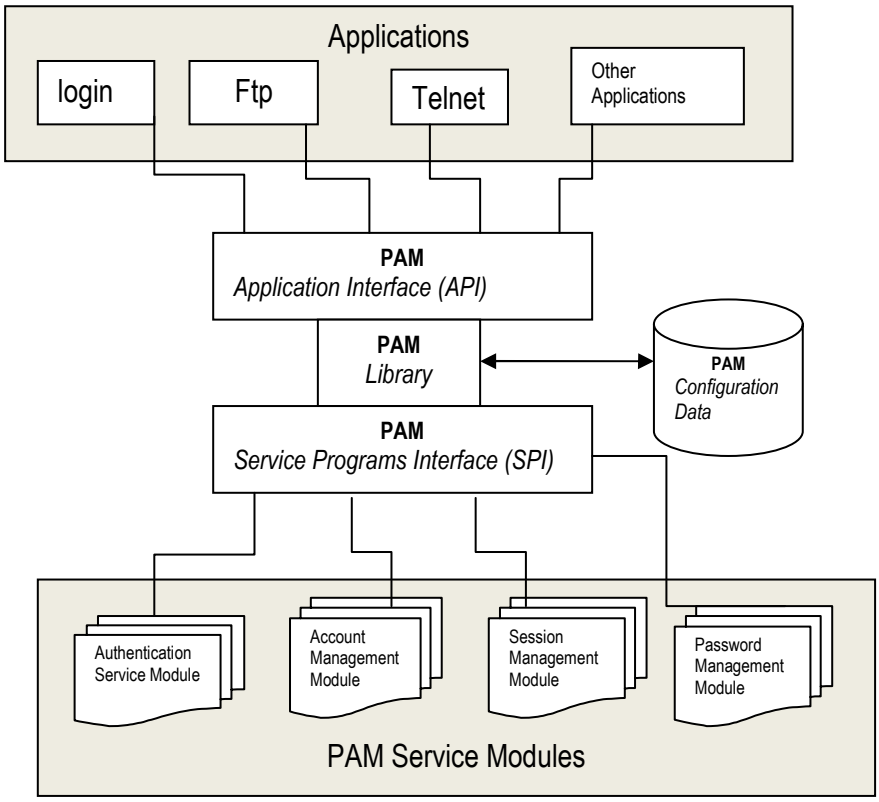


Fig.2.10: PAM Framework

In most operating systems, every file is owned by a user and can be accessed by its owner, group or public, depending upon access permissions. When a user creates a file or directory, that user becomes its default owner. . A user may be a member of one group or several groups. Further, a user owner of a file may not be a part of the group that may have access to the file. Again, most operating systems have at least three types of file permissions; read, write and execute. The users have to be given at least read access to many of the system files.

Access Control Lists (ACL)

An access control list is a table that tells the computer operating system which access rights each user has to a particular system object, such as a directory/folder or an individual file. Each object has a security attribute that identifies its access control list. The list has an entry for each system user with his access privileges. The most common privileges include the ability to read a file (or all the files in a directory), to write to the file or files, and to execute the file (if it is an executable file, or program). The access control list is implemented differently by each operating system and is the foundation of any security functionality. Access control enables one to protect a system or its part (directories, files, file types, etc.). When the system receives a request, it determines access by consulting a hierarchy of rules in the ACL. ACL has one or more access control entries (ACEs), each consisting of the name of a user or a group of users. The user can also be a role name, such as a programmer or tester. For each of these users, groups, or roles, the access privileges are stated in a string of bits called an access mask. Generally, the system administrator or the object owner creates the access control list for an object.

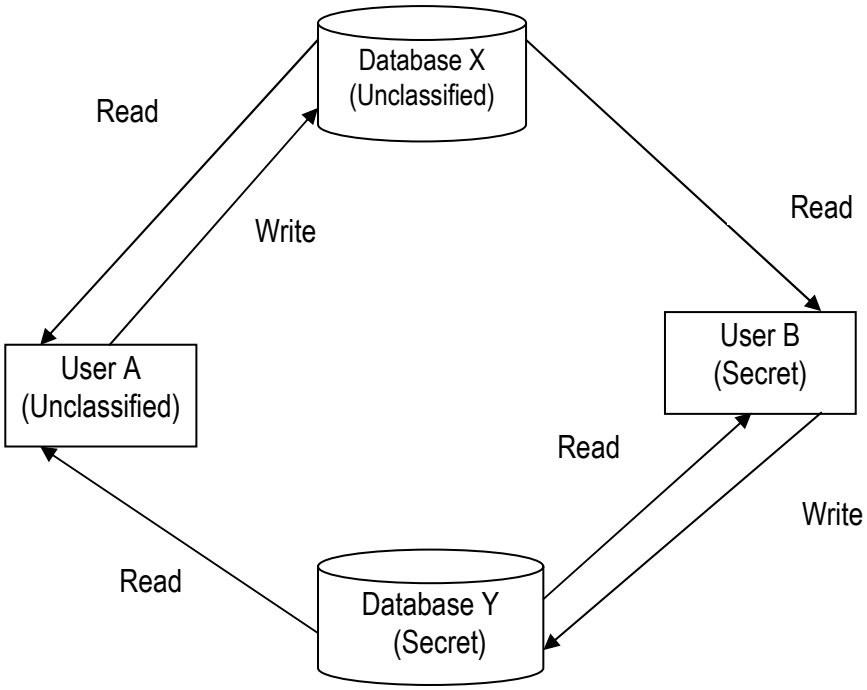


Fig.2.11: Read and Write Access Policy.

Discretionary access privileges to users pose problems within the organization, giving rise to the need for mandatory and well documented access control policies. These policies use attributes to determine which user can access which resource. For example, users A and B may read from the unclassified database, but the secret database can be read only by the secret user B. Suppose if both A and B can write to the unclassified and secret database then the unclassified user can read the secret information written by the secret user B. Here the user B is responsible for downgrading the information.

User	**Resource**	**DataBase X**	**Database Y**
User A		Read & Write	Read
User B		Read	Read & Write

Few examples of Logical Access Control Techniques and their **Suggeste**d Audit Procedures. **Control Activity**	**Control Techniques**	**Audit Procedures**
User accounts are appropriately controlled	Resource owners have a list of identified authorized users and the access they are authorized to have. Passwords, tokens, biometric, smartcards, etc are used to identify and authenticate users. Security administration parameters are set for access to data files, software code libraries, security files and important operating system files. Naming conventions are established for controlling	Segregation of duties- To ensure that users do not have access to incompatible functions. Review policies and procedures which spell out access authorization documentation and user rights and privileges in the information system. Determine directory names for sensitive directories, files and their access levels and types of access.

	access to data and programs. Redundant accounts like default, guest are removed, disabled or secured. Review access to - shared files -emergency or temporary access to files and hosts These are to be controlled, documented, approved by managers and logged.	Review naming conventions and their use effectively. Verify logs of redundant accounts. Interview Security managers. Review periodically with appropriate logical access documentation, which includes comparisons on account expiry, termination, and temporary access and timeliness authorization policies.
Process and Services are adequately controlled	Available processes and services installed need only required to process and services based on least functionality. Restrict the number of individuals with access to services based on least privileges. The function of processes and services are monitored, documented and approved by management.	Check procedures for optimized usage of processes and services. Interview the system administrator on – -Services installed and their requirement. -those who possess the access rights and their need. - Monitoring and updation of services and processes. -Scan for inadequately configured, redundant and hazardous processes and services.

Access to sensitive system resources is restricted and monitored.	Access and use of sensitive/privileged accounts have justified need aligned with valid business purpose. Logical access to these are to be adequately controlled- -Remote maintenance. -System libraries. -password/authentication services and directories are controlled and encrypted. -Access restriction based on time/location. -Segregation between user interface services and system management functionality.	Review policies and procedures used for sensitive/privileged accounts. Interview management personnel on access restrictions by testing the need and reasons for access. Review the accessing system activity logs maintained for – -personnel accessing system software, controls acquired to gain access. Access controls implemented in the operating system software, system libraries etc. Interview officials along with review related system documentation and coordinate the vulnerability analysis.
Appropriate and adequate media controls are to be implemented.	Only authorized users have access to printed and digital media removal or movement from the information system. Systems media is securely stored with respect to its sensitivity. If sensitive data are protected by approved equipment, techniques and procedures for disposal or exchange of information.	Ensure entity practices and review selected access logs. Review selected media transport practices and receipts. Check if media storage practices are adequate and comply with security parameters associated with information exchange.

Effective use of Cryptographic controls.	For integrity and confidentiality of critical data and programs they are to be protected using cryptographic tools. Based on risk of data communication encryption procedures are implemented. Authentication methods are implemented within the information system along with online or manual procedures for cryptographic key exchange and key management.	Evaluate the strength of cryptographic tools by with the help of experts. . Capture passwords or data transmitted over the network to evaluate their effectiveness. Interview appropriate officials to compare policies and procedures followed along with supporting documents. Evaluate the practices followed for cryptographic key exchange and management.

Database Controls

The current trends in application software design include the frequent use of a Database Management System (DBMS) to t actually handle data manipulation inside its tables, rather than let it be done by the Operating System (OS) software itself in flat files. The DBMS acts as a layer between the application software and the OS. The application passes on the instructions for manipulating data, which are executed by the DBMS, following the integrity rules and constraints built into the database definitions.

However, using a utility such as a text editor in the OS, the data in the DBMS can be manipulated directly, without the application. This can be done by using DBMS utilities and features, such as SQL (Structured Query Language)—if the user possesses the privileges to gain access to the DBMS.

Object granularity	DBMS has a finer granularity (relations, rows, columns, fields) than the OS (files, devices).
Semantic correlation among data	Relations between data pose a threat of security violations through inference.
Meta-data	DBMS provides metadata describing relations, attributes, domains, constraints, etc. OS provides limited or no meta-data.

Logical and physical objects	OS only deals with physical objects (files, devices). DBMS deals with logical objects, independent of OS objects (relations, views).
Multiple data types	OS only knows files and read, write and execute permissions. DBMS has many data types and operations, with separate access modes e.g. for individual access, grouped access, statistical operations, administrative operations, etc.
Static and dynamic objects	Virtual objects in a database, like views and query results can be used in the same way as physical objects.
Multilevel transactions	In OS an object can only have data of one security level. There is no need for polyinstantiation, which is not like what it is in databases.
Data life cycle	Data in a database must be stored securely and permanently right after the completion of the transaction that creates it. In the OS data is often not immediately stored securely.

Table: Differences between DBMS and OS

Threats to Databases are:

Loss of confidentiality: Protection of data from improper disclosure.
Loss of integrity: Information is protected from improper modification.
Loss of availability: Making data available to a user with a legitimate right.

Objectives of Database Controls

i. **Different degrees of granularity of access:** The DBMS offers access controls at various degrees of granularity, such as relations, columns, rows or individual data items.
ii. **Different access modes:** Typical database access modes are ***select***, ***insert***, ***update***, ***delete***. (select means "read", update means "edit")**.**
iii. **Different types of access controls:**

- **name-dependent:** depends on the name of the object.
- **data-dependent:** depends on the value of the object. (can be a constant value in a query).
- **context-dependent:** depends on other objects being accessed, on time, location of user, etc.

iv. **Dynamic authorization:** a user's authorization can be modified while the database is operational.

v. **Multilevel protection:** The DBMS should support multilevel protection through a mandatory policy.
vi. **Covert channels:** The DBMS should not have concealed channels.
vii. **Inference controls:** The DBMS should provide a way to assign classifications to aggregate information.
viii. **Polyinstantiation:** This mechanism allows the database to have multiple instances of objects, each having their own classification level.
ix. **Auditing:** Security-related events should be reported in a structured format such as system journals, audit trails and system logs.
x. **Flow controls:** Check the destination of output obtained through authorized access.
xi. **No back doors:** Access to data should be available only via the DBMS.
xii. **Reasonable performance:** Security controls should not increase the execution time significantly.

Integrity mechanisms in DBMSs

i. **Well-formed transactions**: Updates may occur only via transactions. (Correct execution is guaranteed via locking.)
ii. **Authenticated users**: Updates may only be provided by authorized and authenticated users. Authenticating users is typically performed by the OS and need not be duplicated in the DBMS.
iii. **Least privilege:** It should be possible to give users minimum update rights for their task.
iv. **Separation of duties**: No single user should be able to corrupt data. .
v. **Continuity of operation:** The DBMS should continue to function, without data loss, in case of disasters.
vi. **Reconstruction of events**: Improper behavior should be detected through audit trails(archieved logs) .
vii. **Reality checks**: This goes beyond the duty of the DBMS. But through constraints a few “impossible” data can be avoided and ensure data integrity.
viii. **Delegation of authority:** The DBMS should support ways to assign privileges according to mandatory or discretionary policies. Typically the SQ*L grant/revoke* statements are used to delegate authority.

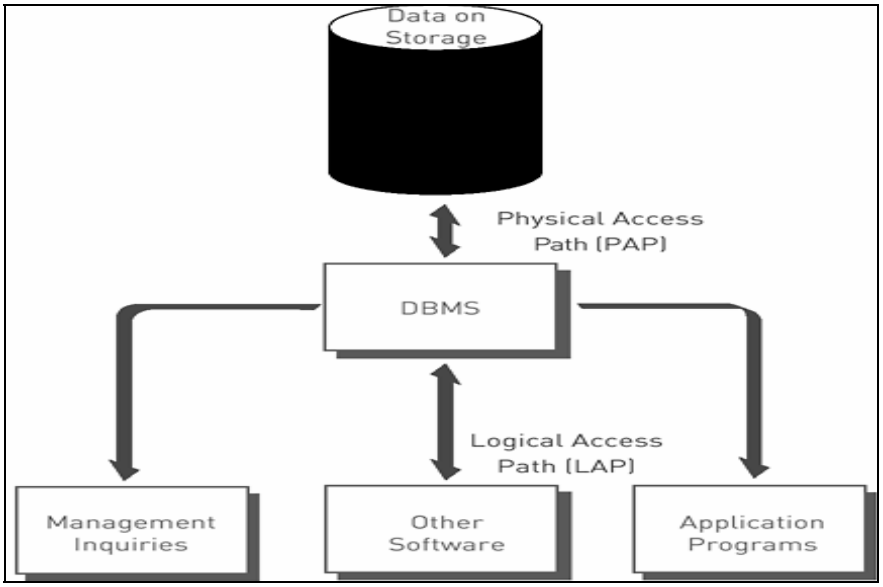


Fig.2.12: PAP and LAP paths of a DBMS

Database Roles and Permissions

Databases are very useful because they centralize the storage and maintenance of data, and limited access is both a responsibility and a benefit of this centralization. Databases are often separated logically by user access privileges. The database administrator specifies who should be allowed access to which data, at the view, relation, field, record, or even element level. The DBMS must enforce this policy, granting access to all specified data or no access where it is prohibited. Furthermore, the number of modes of access can be many. A user or program may have the right to read, change, delete, or append to a value, add or delete entire fields or records, or reorganize the entire database.

i. **Privileges:** Access control is based on the notion of privileges: that is; the authorization to perform a particular operation, to gain access to information in the database. While privileges enable restriction of the types of operations a user can perform in the database.

ii. **Roles**: This addresses the complexity of privilege management by providing user-defined collections of privileges that can be granted to (and revoked from) users and from other roles. One could create the role of a MARKETING MANAGER, and grant it all the needed privileges to perform his/her jobs, and then grant this role to all other marketing managers. A role can be a foundation for other roles. For example, a VP MARKETING role could be granted the basic

MARKETING MANAGER role plus additional privileges necessary all vice presidents in Marketing. Revoking a privilege would mean revoking it from all other marketing managers and VPs. A role can also be defined to prompt the user for a password when invoked, providing another layer of security.

iii. **Protection is done at the level of tables.** There are five access modes:

Read: to read tuples from a table. A user with read access may also define views on the table.

Insert: to add tuples to a table.

Delete: to delete tuples from a table.

Update: to modify existing tuples in a table. This privilege may be restricted to certain columns of a table.

Drop: to delete an entire table.

iv. **Administration of authority**: the creator of a table has all privileges on the table and can grant rights to others and revoke rights:

<s, p, t, ts, g, go>

s: subject or grantee.

p: the privilege to be granted.

t: the table on which access is to be granted.

ts: timestamp for the grant operation.

g: user who grants the privilege (the grantor).

go $\in$ {yes,no}: the grant option.

Views

While privileges allow control of the operations that a user can perform on database objects (such as tables), views enable further data access limitations. A view is a content or context-dependent subset of one or more tables (or views). A view might be created to allow a sales manager to view only the information in a customer table that is relevant to customers in their own territory. The view can contain selected columns from the base table(s) in which a subset of customer information is contained. A view can also be limited to a subset of the rows in the base table, such as a view of a customer table containing records of customers located in a certain territory

Perhaps the most commonly used method of controlling data access is views. Views are generally defined simply the result of a (SQL) query. This query, in turn, can pull information from many different tables and can also perform common calculations on the data. Although views provide many advantages to database developers, they can also be very valuable from the security standpoint. Views provide database administrators with a method to define granular permission settings that would not otherwise be possible.

The access rights with respect to a view are:

- The owner of a view has the same rights as on the base tables, plus the drop right.
- The owner of a view (on tables for which he has rights with the grant option) can grant others access rights on the view, even if they do not have access rights on the base tables.
- Access rights on base tables, given to the owner of a view after the creation of the view are not added to the view.
- Access rights on base tables, revoked from the owner of a view, are also removed from the view.

Stored Procedures

A stored procedure is a function / subroutine (group of SQL statements) that form a logical unit and performs a particular task. It is available to applications accessing a database system and is actually stored in the database. Large or complex data processing that might require the execution of several SQL statements is moved into stored procedures that are resident on the server and all applications can issue a remote procedure call (RPC. to these procedures. Therefore, stored procedures are used to consolidate and centralize logic that was otherwise implemented in applications which required movement of raw data for calculations.

Database Server compiles each stored procedure once and then reutilizes the execution plan which results in tremendous performance boosts. Stored procedures reduce the long SQL queries to a single line that is transmitted over the wire and, therefore, reduce the network traffic tremendously.

Typical uses for stored procedures include data validation (integrated into the database) or access control mechanisms. Carefully written stored procedures may allow for fine grained security permissions to be applied to a database. For example, client programs might be restricted from accessing the database via any means except those that are provided by the available stored procedures.

Triggers

Triggers are designed to be automatically "fired" when a specific actions/event takes place within a database. For example, a trigger is fired when an order entry in the "SalesOrder" table will automatically create a corresponding row in the Invoice table. Triggers can also be used to perform complex data validation. For example, one that checks to ensure that certain transactions are entered only at the end of a financial period. From the security standpoint, triggers can be used in the following ways: -

- To perform detailed Auditing (Audit Trail) - When a change is made in the "Employee Salary" table, a trigger may notify a high-level manager, or it may write a row logging this action to another table.
- To enforce complex database-related rules. For example, an appropriate trigger can ensure that a number of actions are always taken when data changes are made. .

Database Restrictions-Access Control Policy

One of the important objectives of access controls is to prevent any threats to the integrity of data and unauthorized access to the database resources. Relational Database works on the principles of tables and relations and allows rules of integrity and access to be specified. The principle of least privileges to data items can be enforced using views . Such rules can be restricted by a range of parameters such as permissible values or limits, operations up to the granularity of a data field, etc. Restrictions can also be implemented at schema level based on the following:

- **Discretionary**– users can specify those who can access the data they own and the action privileges they have.
- **Name-Dependent:** Users access to a data resource is restricted on the basis of their action privileges with respect to the data resource. For example, a payroll clerk is allowed to view all the data fields, except those related to, say, and employee's medical history.
- **The Access Matrix Model:** The row represents subject (user, account, program) and columns represents objects (relations, views, columns, etc.) a cell M(i, j) in the matrix represents the types of privileges.

R-Read,W-Write, X-Execute , OWN- Owner/Creator

Subjects	**Objects**			
	Table 1	*Table 2*	*View 1*	*View 2*
User 1	R,W,X		OWN,R,W,X	
User 2		R,W	R	

Table: Access Matrix

- **Content Dependent:** Access to resources can also be based on contents being queried/accessed. Where the access to a user is restricted to a particular degree of sensitivity of resources, he/she is not allowed to access the content of a higher degree of sensitivity. A petty cashier is not allowed access to transactions with value in excess of a prescribed limit.
- **Types of Privileges in SQL:** Account level: Privileges that each account holds independently of the relations in the database (CREATE TABLE, CREATE VIEW, DROP, and ALTER). Relation level: Specify which types of commands can be applied on each relation (Select/Modify/Reference) privilege on a relation.
- **Data Dependent Access Control:** This is an access-control decision based on the data contained in the records. For example, some users may be limited to viewing salaries which are less than Rs.30, 000. A manager may be restricted to viewing salaries of employees in his department. The two approaches to this access control are View-based access control and Query modification.
- **Context Dependent:** Using diverse queries, a user may be able to infer about content permissible only for higher access security. In order to secure against such improper access, access is restricted to all contextual references to data with higher security. For example, even though a user may not have access to income data of customers, he may query the database and obtain a list of customers making high value purchases, which allows him to infer the income of the customer.
- **History Dependent:** Sometimes a user may construct a series of queries in such a way that each query and its answer does not violate the security policy, but taken together the answer to all the queries would give her information to which she is not authorized. In addition to views and triggers, restrictions can also be technically applied by using menu based controls. That is access to data and action privileges can be restricted using well structured menus and restricting access to menu options based on the category of users.
- **Granularity of Access Control:** Access controls can be imposed at various degrees of granularity in a system. The entire database in a collection of relations, a relation, columns of one relation, few rows of a relation, few columns of some rows of a relation.
- **Mandatory:** – System administrator assigns security aspects to data that cannot be changed by database users.
- Classification level assigned to specific data attributes or relations in a record or to records or relations as a whole
- Clearance level assigned to users
- Security system compares 2 >= 1 OK

Application Software Controls in a Database

The integrity of a DBMS system depends in part on the controls implemented in the application programs that provide the interface to the user to perform a job process activity with a sequence of commands and update parameters that are passed with respect to certain considerations or actions. Hence to deal with the controls that affect the data integrity are:

i. Update Protocols:

- Sequence check order of transaction and master files.
- Ensure that all records on files are processed.
- Process multiple transactions for a single record in the correct order.
- Maintain a suspense account.

ii. Report Protocols

- Print control data for internal table (standing data..
- Print run-to-run control totals.
- Print suspense account entries.

Concurrency Controls in a Database

One of the most important goals of a DBMS is to allow users to share the same data resource. So integrity becomes a critical issue, for it leads to the deadlock problems called lockout, circular wait, no preemption and priority request. Solution that prevents Deadlock is prevented by two-phase locking that before a transaction can read a data item " read-lock" and before a transaction can write a data item "write-lock". In case of a distributed enterprise DBMS environment the deadlock is prevented by using a replicated copy of the database at strategy sites or by database fragmentation, i.e., into non-overlapping partitions.

Cryptographic Controls

Data storage integrity is done by using the block encryption method. While the stream encryption method requires extra data be accessed and slows down retrieval. Data stored on portable media is encrypted by a secure encryption device part of the controller of the device. To protect privacy of a users' data even when media is stolen, the cryptographic keys are used. To facilitate ease of sharing data the schemes used are file key, secondary key, and master key. To protect access to the secondary key a password or authorization mechanism using hardware, software and manual methods are implemented.

File Handling Controls

To prevent accidental destruction of data contained on a storage medium controls are implemented by using hardware, software, and also manual methods. These are:

- Internal Label: To identify a table, file or a database by the application program access.
- Generation Number: Version of the backup.
- Retention Date: Prevent overwriting of a table, file or a database.
- Control Totals: Check sum to ensure correct file or record being accessed.
- Read only switches: These are plastic tabs on devices which slide to open or close a reading hole.
- External labels: Labels on the storage devices that assist users by providing information about database name, creation, transaction file, back up info, etc.

Audit Trail Controls

The need for trail is to maintain the chronology of events that occur in the database to replay the current state of a database. The accounting and operational trails help to determine the sequence of events and resource usage of a DBMS.

i. **Accounting Audit Trail** maintains the chronology of events that occur in the database definition or in the database itself with the following two operations:
   - *Implosion operation:* data can be traced from its source to the items it affects.
   - *Explosion operation:* the sequence of events that have occurred in a data item in the database definition or the database can be reconstructed.
   - *Unique time stamp:* can be put on all transactions.
   - Attach *before and after images* of the data item against which the transaction is applied.
   - Facilities to define, create, modify, delete and retrieve data in the *audit trail*
   - *Retention time* for the audit trail with respect to business policy.

ii. **Operational audit trail** Maintains the chronology of resource consumption events that affect the database or its definition. Database Administrators use the operation audit trail to determine when the database needs to be reorganized or when the processes that access the database need to be rewritten to improve their efficiency.

Existence Controls (Recovery)

The cause of destruction or damage to a database can be one of the following:

- Application program error
- System software error
- Hardware failure
- Procedural error
- Environmental failure

Existence controls in the database system must restore the database in the event of loss by backup and recovery strategies.

i. **Backup strategies:** All backup strategies require maintenance of a prior version of the database and a log of transactions or changes to the database. [While an update creates new physical version of the backup is retained otherwise a periodic dump of the database is maintained Very confusing sentence. Please rethink and rewrite] Options for how a database is backed up and restored are:

   - A transaction log backup copies only the transaction log.
   - Differential backup copies only the database pages modified after the last full database backup.
   - (A file or file-group restore?) allows the recovery of just the portion of a database that was on the failed disk.
   - A full database backup is a full copy of the database.

ii. **Recovery Strategies**

   - Roll-forward operation: in which the current state of the database is recovered from a previous version.
   - Roll-back operation: in which a previous state of the database is recovered from the current one.

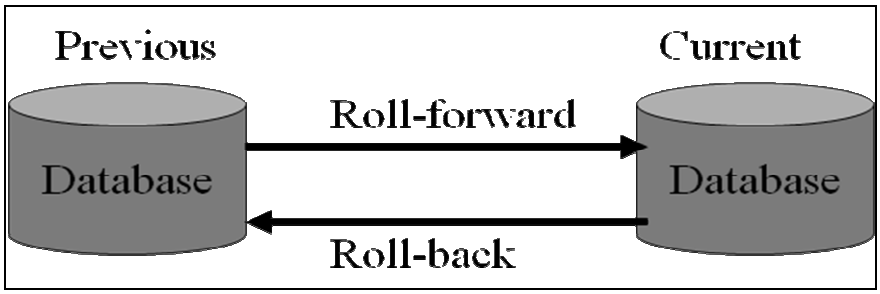
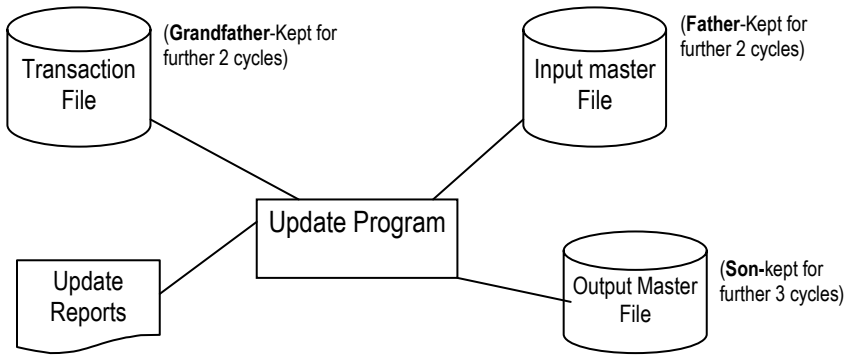


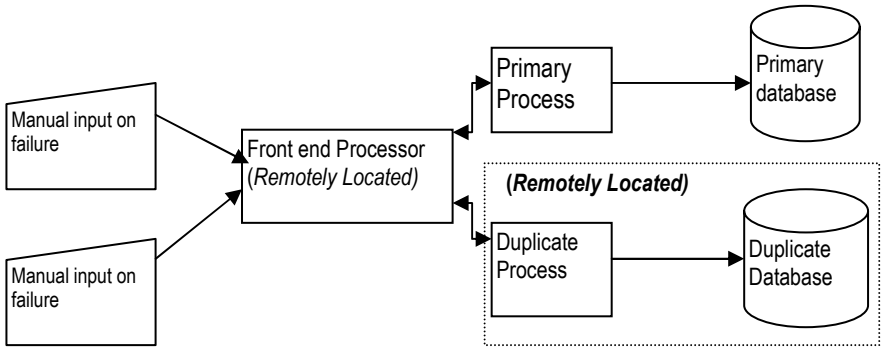
Fig.2.13: Recovery Strategies

Existence Control Strategies (Backup and Recovery)

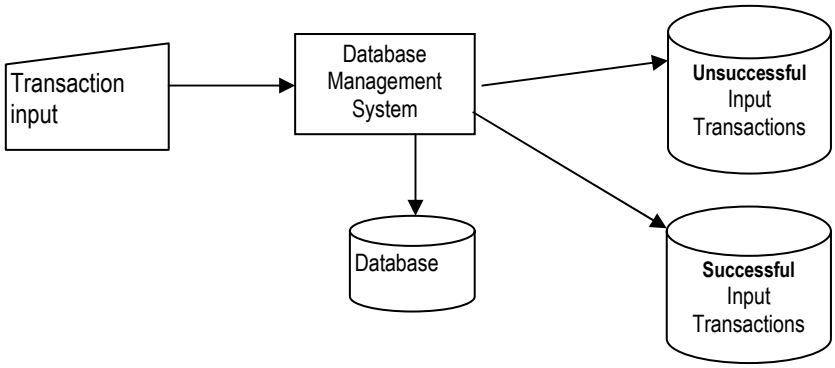
i. Grandfather, Father, Son Strategy



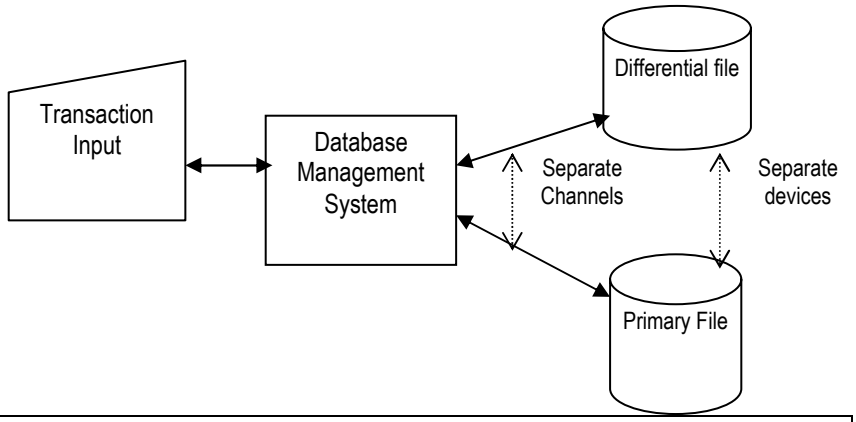
ii. Dual recording/mirroring strategy



iii. Separate logging of successful and unsuccessful input transactions



iv. Differential file strategy for backup and recovery



Did you Know?

Phishing is a cyber crime by which the victims are directed to a phony telephone call number instead of a website!
Victims are asked to call a phone number to get information on reactivation of their credit/debit cards

Few examples of Logical Access Control Techniques for Databases and their Suggested Audit Procedures Control Activity	**Control Techniques**	**Audit Procedures**
Implement an effective DBMS system with consistent control requirements in the application and data. The strategy is to address: -Data management -Middleware	Appropriate functions are implemented in both physical and logical structure of data storage and retrieval procedures. Isolate production DBMS from non-production DBMS and other DBMS requiring medium controls.	Check the documentation of the DBMS linked with the application. Assess the design and implementation of the DBMS.

-Cryptography -Data Warehouse -Data Reporting and -Data Extraction.	Ensure the database schema is consistent with the organization of data and functions that align with the access limitations imposed on different groups of users.	Verify that all access paths to sensitive data and administrative functions are identified and satisfactorily controlled.
Detective controls are implemented to effectively identify requirements and action to specific system or user activity within the system.	To accurately identify historical system activity and data access with adequate logging and monitoring controls.	Identify the critical security events that are logged and determine the adequacy of controls to monitor the audit logs and detect abnormal activity.
The specialized control requirements of data management processes to facilitate interoperability between applications, ad-hoc reporting or other applications not integrated into the system.	For correction and/or detection of data anomalies effective data accuracy and completeness controls are in place. The system connectivity configuration that facilitates application to application and application to non-integrated functions are controlled for limited access.	The facility used for interoperability is to be identified and gain understanding. To understand the exchange of data and interconnectivity between different DBMS. Assess and test the general controls related to DBMS (Data design, storage, exchange format etc.)

Audit Trail

An Audit Trail enables the reconstruction and examination of the sequence of events leading to a transaction, from its inception to its final results or from output to the initial trigger to the events resulting in the transactions. Audit trails maintain a record of-

- Computer systems and users.
- Security violations and segregation of duties.
- Events at various levels, operating systems, network component, application and database.
- Events logged at the OS and Network component level -System-level logs.

- Events logged at application level-Application level log.

For a complete reconstruction of the scenario the application level audit trail augments the system level log by logging user activities including specific actions at the data level.

Audit of Access Controls

Some of the factors critical to the successful achievement of audit objectives with regard to the evaluation of logical access are:

- The *understanding of an organisation's information security framework*, security policy linkage of IT objectives to its business objectives and assessment of risk and controls. This often forms the foundation for risk management and criteria for determining the amount of investment and the philosophy of access controls.
- *Selection and implementation of appropriate access controls* should be consistent with the organisation's structure, management controls and organisational culture.
- *Top management's commitment*, support and control must be communicated to all levels in the organisation and concerned stakeholders. Management support is reflected in the emphasis and investment on training and education of users, the importance given to access controls and the enforcement of access control discipline.
- *Management controls* should be evaluated to determine if adequate systems are in place. This also helps to detect reports and take corrective action on access security violations.

The auditor's opinion depends on his understanding of the general and IS controls and audit procedures. These are used to test the effectiveness and efficiency of access to organizational information, in respect of which the auditor should exercise due care and diligence.

Audit Procedures: Special Considerations

The objective and scope of audit would determine the audit procedures and IS resources to be covered. Often evaluation of logical access controls forms a part of a generic IS audit, covering various other controls. However, the auditor may be required to evaluate the logical access security of a system, sub- system, application, operating software, database management software and the like.

Identification of logical access paths

An auditor has to identify the possible access paths permitting access to information resources. He must document the logical paths and prescribe appropriate audit

procedures to evaluate every component in the information systems infrastructure to enable identification of logical access paths. This is often a challenging and complex task, when it comes to auditing in networking computing environments. Identification and documentation of access paths involves testing security at various layers:

i. **Hardware:** This includes computer workstations, terminal devices, communicationdevices, and peripheralsconstituting the physical interface with the users. Physical security of hardware forms a critical and inseparable component of information security, which works in tandem with logical security and other category of controls. The auditor has to have special consideration for covert channels and communication devices (e.g. modems, network interface cards) connected to computers.
ii. **Systems software:** The command and control of hardware rests on the proper implementation of operating systems and other systems software. From a logical perspective, a wrong setting of systems level parameters can compromise the security of the application and other systems software, which talk to the systems software. Besides, firmware and systems software often enforce the first level access control including logon and password security. Systems with communication facilities enable sharing of resources, but present higher level of risks. The IS auditor should take special care to ascertain which devices are connected and provided with communication capabilities. He would need to evaluate the management standards and implementations of network access controls.
iii. **Database Management System:** In environments involving voluminous data handling, a Database Management System (DBMS) manages the organisation of data in the databases. The auditor is required to evaluate the access security enforced by it, which could include schema definitions, access to data dictionary, directory services and scripts to restrict access implemented by it.
iv. **Application software:** Application software represents the business logic, which interfaces with the user and business process requirement, from the user perspective. The auditor focuses on the effectiveness of boundary controls and other input, processing and output controls, discussed elsewhere in this module.
v. **Access control software:** The auditor may also encounter situations in networked environments with users having access to various applications. In such cases user and program access to applications and IS resources are controlled by an access control software. The auditor should evaluate the access permissions configured in the software and ascertain their appropriateness to the organisation's functional requirements.

In the above cases, the assessment of the state of access controls can be quite technical, and often complex for the IS auditor. Therefore, in cases, where the

controls involve technical sophistication, the auditor has to rely on the services of a technical expert, who possesses special skills, knowledge and experience in the relevant field. Where the auditor relies on the work of an expert he is expected to evaluate the work and its extent of reliance on the work of the expert. Attention in this regard is also invited to AAS 9(SA-620) on "Using the work of an Expert". The auditor's report should explicitly state the fact of IS auditor's reliance on the work of other experts.

Audit Test Procedures

IS auditors should evaluate whether logical access policies and procedures are effectively implemented. For this the auditor has to test if:

- Necessary access control framework in the form of logical access security policies and standards are in place. The auditor does this by studying various organizational acess policies, the system of education and training for users, and the levels of their knowledge with respect to logical security of information resources.
- Procedures and mechanisms for logical access are properly implemented. The IS auditor has to evaluate the various logical security techniques and mechanisms for their effective implementation, operation and administration.
- The auditor needs to conduct compliance and substantive tests to determine if the logical security of information resources is actually effective by examining procedure manuals such as administrator manuals and user manuals, interviewing users and administrators to assess any weaknesses or incompatibilities.

Some audit considerations in this regard are outlined below:

Systems Configuration

Test the appropriateness of system configuration and parameter settings. It is because appropriate configurations of access security parameter systems at the time of installing/ upgrading hardware, system software such as operating systems, DBMS and application software are critical for building a strong foundation for access security. In this respect the auditor would have to evaluate whether:

- The system configuration complies with the organisational security standards, security policy requirements, manufacturer specifications and best practices for security.
- There is a process to ensure that configuration of access security settings and parameters and changes thereto are authorised, documented and tested.
- Privileged and special purpose logons are controlled and documented.

- There is a procedure for control over purchase, custody and management of system utilities. Many systems utilities are powerful and can break through the various levels of access security.

Logical Access mechanisms

For testing various logical access mechanisms such as token based authentication systems and biometric access control systems, the auditor should conduct tests that determine:

- The control of authorisation, operation and termination overuse of tokens such as memory and smart cards.
- Control over special terminals and devices. For instance, a hub may be exposed physically but with proper levels of encryption, logical security of information can be ensured.
- Security practices with respect to unattended terminals, security of data in transit and control over production resources.
- Whether logging of transactions and events is appropriately enabled.

i. User account management and password management

Logon and passwords are the most commonly used mechanisms to secure logical access to information resources. The auditor should

- Evaluate mechanisms such as access control features and software to identify weaknesses, if any.
- Evaluate the effectiveness of user management procedures through audit of access control lists to assess if access is permitted according to the principle of least privileges and "need to know-need to do" basis, scan audit logs to determine the effectiveness of access control and interview users, and identify all entries in ACL with the authorized list of employees permitted to access systems.
- Test user profiles and group profiles to determine the access privileges and controls.

ii. Privileged logons and special user accounts

Privileged logons and special user accounts provide higher level of access to systems resources. Hence, they require a higher level of access security and management. The auditor should evaluate:

- The strength of controls on such privileged access. He should identify all individuals having access to such privileged logon facilities and special user accounts, and critically evaluate the need for such access.

- Review audit trails, access violation reports in respect of all privileged logons and special user accounts.
- The strength and adequacy of monitoring and incident handling procedures.

iii. Access to file directories and application logic and system instruction sets The auditor should evaluate the protection of

- Systems files and directories containing critical hardware and systems software configuration and parameter files such as driver information, etc.
- Application files and directories containing application programs, support files, program libraries, parameter files, initialisation files, etc.
- Production data and directories containing production files and production resources.

iv. Bypass Security Procedures

There may be various situations in the routine course of operations where security features are bypassed for operational and functional convenience during certain controlled operations. For instance, privileged logons may be provided to systems engineers to meet emergency situations, bypass label processing may be provided to meet certain bulk processing requirements or systems exits may be enabled during software implementation and maintenance phases. The auditor should identify all such provisions and critically audit the events.

Summary

Logical access controls involve securing both stored and transmission data. In order to secure this data, one of the key steps is authentication. Authentication involves ensuring the genuineness of the identity claimed by the user. The chapter has demonstrated various techniques to provide internal and external access controls. They are systems with varying degrees of reliability, precision, sophistication and cost.

The common access control technique is logon IDs and passwords. In token-based authentication the user possesses identification to enable authentication. Biometrics offers a authentication based on “what the user is”, based on characteristics of the human body e.g. fingerprints, facial scans, etc.

Any compromise of operating systems processing can lead to improper access to system resources. In order to safeguard against improper access, a concept of reference monitor implements logical control over access to objects. Reference monitor is an abstract mechanism that enables enforcing the security policy. Relational database security works on the principles of tables and relations and

allows rules of integrity and access to be specified. The principle of least privileges to data items can be enforced using views as against reads.

The audit steps involved are identification of logical access paths at all levels like hardware, system software, and database management and application control. The various components to be evaluated during such audit are:

- Testing the appropriateness of system configurations and parameter settings.
- Testing various logical access mechanisms, evaluation of user account management and password management.
- Evaluation of access to file directories and application logic and system instruction set.
- Adequacy of monitoring and reporting procedures and bypass security procedures.

Master Checklist on Logical Access Controls

The following is an illustrative questionnaire that could be used to review Logical Access Controls within operating systems and databases

No	Checkpoints
	User Access Management Policy and Procedure
1.	Check if the user access management policy and procedure have been documented.
2.	Whether the user access management policy and procedure are approved by the management.
3.	Whether the user access management policy and procedure document includes: - Scope and objective. - Procedure for user ID creation, approval, review, suspension, and deletion. - Granting access to third parties. - Password management. - User access rights assignment & modifications. - Emergency access Granting. - Monitoring access violations. - Review and update of document.
	User Access Management
1.	Check whether User ID & access rights have been granted with approval from appropriate level of IS and functional heads *(Verify the user ID creation, granting of access right and approval process)*

No	Checkpoints
2.	Check whether the organization follows the principle of segregation of duties adequately in granting access rights *(Verify Access rights should be given on need to know and need to do basis – without unchecked concentration of power.)*
3.	Make sure that User IDS are in a unique format. *(Verify the naming conventions for the user IDs)*
4.	Check whether invalid log in attempts are monitored and User IDs are suspended after specific number of attempts? *(Verify the parameters set for unsuccessful log in attempt)*
5.	Ascertain if the organisation follows the principle of complex composition for password parameters. *(Complex composition of password parameter should be used as to make guesswork difficult and thus prevent unauthorised users from getting access. Ee.g., a special character and numbers should form a password., Restrict the use of organisation's name, 123, xyz or other generic terms as password..*
6.	Check whether granting access to the third parties is according to the User Access Management policy and procedure *(The organization should specify and implement a process for granting access to third parties like contractors, suppliers, auditors, consultants, etc.)*
7.	Check that users are forced to change password on first log-on and periodically. V*erify password parameters for first log on and password aging).*
8.	Check if the organisation has implemented the clear screen and clear desk policies *(Terminals should be automatically logged off if remaining idle for specific time.)*
9.	Has the organisation restricted the concurrent log- on? *(One user ID should not be allowed to be logged-in for two different terminals at the same time)*
10.	Are the users' IDs shared? *(Verify whether users' IDs are shared among the employees/ users or not?)*
11.	Are multiple user IDs allocated to a single individual?
12.	Are user access policy and procedure documents communicated / available to the respective users?

No	Checkpoints
13.	Are User IDs and Passwords communicated to the users in a secure manner? *(Verify the procedure for communicating user ID and password for the first time and after suspension).*
14.	Check if the organisation reviews user IDs and access rights periodically.
15.	Does the organisation monitor logs for the user access?
16.	Are policies and procedure documents reviewed and updated at regular intervals?
17.	Is access to scheduled jobs restricted to the authorised?
18.	Is the emergency user creation made according to the policy and procedures for User Access Management? *(Verify the emergency access granting procedure, including approvals and monitoring).*
19.	Whether periodic review process ensures user accounts align with business needs and removal on termination/transfer. (*Review and evaluate procedures for creating user accounts and ensure that accounts are created only when there's a legitimate business need and that accounts are removed or disabled in a timely fashion in the event of termination or job change.)*
20.	Check if passwords are shadowed and use strong hash functions *(Ensure the strength of passwords and access permission to password files. Review and evaluate the strength of system passwords and the use of password controls such as aging.)*
21.	Review the process for setting initial passwords for new users and their mode of communication and evaluate the tracking of each account to a specific employee.
22.	Does the use of groups and access levels set for a specific group determine the restrictiveness of their use? (Evaluate the use of passwords, access rights at the group level)
23.	Ensure that the facility to logon as super/root user is restricted to system console for security reasons.
24.	Check whether the parameters to control the maximum number of invalid logon attempts has been specified properly in the system according to the security policy.

No	Checkpoints
25.	Check whether password history maintenance has been enabled in the system to disallow same passwords from being used again and again on a rotation basis.
26.	Verify the parameters in the system to control automatic log-on from a remote system, concurrent connections a user can have, users logged on to the system at odd times (midnight, holidays, etc. and ensure whether they happen according to the set security policy.
	Maintenance of sensitive user accounts
1.	Ascertain as to who is the custodian of sensitive passwords such as super/root user and verify if that person is maintaining secrecy of the password, and the password has been preserved in a sealed envelope with movement records for usage in case of emergency.
2.	From the log file, identify the instances of use of sensitive passwords such as super user and verify if records have been maintained with reason for the same. Ensure that such instances have been approved/ authorized by the management.
3.	From the log file, identify the instances of unsuccessful logon attempts to super user account and check the terminal ID / IP address from which it is happening. Check if appropriate reporting and escalation procedures are in place for such violations.
	Database controls
1.	Check if the policy and procedure documented and approved for database activities is being followed.
2.	Check if the policy and procedures cover the following: - Appointing administrator - Conventions for database creation, storage, naming and archival - Monitoring of triggers and queries to prevent overloading of database - Configured to ensure audit trails, logging of user sessions and session auditing - Reconciliation between source and receiving system in case of interface - Review of activities by admin.
3.	Are the policy and procedure documents available to respective users?
4.	Are the policy and procedure documents reviewed and updated at regular intervals?

No	Checkpoints
5.	Has the organization assigned administrators and users to the database?
6.	Has the IS Department laid down standards / conventions for database creation, storage, naming and (archival?)?
7.	Check if the vendor-supplied passwords to the default users have been changed. *(Verify the removal of demo user, guest users and demo databases removed.*
8.	Check if the design or schema of tables/ files in database contains fields for recording makers, checkers and time stamp
9.	Have standards been set for database control reports to ensure accuracy and integrity of the databases *(Verify the control total / reports like Total of transactions and balances, record counts and hash totals).*
10.	Check the reconciliation between the source and receiving system for critical information transferred through interface system
11.	Verify that database permissions are granted or revoked appropriately for the required level of authorization. *(Review database permissions granted to individuals instead of groups or roles and are not implicitly granted incorrectly.)*
12.	Review the execution of dynamic SQL in stored procedures and ensure that row-level access to table data is implemented properly.
13.	Check if PUBLIC permissions are revoked when not needed. (Restrict access to the operating system, the directory to which the database is installed and the registry keys used by the database.)
14.	Verify that encryption of data-at-rest is implemented appropriately. *(Ensure that encryption key management is part of the disaster-recovery plan.)*
15.	Verify that the database is running a current version that the vendor continues to support. *(Ensure procedures to maintain database integrity by use of root kits, viruses, backdoors, and Trojan horses.)*
16.	Doe the IT Department identify and segregate hardware hosting these databases?
17.	Check if there is a clear partition between application area and data areas within the system.

No	Checkpoints
18.	Does the IT Department have laid down standards / conventions for database creation, storage, naming and archiving?
19.	Are users denied access to the database except through the application?
20.	Are direct query / access to database restricted to the concerned database administrators?
21.	Check if triggers and large queries monitored to prevent overloading of database and consequent degradation of database performance are in place.
22.	Have all the vendor-supplied passwords and default passwords changed? Have all the demo user and demo databases been removed?
23.	Are there controls on sessions per user, number of concurrent users, etc? Is creation of users restricted and need based? Are the rights granted to users reasonable and based on requirement? Is the database configured to ensure audit trails, logging of user sessions and session auditing?
24.	Does the administrator maintain a list of batch jobs executed on each database, severity of access of each batch job and timing of execution? Are Batch Error Logs reviewed and is corrective action being taken by the Administrator periodically?
25.	Is there a separate area earmarked for temporary queries created by power users or database administrator based on specific user request? Are temporary sub databases created removed periodically or after the desired purpose has been achieved?
26.	Does the design or schema of all tables / files in database contain fields for recording makers, checkers and time stamp? Are database administrators rotated periodically? Does the organization have confidentiality undertakings from external service providers?
	Referential Integrity and Accuracy
1.	Is there a standard set of database control reports designed in consultation with the user department for ensuring accuracy and integrity of databases?
2.	Are these reports run directly from the back-end database periodically and the results both positive and negative are communicated by the administrators to senior management?
3.	Are these reports run periodically and taken directly by the User Department to ensure accuracy?

No	Checkpoints
4.	In case of automated interface between systems is there a system of reconciliation between the source and receiving system for critical information?
5.	In cases where data is migrated from one system to another has the user department verified and satisfied itself about the accuracy of the information migrated? Is there a formal data migration report?
6.	Are entries made directly to the back end databases under exceptional circumstances? Is there a system of written authorization?
7.	Are entries in the database e updated / deleted due to any exceptional circumstances (e.g. during trouble shooting, etc.), approved in writing and recorded?
	Administration and House Keeping
1.	Does the administrator periodically review the list of users to the database? Is the review documented?
2.	Are inactive users deactivated?
3.	Is there a back-up schedule?
4.	Are databases periodically retrieved from the back up in test environment and is accuracy being ensured?
5.	Are senior personnel from the user department involved in testing backup retrieval?
6.	Is there a periodic purging / (archival?) of databases?
	LOGICAL DATA SECURITY
1.	Check whether the data accessed on the least privilege basis is established by the data owner
2.	Check if a clear definition of data access matrix has been established
3.	Are access privileges periodically reviewed by data owners?
4.	Check if authorized access to sensitive data is logged and the logs are regularly reviewed to assess whether the access and use of such data was appropriate
5.	Are unauthorized access attempts detected?
6.	Check if encryption is being used for sensitive and/or proprietary data.
7.	Check if authentication mechanism, such as passwords and tokens, are required for data access.

No	Checkpoints
8.	Check if the security administrator is notified of employees who have changed roles and responsibilities, transferred, or been terminated and access privileges of such employees immediately changed to reflect their new status.
	Checklist for Auditing file Security and Controls
1.	Evaluate the file permissions for a judgmental sample of critical files and their related directories.
2.	Look for open/shared directories (directories with permission set to read, write, execute) on the system.
3.	Evaluate the security of all system administrator access/login files on the system, especially those that are to "root"/sys-admin. Review and evaluate the security of the operating system.
4.	Ensure that all files have a legal owner in the system access list. Ensure also that system level commands are not to be used by users to compromise user accounts. Examine the system's default scheduled jobs; especially root's/sys-admin, for unusual or suspicious entries.

S.No	Operating System Security
1.	Obtain the system information and service pack version, and compare with policy requirements.
2.	Determine if the server is running the company-provisioned firewall and antivirus program.
3.	Ensure that all approved vendor-support patches are installed as per the server management policy approved by the management.
4.	Review and verify startup information.
5.	Determine what services are enabled on the system and validate their necessity with the system administrator. For necessary services, review and evaluate procedures for assessing vulnerabilities associated with those services and keeping them patched.
6.	Ensure that only approved applications are installed on the system as per the server management policy.
7.	Ensure that only approved scheduled tasks run on the system.
8.	Review and evaluate procedures for creating user accounts and ensure that accounts are created only when there is a legitimate business need. Also

	review and evaluate processes for ensuring that accounts are removed or disabled in a timely fashion in the event of termination or job change.
9.	Ensure that all users are created at the domain level and clearly annotated in the active directory. Each user should trace to a specific employee or team.
10.	Evaluate the use of password controls on the server, such as password aging, length, complexity, history, and lockout policies. Evaluate the use of user rights and security options assigned to the elements in the security policy settings.
11.	Review and evaluate the use and need for remote access, including connections like FTP, Telnet, SSH, VPN, and other methods and see to it that a legal warning banner is displayed when connecting to the system.
12.	Review and evaluate system administrator procedures for monitoring the state of security on the system.
13.	Ensure that the server has auditing enabled aligned with the best practices of a standard security policy or organization's practices
	Auditing Clients/Hosts on the Enterprise Network
1.	Determine if the client is running the company-provisioned firewall and antivirus program.
2.	Determine if the client/host is equipped with the minimum recommended service pack antivirus software.
3.	Review and evaluate the clients/hosts with a basic security analyzer and a commercial-grade network scanner.

Questions:

1. Logical access controls within an organization's enterprise-wide information system are implemented to protect its information assets that include ……
   a. Data stored, Data backup and Data in transit.
   b. Data processed, Data devices, Data wires.
   c. Data forms, Data plans, Data memory.
   d. Data schema, Data size, Data layout.

2. ……..is the most difficult database access control to implement. .
   a. Content-dependent access control
   b. Name-dependent access control
   c. History-dependent access control
   d. Granularity of Access Control

3. Identify the application level software controls used in a database to maintain data integrity.
   a. Network protocols
   b. ICMP protocols
   c. Re-process protocols
   d. Update protocols

4. Which one of the following is not an objective of file handling controls?
   a. To ensure data is retained for a certain period
   b. To prevent data from being accidentally overwritten
   c. To ensure incompetent access by programs into the data
   d. To ensure the correct version of the file being loaded

5. ……..is a technical logical exposure which involves unauthorized modification of manual input data being submitted for processing.
   a. Wire tapping
   b. Data Diddling
   c. Piggybacking
   d. Key logging

6. When a hacker attempts to flood memory buffers and communication ports on the network which results in blocking the delivery of normal services is called…...
   a. Social engineering.
   b. Masquerading.
   c. Data Dumpster.
   d. Denial of service.

7. The Trojan horse malicious code is a program that……
   a. Blow-up on occurrence of a logical event.
   b. Carry out program commands automatically.
   c. Hides inside a host program.
   d. Attacks network devices.

8. ….....is a process by which a user provides a claimed identity to access a system.
   a. User authorization
   b. User registration
   c. User identification
   d. User logging

9. User access to applications with respect to their job responsibilities is a logical access control called…….
   a. Privilege management
   b. User password management

c. Equipment management
d. Network management

10. Maintenance of event logs across an enterprise network plays a significant role in correlating an event and generating report using the…… application control.
    a. System monitoring
    b. Clock synchronization
    c. Flood synchronization
    d. Network isolation

11. One of the weaknesses of the password logon mechanism is…….
    a. Repeated use of the same password
    b. Periodic changing of password
    c. Encrypted password
    d. One user one password

12. Facial scan, iris and retina scanning are used in ……..
    a. Smart tokens
    b. Biodirect security
    c. Backup Security
    d. Biometric Security

13. The…… provides system administrators with the ability to incorporate multiple authentication mechanisms into an existing system through the use of pluggable modules.
    a. Personal Authentication Module
    b. Pluggable Authentication Module
    c. Password Processing Module
    d. Login identification Module

14. The access privileges of a user for two entities say A and B for read and write are maintained in the ……within an application.
    a. Actual access control list
    b. Acquired control entry
    c. Access control list
    d. Secret policy entry

15. A ……can help a sales manager to read only the information in a customer table that is relevant to customers in their own territory.
    a. View
    b. Procedure
    c. Trigger
    d. Table

16. Which one of the following is a logical control method?
    a. Work area separation
    b. Policies and Procedures
    c. Audit Trail
    d. Order form data entry

17. ……is an integrity mechanism in DBMS which helps in continuous function without data loss in case of disaster.
    a. Well-formed transactions
    b. Continuity of operation
    c. Flow Controls
    d. No backdoors

18. Data storage integrity is maintained by using the …….. encryption method.
    a. Stream
    b. End-End
    c. Block
    d. Link-Link

19. The implosion operation method is an accounting audit trail method used in DBMS to…….
    a. Maintain a log of backup of data in the DBMS
    b. To trace triggers error of the DBMS
    c. To trace a data from its source to the items affected.
    d. To trace the business policy of DBMS

20. When a manager is restricted to view or access the salary details of the employees in his department only, the discretionary access is……
    a. Name-Dependent access control
    b. Content-Dependent access control
    c. Context-Dependent access control
    d. Data-Dependent access control

21. Name the DBMS control that read-locks a data item before a transaction can read and implements a write-lock before a transaction can update a data item.
    a. Update control
    b. Trigger Control
    c. Concurrency control
    d. Commit control

22. ……is the control that restores a database from hardware failure, environmental failure or a system software failure.
    a. Application control
    b. Existence control

c. File control
d. Update Control

23. The differential backup recovery strategy is ……..
    a. Previous state of the data files
    b. Pages or files modified after a full backup
    c. Roll-back transaction log backup
    d. Pages or files updated before a full backup

24. In case of automated interface between database systems- To check if there is a system of reconciliation between the source and receiving system for critical information? – is a checklist entry to audit …….of a DBMS.
    a. User management
    b. Recovery and Backup
    c. Referential Integrity
    d. Roll-forward Backup

25. Whether invalid log-in attempts are monitored and User IDs are suspended on specific number of attempts. – is a checklist question under the …….control mechanism.
    a. User Access management
    b. Operating system management
    c. Transaction log management
    d. Database retrieving management.

Answers:

1. a	2. c	3. d	4. c	5. b	6. d	7. c	8. c	9. a
10. b	11. a	12. d	13. b	14. c	15. a	16.c	17.b	18.c
19.c	20. d	21.c	22. b	23. b	24. c	25. a		

3 Network Security Controls

Learning Objectives

- The characteristics of a network
- The threats to a network
- The controls that protect a network

Introduction

In this section, we examine the risks and controls that are specific to networked computers. It is rare these days to find a standalone computer in any commercial environment, as networks offer tremendous advantages that far outweigh the cost of creating them. However, networks are also far more vulnerable to external and internal threats than standalone systems. The internet, while offering tremendous advantages, also poses several security challenges to organizations. In this section, we shall look at the threats and risks that arise in a networked environment and the controls and countermeasures that prevent or mitigate such risks.

Network Characteristics

The characteristics of a network are:

- **Anonymity:** A network removes most of the clues, such as appearance, voice, or context, by which we recognize acquaintances.
- **Automation:** In some networks, one or both endpoints, as well as all intermediate points, involved in a given communication may be machines with only minimal human supervision.
- **Distance:** Many networks connect endpoints that are physically far apart. Although not all network connections involve distance, the speed of communication is so fast that humans cannot usually tell whether a remote site is near or far.
- **Opaqueness:** Because the dimension of distance is hidden, users cannot tell whether a remote host is in the room next door or in a different country. In the same way, users cannot distinguish whether they are connected to a node in an office, school, home, or warehouse, or whether the node's computing system is large or small, modest or powerful. In fact, users cannot identify if the current

communication involves the same machine with which they communicated before.

- **Routing diversity:** To maintain or improve reliability and performance, routings between two endpoints are usually dynamic. That is, the same interaction may follow one path through the network at an instance and a different path another time. In fact, a query may take a different path from the response that follows a few seconds later.

Threats and Vulnerabilities

This section describes the various kinds of vulnerabilities and threats associated with networks that aim to compromise the confidentiality, integrity, or availability of data, and software and hardware by non-malicious and malicious attackers. The threats and vulnerabilities are listed under the following heads:

- Information Gathering
- Communication Subsystem Vulnerabilities
- Protocol Flaws
- Impersonation
- Message Confidentiality Threats
- Message Integrity Threats
- Web Site Defacement
- Denial of Service

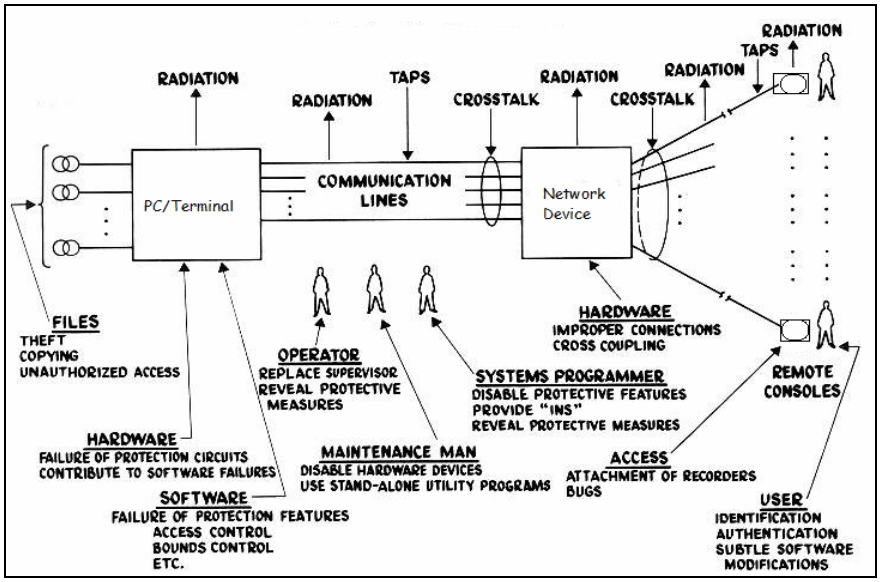


Fig.3.1: Network Vulnerabilities

However, most of these threats operate in tandem and it is difficult to associate them with network security alone.

Information Gathering

A serious attacker will spend a lot of time obtaining as much information as s/he can about the target before launching an attack. The techniques to gather information about the networks are examined below:

i. **Port Scan**: An easy way to gather network information is to use a port scanner, a program that, for a particular IP address, reports which ports respond to messages and which of the several known vulnerabilities are present.
ii. **Social Engineering**: Social engineering involves using social skills and personal interaction to get someone to reveal security-relevant information and even actions that can lead to an attack. The point of social engineering is to persuade the victim to be helpful. The attacker often impersonates someone occupying a senior position inside the organization that is in some difficulty. The victim provides the necessary assistance without verifying the identity of the caller, thus compromising security.
iii. Reconnaissance: Reconnaissance is a generally used term for collecting information. In security, it often refers to gathering discrete bits of information from various sources and then putting them together to make a coherent whole.

One commonly used reconnaissance technique is "**dumpster diving.**" It involves looking through items that have been discarded in garbage bins or waste paper baskets. One might find network diagrams, printouts of security device configurations, system designs and source code, telephone and employee lists, and more. Even outdated printouts may be useful. Reconnaissance may also involve eavesdropping. The attacker or his accomplice may follow employees to lunch and try to listen to a coworkers discussing security matters.

iv. **Operating System and Application Fingerprinting**: Here the attacker wants to know which commercial server application is running, what version, and what the underlying operating system and version are. While the network protocols are standard and vendor independent, each vendor has implemented the standard independently, so there may be minor variations in interpretation and behaviour. The variations do not make the software noncompliant with the standard, but they are different enough to make each version distinctive. How a system responds to a prompt (for instance, by acknowledging it, requesting retransmission, or ignoring it) can also reveal the system and version. New features also offer a clue. For example, a new version will implement a new feature but an old version will reject the request. All these peculiarities, are called

the operating system or application fingerprint, can mark the manufacturer and version.

v. **Bulletin Boards and Chats**: Underground bulletin boards and chat rooms support exchange of information among the hackers. Attackers can post their latest exploits and techniques, read what others have done, and search for additional information on systems, applications, or sites.

vi. **Documentation:** The vendors themselves sometimes distribute information that is useful to an attacker. For example, resource kits distributed by application vendors to other developers can also gives attackers tools to use in investigating a product that can subsequently be the target of an attack.

Did you know?

According to the statistics released by the *Federal Bureau of Investigation* (FBI):

a. 90% of companies admitted to a security breach in the last 12 months
b. 80% of companies admitted a loss; financial loss and loss of intellectual property are the highest.
c. 78% of companies report abuse of Internet access by insiders.

Communication Subsystem Vulnerabilities

i. **Eavesdropping and Wiretapping**: An attacker can pick the content of a communication passing in unencrypted form. The term eavesdrop implies overhearing without expending any extra effort. For example, an attacker (or a system administrator) is eavesdropping by monitoring all traffic passing through a node. The administrator might have a legitimate purpose, such as watching for inappropriate use of resources. A more hostile term is wiretap, which means intercepting communications through some effort. Passive wiretapping is just "listening," almost like eavesdropping. But active wiretapping means injecting something into the communication stream. A wiretap can be done in such a manner that neither the sender nor the receiver of a communication will know that the contents have been intercepted.

ii. **Microwave signal tapping: Micro**wave signals are broadcast through the air, making them accessible to outsiders. An attacker can intercept a microwave transmission by interfering with the line of sight between sender and receiver. It is also possible to pick up the signal from an antenna located close to the legitimate antenna.

iii. **Satellite Signal Interception**: In satellite communication, the potential for interception is even greater than the microwave signals. However, because

satellite communications are heavily multiplexed, the cost of extracting a single communication is rather high.

iv. **Wireless:** Wireless networking is becoming very popular, but threats arise because of the ability of intruders to intercept and spoof a connection. A wireless signal is strong for approximately 30 to 60 meters. A strong signal can be picked up easily. Another problem is the possibility of unauthorized use of a network connection, or a theft of wireless service.

v. **Optical Fiber**: It is not possible to tap an optical system without detection. Further, optical fiber carries light energy, not electricity, which does not emanate a magnetic field as electricity does. Therefore, an inductive tap is impossible on an optical fiber cable. However, the repeaters, splices, and taps along a cable are places at which data may be intercepted more easily than in the fiber cable itself.

vi. **Protocol Flaws:** Internet protocols are publicly posted for scrutiny. Many problems with protocols have been identified by reviewers and corrected before the protocol was established as a standard. Despite this process of peer review, flaws exist in many of the commonly used protocols, which can be exploited by an attacker.

Did you know?

A zombie computer is a computer attached to the Internet that has been compromised by a hacker, a computer virus, or a Trojan horse. Most owners of zombie computers are unaware that their system is being used in this way. Zombies have been used extensively to send e-mail spam; as of 2005, an estimated 50–80% of all spam worldwide was sent by zombie computers. This allows spammers to avoid detection and presumably to reduce their bandwidth costs, since the owners of zombies pay for their bandwidth.

Impersonation

In many instances, an easy way to obtain information about a network is to impersonate another person or process. An impersonator may foil authentication by any of the following means:

i. **Authentication foiled by guessing: An attacker can g**uess the identity and authentication details of the target, by using common passwords, words in a dictionary, variations of the user name, default passwords, etc.

ii. **Authentication foiled by eavesdropping or wiretapping:** When the account and authentication details are passed on the network without encryption, they are exposed to anyone observing the communication on the network. These authentication details can be reused by an impersonator until they are changed.

iii. **Authentication Foiled by Avoidance:** A flawed operating system may be such that the buffer for typed characters in a password is of fixed size, counting all characters typed, including backspaces for correction. If a user types more characters than the buffer can hold, the overflow causes the operating system to by-pass password comparison and act as if a correct authentication has been supplied. Such flaws or weaknesses can be exploited by anyone seeking unauthorized access.

iv. **Nonexistent Authentication: The** attacker can circumvent or disable the authentication mechanism at the target computer. If two computers trust each other's authentication, an attacker may obtain access to one syste**m** through an authentication weakness (such as a guessed password. and then transfer to another system that accepts the authenticity of a user who comes from a system on its trusted list. The attacker may also use a system that has some identities requiring no authentication. For example, **so**me systems have "guest" or "anonymous" accounts to allow outsiders to access things the systems want to release to the public. These accounts allow access to unauthenticated users.

v. **Well-Known Authentication:** Most vendors often sell computers with one system administration account installed, having a default password. Or the systems come with a demonstration or test account, with no required password. Som**e** administrators fail to change the passwords or delete these accounts, creating vulnerability.

vi. **Spoofing and Masquerading:** Both of them are forms of impersonation. (Refer to chapter on logical access controls for details.)

vii. **Session Hijacking:** Session hijacking is intercepting and carrying on a session begun by another entity. In this case the attacker intercepts the session of one of the two entities that have entered into a session and carry it over in the name of that entity. For example, in an e-commerce transaction, just before a user places his order and gives his address, credit number etc. the session could be hijacked by an attacker.

viii. **Man-in-the-Middle Attack:** A man-in-the-middle attack is a similar to session hijacking, in which one entity intrudes between two others. The difference between man-in-the-middle and hijacking is that a man-in-the-middle usually participates from the start of the session, whereas a session hijacking occurs after the session has been established. The difference is largely semantic and not particularly significant.

Message Confidentiality Threats

An attacker can easily violate message confidentiality (and perhaps integrity) because of the public nature of networks. Eavesdropping and impersonation attacks

can lead to a confidentiality or integrity failure. Here we consider several other vulnerabilities that can affect confidentiality.

i. **Misdelivery:** Message misdelivery happens mainly due to congestion at network elements which makes the buffers overflow and packets get dropped. Sometimes messages are misdelivered because of some flaw in the network hardware or software. Most frequently, messages are lost totally, , which is an integrity or availability issue. Occasionally, however, a destination address will be modified or some router or protocol will malfunction, causing a message to be delivered to someone other than the intended recipient. All of these "random" events are uncommon. More frequent than network flaws are human errors, caused by typing a wrong address.
ii. **Exposure: The** content of a message may be exposed in temporary buffers, at switches, routers, gateways, and intermediate hosts throughout the network; and in the workspaces of processes that build, format, and present the message. A malicious attacker can use any of these exposures as part of a general or focused attack on message confidentiality.
iii. **Traffic Analysis (or Traffic Flow Analysis):** Sometimes not only is the message sensitive but the fact that it exists is also sensitive. For example, if a wartime enemy sees a large amount of network traffic between headquarters and a particular unit, the enemy may be able to infer that significant action is being planned involving that unit. In a commercial setting, messages sent from the president of a company to the president of its competitor could lead to speculation about a takeover or a conspiracy to fix prices.

Message Integrity Threats

In most cases, the integrity or correctness of a communication is more important than its confidentiality. Some of the threats which could compromise integrity are by:

- Changing some part or all of the content of a message
- Replacing a message entirely, including the date, time, and sender/ receiver identification
- Reusing (replaying) an old message
- Combining pieces of different messages into one false message
- Changing the source of a message
- Redirecting a message
- Destroying or deleting a message

These attacks can be perpetrated in the ways already stated, including:

- Active wiretap
- Trojan horse

- Impersonation
- Compromised host or workstation

WebSite Defacement (website)

Website defacement is common not only because of its visibility but also because of the ease with which it can be done. Websites are designed so that their code is downloaded and executed in the client (browser). This enables an attacker to obtain the full hypertext document and all programs and references programs embedded in the browser. He gets the information necessary to attack the website. Most websites have quite a few common and well known vulnerabilities that an attacker can exploit.

Denial of Service

Denial of Service (DoS) attacks lead to loss of network availability. The electronic threats are more serious but less obvious. Some of them are described below:

i. **Connection Flooding:** This is the oldest type of attack where an attacker sends more data than what a communication system can handle, thereby preventing the system from receiving any other legitimate data. Even if an occasional legitimate packet reaches the system, communication is seriously degraded.
ii. **Ping of death: It is** possible to crash, reboot or kill a large number of systems by sending a ping of a certain size from a remote machine. This is a serious problem, mainly because it can be reproduced very easily, and from a remote machine. Ping is an ICMP protocol which requests a destinatio**n to return a r**eply, intended to show that the destination system is reachable and functioning. Since ping requires the recipient to respond to its ping request, all that the attacker needs to do is to send a flood of pings to the intended victim.
iii. **Traffic Redirection:** A router is a device that forwards traffic on its way through intermediate networks between a source host's network and a destination's. So if an attacker can corrupt the routing, traffic will disappear.
iv. **DNS Attacks:** DNS attacks are actually attacks based on the concept of domain name server (DNS), which is a table that converts domain names like www.icai.org into network addresses like 202.54.74.130, a process called resolving the domain name or name resolution. By corrupting a name server or causing it to cache spurious entries, an attacker can redirect the routing of any traffic, or ensure that packets intended for a particular host never reach their destination.

Did you know?

On two occasions to date, attackers have performed DNS Backbone DDoS Attacks on the DNS root servers. Since these machines are intended to provide service to all Internet users, these two attacks might be classified as attempts to take down the entire Internet, though it is unclear what the attackers' true motives were. The first took place in October 2002 and disrupted service at 9 of the 13 root servers, and the second in February 2007, which disrupted and caused disruptions at two of the root servers.

Distributed Denial of Service

In distributed denial of service (DDoS) attack more than one machine is used to attack the target. These multiple machines are called zombies that act on the direction of the attacker, but they don't belong him/her. These machines are vulnerable but can be exploited to attack another machine. The attacker takes advantage of this and uses them to attack the target simultaneously. In addition to their tremendous multiplying effect, they can also cause serious problems, because they are easily launched by using scripts.

Did you know?

There are millions of people constantly on the Internet, but that doesn't mean you have a lower chance of being a target than the next person. There are tools freely dispersed throughout the Internet that let anyone quickly (and easily) scan large amounts of computers for vulnerabilities.

Threats from Cookies, Scripts and Active or Mobile Code

Some of the vulnerabilities relating to data or programs that are downloaded from the server and used by the client are:

i. **Cookies:** Cookies are data files created by the server that can be stored on the client machine and fetched by a remote server usually containing information about the user on the client machine. Anyone intercepting or retrieving a cookie can impersonate the cookie's legitimate owner.

ii. **Scripts:** Clients invoke services by executing scripts on servers. A malicious user can monitor the communication between a browser and a server to see how changing a web page entry affects what the browser sends and then how the server reacts. With this knowledge, the malicious user can manipulate the server's actions. The common scripting languages for web servers, CGI (Common Gateway Interface), and Microsoft's active server pages (ASP) have vulnerabilities that can be exploited by an attacker.

iii. **Active Code:** Active code or mobile code is a general name for a code that is downloaded from the server by the client and executed on the client machine. The popular types of active code languages are Java, JavaScript, VBScript and ActiveX controls. Such an executable code is also called an applet. A hostile applet is a downloadable code that can cause harm to the client's system. Because an applet is not screened for safety when it is downloaded and because it typically runs with the privileges of its invoked user, a hostile applet can cause serious damage.

Did you know?

- Computer pests can potentially stop an organization in its tracks. An infection may cause a loss of computing power. Servers and
- Workstations either slow down or quit responding. In addition, network bandwidth and Internet connections (a primary means of communications with other organizations), may slow so much that essential performance is affected."

Network Security Controls

This section examines the controls available to ensure network security from the various threats identified earlier. The controls are listed under the following broad heads:

- Architecture
- Cryptography/Encryption
- Content Integrity
- Strong Authentication
- Remote Access Security
- Firewalls
- Intrusion Detection Systems

i. Architecture

The architecture or design of a network has a significant effect on its security. Some of the major considerations are:

- **Segmentation / Zoning:** Segmentation / Zoning limit is the potential for harm in a network in two important ways. Segmentation reduces the number of threats, and limits the damage caused by a single vulnerability. A web server, authentication server, applications and database resides on a single server or segment for facilitating electronic commerce transactions are a very insecure configuration. A more secure design uses multiple segments. Since the web server is exposed to public, it should not have other sensitive functions on it or

residing on the segment that has user authentication or access to the database. Separate segments and servers reduce potential harm, should any subsystem be compromised.

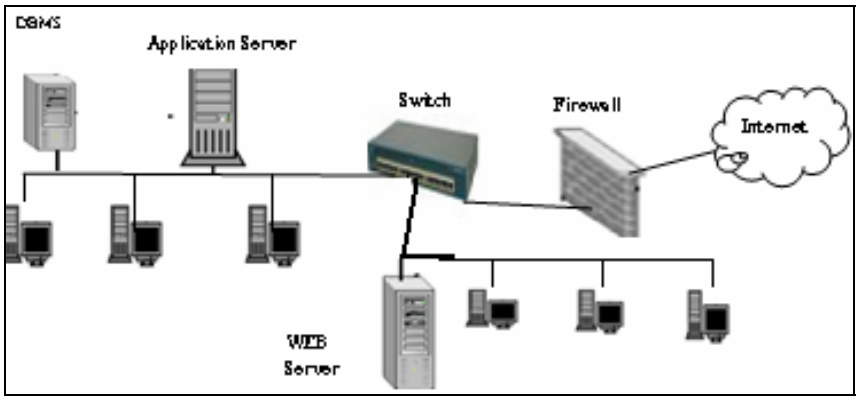


Fig.3.2: Segmented Architecture.

Redundancy: Another key architectural control is redundancy, which is allowing a function to be performed on more than one node. Instead of having a single web server, a better design is to have two servers, using a "failover mode". If one server is used and that server is down for some reason the whole application is not available. In the failover mode, servers communicate with each other periodically, each determining if the other is still active. If one fails, the other takes over processing for both of them. Although performance is cut approximately in half when a failure occurs, some minimum processing is being done which can be used to maintain critical functions.

- **Eliminate Single Points of Failure**: The optimized network architecture ensures its availability by eliminating single points of failure. The failure will also result in compromising the critical components including servers, network devices and communication channels in a network and their availability.
- **Cryptography/Encryption:** The technical details of cryptography have been dealt with in the earlier module. But some applications of cryptography that are relevant to Network security are discussed here.

ii. Link Encryption

In link encryption, data is encrypted just before the system places them on the physical communications link, that is, encryption occurs at the Data Link layer of the OSI model. Correspondingly, decryption occurs at the Data Link layer of the receiving host.

Link encryption protects the message in transit between two computers, but the message is in plaintext inside the hosts (above the data link layer). Headers added by the network layer (which includes addresses, routing information and protocol) and above are encrypted, along with the message/data. The message is, however, exposed at the Network layer and thus all intermediate nodes through which the message passes can read the message. This is because all routing and addressing is done at the Network layer. Link encryption is invisible to users and appropriate when the transmission line is the point of greatest vulnerability in the network. Link encryption provides protection against vulnerabilities that depend on network traffic analysis.

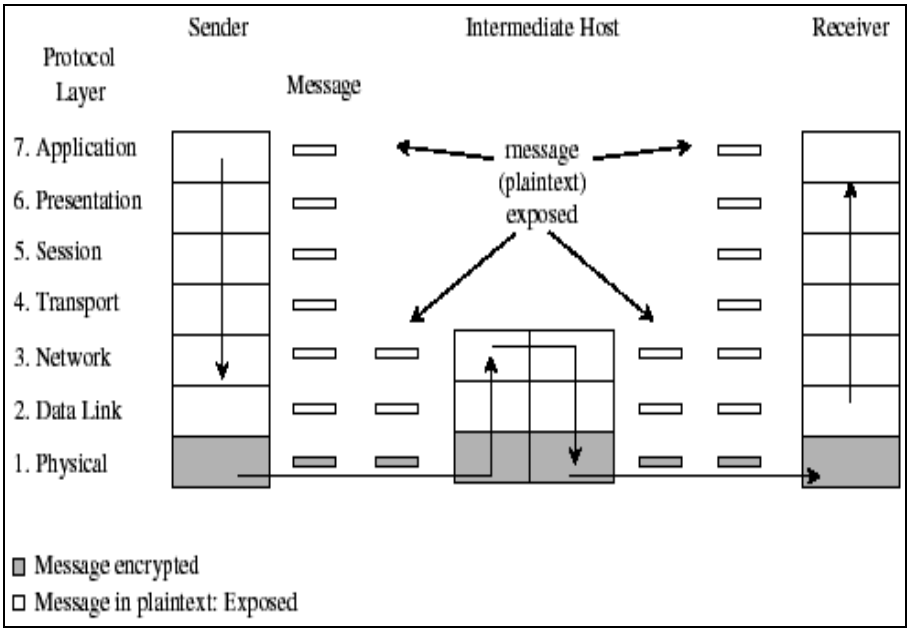


Fig.3.3: Link Encryption

iii. End-to-End Encryption

End-to-end encryption provides security from one end of a transmission to the other. It can be applied by a hardware device between the user and the host or by software running on the host computer. In both cases, encryption is performed at the higher layers, usually application or presentation layer. When end-to-end encryption is used, messages, even when sent through several insecure intermediate hosts, are protected. This is because the data content remains encrypted at all the intermediate layers. However, since the headers below the transport is not encrypted (networks, data link, etc.) end-to-end does not provide protection against traffic analysis. It is

possible use both Link and End-to-end encryption at the same time; one does not preclude the other.

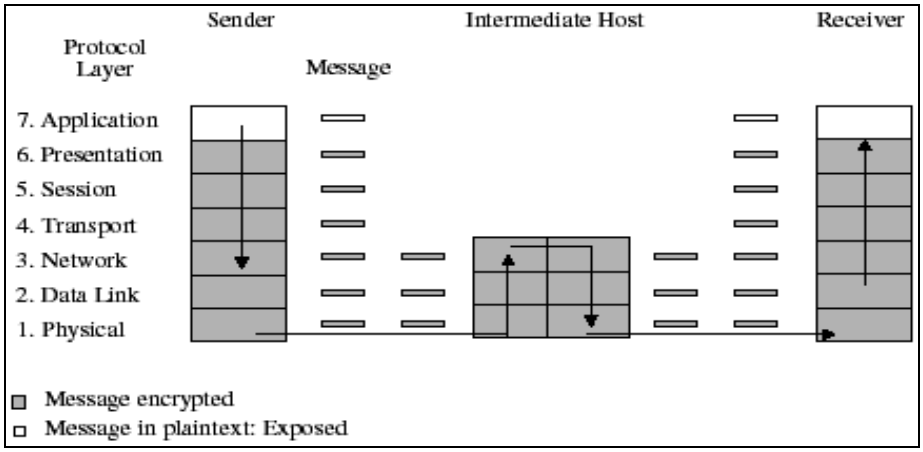


Fig.3.4: End-to-End Encryption

Table: Comparison of Link and End-to-End Encryption.

Link Encryption	End-to-End Encryption
Security within hosts	
Data exposed in sending host	Data encrypted in sending host
Data exposed in intermediate nodes	Data encrypted in intermediate nodes
Role of user	
Applied by sending host	Applied by sending process
Invisible to user	User applies encryption
Host maintains encryption	User must find algorithm
One facility for all users	User selects encryption
Typically done in hardware	Either software or hardware implementation
All or no data encrypted	User chooses to encrypt or not for each data item
Implementation concerns	
Requires one key per host pair	Requires one key per user pair
Provides node authentication	Provides user authentication

iv. PKI and Certificates

A public key infrastructure (PKI) is a process which enables users to implement public key (asymmetric. cryptography, usually in a large and distributed setting. It offers each user a set of services, related to identification and access control:

- Creates certificates associating a user's identity with a (public. cryptographic key
- Issues certificates from its database
- Signs certificates, adding its credibility to the authenticity of the certificate
- Confirms (or denies) the validity of a certificate
- Revokes certificates for users who are no longer allowed access or whose private key has been exposed.

PKI is a set of policies, procedures and products but not a standard. The policies define the rules under which the cryptographic systems operate. In particular, these specify how to handle keys and valuable information and match levels of control to level of risk. The procedures dictate how the keys should be generated, managed, and used. Finally, the products actually implement the policies, and they generate, store, and manage the keys. Entities, called certificate authorities, implement the PKI policy on certificates. The functions of the authority can be done in-house or by a commercial service or a trusted third party. PKI may also involve a registration authority that acts as an interface between a user and a certificate authority. The registration authority captures and authenticates the identity of a user and then submits a certificate request to the appropriate authority.

v. SSL Encryption

The SSL (Secure Sockets Layer) protocol was originally designed by Netscape to protect communication between a web browser and server. It is also known now as TLS, transport layer security. SSL interfaces between applications (such as browsers) and the TCP/IP protocols to provide server authentication, optional client authentication, and an encrypted communications channel between clients and servers.

To create the SSL connection, the client requests an SSL session. The server responds with its public key certificate so that the client can determine its authenticity. The client returns symmetric session key encrypted under the server's public key. The server decrypts the session key and then they switch to encrypted communication, using the shared session key.

vi. IPSec

IETF (Internet Engineering Task Force) has adopted IPSec, or the IP Security Protocol Suite. Designed to address spoofing, eavesdropping, and session hijacking,

the IPSec protocol defines a standard means for handling encrypted data. IPSec is implemented at the IP layer, so it affects all layers above it, in particular TCP and UDP.

IPSec is similar to SSL, in that it supports authentication and confidentiality in a way that does not necessitate significant change either above it (in applications) or below it (in the TCP protocols). Like SSL, it was designed to be independent of specific cryptographic protocols, and to allow the two communicating parties to agree on a mutually supported set of protocols.

Physical Header	IP Header	TCP Header	Data	Physical Trailer

Fig.3.5: Traditional Packets

Physical Header	IP Header	ESP Header (TCP Header + Data).	Physical Trailer

Fig.3.6: IPSec Packet

vii. Signed Code

As already noted, it is possible for someone to place malicious active code on a website to be downloaded by unsuspecting users. A partial solution to reduce this risk is to use signed code. A trustworthy third party appends a digital signature to a piece of code (or macro), supposedly connoting more trustworthy code. A signature structure in a PKI helps to validate the signature. A well-known manufacturer would be recognizable as a code signer.

viii. Encrypted E-Mail

An electronic mail message generally has no privacy at all. The service provider and any intermediate host can read not just the address but also everything in the message field. To protect the privacy of the message and routing information, we need encryption to protect the confidentiality and integrity of the message. The two popular approaches to key management are using a hierarchical, certificate based PKI solution for key exchange and using a flat, individual-to-individual exchange method. The hierarchical method is called S/MIME (Secure Multi Purpose Mail Extensions) and is employed by many commercial mail programs, such as Microsoft Exchange. The individual method is called PGP (Pretty Good Privacy) and is a commercial add-on.

Content Integrity

Content integrity is automatically implied when cryptographic systems are used. Most kinds of malicious threats are addressed by cryptographic systems very effectively. For non-malicious threats to integrity, the controls are Error Correcting codes and Message Digests (Cryptographic Checksums)

i. Error Correcting Codes

Error detection codes detect an error when it has occurred, and error correction codes can actually correct errors without requiring retransmission of the original message. The error code is transmitted along with the original data, so the recipient can re-compute the error code and check whether the received result matches the expected value.

- **Parity Check**: The simplest error detection code is a parity check. An extra bit (the parity bit) is added to an existing group of data bits depending on their sum. With even parity the extra bit is 0 if the sum of the data bits is even and 1 if the sum is odd; that is, the parity bit is set so that the sum of all data bits plus the parity bit is even. Odd parity is the same except that the sum is odd. Parity bits are useful only when the error is in a single bit (called single bit error).
- **Checksum and CRCs**: A checksum is a form of redundancy check that, at its simplest, works by adding up the basic components of a message, usually the bits or bytes, and storing the resulting value. Later, anyone who has the authentic checksum can verify that the message was not corrupted by doing the same operation on the data, and checking the sum. A more sophisticated type of redundancy check is the cyclic redundancy check (CRC. which considers not only the value of each bit/byte but also the order of the values. A cyclic redundancy check (CRC. uses a hash function to produce a checksum which is a small integer from a large block of data, such as network traffic or computer files, in order to detect errors in transmission or duplication. CRCs are calculated before and after transmission or duplication, and compared to confirm that they are the same.
- **Other Codes**: Other kinds of error detection codes, such as hash codes and Hamming codes are used to detect burst errors (several errors occurring contiguously) and multiple bit errors (multiple errors among non-adjacent bits). Some of the more complex codes (like Hamming codes) can detect multiple-bit errors and may be able to pinpoint which bits have been changed, thus allowing the data to be corrected.

ii. Message Digests (Cryptographic Checksums)

Checksums and CRCs are useful in detecting accidental modifications such as

corruption to stored data or errors in a communication channel. However, they provide no security against malicious agents, as their simple mathematical structure makes them trivial to circumvent. To protect against malicious changes, cryptographic checksum are used. A cryptographic checksum is created by performing a complicated series of mathematical operations (the cryptographic algorithm) that translates the data in the file into a fixed string of digits called a hash value, which is then used as a checksum. Without knowing which cryptographic algorithm was used to create the hash value, it is highly unlikely that an unauthorized person would be able to change data without inadvertently changing the corresponding checksum.

A cryptographic hash function must ensure that the following is computationally not feasible to :

- Determining the content of a message from its Cryptographic Checksums.
- Finding "collisions", wherein two different messages have the same Cryptographic Checksums.

Cryptographic checksums are also known as message digests, message authentication codes, integrity check-values, modification detection codes, or message integrity codes.

Strong Authentication

A security policy specifies who, that is, individuals, groups, subjects who can access which resource and objects. Crucial to the policy is authentication: knowing and being assured of the authenticity of identities. The authentication methods appropriate for use in networks are one-time passwords, Challenge Response systems and Kerberos.

i. **One Time Passwords:** A one-time password can guard against wiretapping and spoofing of a remote host. In the simplest case, the user and host both have access to identical lists of passwords. The user would enter the first password for the first login, the next one for the next login, and so forth. As long as the password lists remain secret and as long as no one can guess one password from another, a password obtained through wiretapping can be useless. A more complex but practical implementation uses a password token, a device that generates a password that is unpredictable but that can be validated on the receiving end. The simplest form of password token is a synchronous one. This device displays a random number, generating a new number every minute. Each user is issued a different device (that generates a different key sequence). The user reads the number from the device's display and types it in as a one time password. The computer on the receiving end executes the algorithm to

generate the password appropriate for the current minute; if the user's password matches the one computed remotely, the user is authenticated.

ii. **Challenge Response Systems:** A challenge and response device looks like a pocket calculator. The user first authenticates the device, usually by means of a PIN. The remote system sends a random number, called the "challenge" which the user enters into the device. The device responds to the challenge with another number, which the user then transmits to the system.

The system prompts the user with a new challenge for each use. Thus, this device eliminates the small window of vulnerability in which a user could reuse a time sensitive authenticator. A generator that falls into the wrong hands is useless without the PIN.

iii. **Kerberos:** is a system that supports authentication in distributed systems. Originally designed to work with secret key encryption, Kerberos, in its latest version, uses public key technology to support key exchange. It is used for authentication between intelligent processes, such as client-to-server tasks, or a user's workstation to other hosts. It is based on the idea that a central server provides authenticated tokens, called tickets, to requesting applications. A ticket is an un-forgeable, non-repayable, authenticated object. That is, it is an encrypted data structure naming a user and a service that user is allowed to obtain. It also contains a time value and some control information.

Remote Access Security

Remote access technologies can be defined as data networking technologies that are focused on providing the remote user with access into a network, while striving to maintain the principal tenets of confidentiality, availability, and integrity.

The advantages of employing secure remote network access are: Reducing networking costs by using the Internet to replace expensive dedicated network lines

- Providing employees with flexible work styles such as telecommuting
- Building very efficient ties with customers, suppliers, and employees

i. Virtual Private Networking (VPN)

A virtual private network (VPN) is created by building a secure communications link between two nodes by emulating the properties of a point-to-point private link. It can be used to facilitate secure remote access into a network, securely connect two networks together, or create a secure data tunnel within a network. Encryption coupled with access controls (including firewalls) can provide users with the same level of privacy that can be provided on a private network, even when the

communication traverses a part of the public network. (For more details, see the previous module).

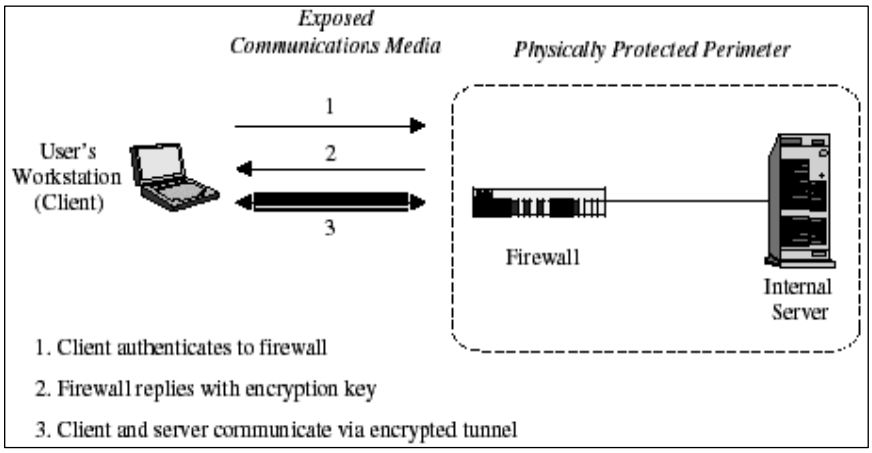


Fig.3.7: Secure VPN

ii. Dial back procedures

In a networked computing environment, users may often require access to the systems resources from remote locations. Dial-back systems are a control to ensure that access is made only from authorised lines or locations. When a user dials into the server and identifies himself. The server hangs up and calls the user at a pre-determined telephone number and then enables the user to access the resources based on password authentication. A weakness in this procedure is call-forwarding. An unauthorized person could enable calls to a pre-determined number to be forwarded to the number designated by him, thus enabling him to gain unauthorized access to the resources.

iii. Other controls

To minimize the risk of unauthorized dial-in access, remote users should never store their passwords in plain text login scripts on notebooks and laptops.

iv. Authentication Servers

In widely spread out networked systems, the problem of user management and authorized access is crucial since users are spread over a wide geographical area. In such cases all access control is transferred to a centralized or decentralized access authentication mechanism. Two of the popular applications of remote authentication mechanisms depending on centralized/decentralized access authentication implementations are TACACS (Terminal Access Controller Access Control System)

and RADIUS (Remote Authentication Dial in User Service). Some of the features of such systems are:

- Enable secure remote access
- Facilitate centralized user management
- Facilitate centralized access monitoring and control
- Change to user access rights made easy
- Provide event logging and extended audit trails

Firewalls

The technical details of firewalls, their types and configurations have been dealt with in the first module. Only specialized applications of firewalls for network security are dealt with here.

i. Virtual Private Networks

Firewalls and firewall environments are used to construct Virtual Private Networks (VPNs). (See the earlier module for more details.)

ii. Intranet

An intranet is a network that employs the same type of services, applications, and protocols present in an Internet implementation, without involving external connectivity. For example, an enterprise network employing the TCP/IP protocol suite, along with HTTP for information dissemination would be considered an Intranet.

Most organizations currently employ some type of intranet, although they may not refer to the network as such. Within the internal network (intranet), many smaller intranets can be created by the using internal firewalls. For example, an organization may protect its personnel network with an internal firewall, and the resultant protected network may be referred to as the personnel intranet.

Since intranets utilize the same protocols and application services that are present on the Internet, many security issues inherent in Internet implementations get bound to them in intranet implementations. Therefore, intranets are typically implemented behind firewall environments.

iii. Extranets

An extranet is usually a business-to-business intranet; that is, two intranets are joined via the Internet. The extranet allows limited, controlled access to remote users via some form of authentication and encryption such as the one provided by a VPN. Extranets share nearly all of the characteristics of intranets, except that extranets are designed to exist outside a firewall environment. By definition, the purpose of an extranet is to provide access to potentially sensitive information to specific remote users or organizations, but at the same time denying access to general external

users and systems. Extranets employ TCP/IP protocols, along with the same standard applications and services. Many organizations and agencies currently employ extranets to communicate with clients and customers. Within an extranet, options are available to enforce varying degrees of authentication, logging, and encryption.

iv. Securing a Firewall

Firewall platforms should be implemented on systems containing operating system builds that have been stripped down and hardened for security applications. Firewalls should never be placed on systems built with all possible installation options.

Firewall operating system builds should be based upon minimal feature sets. All unnecessary operating system features should be removed from the build prior to firewall implementation. All appropriate operating system patches should be applied before any installation of its components.

The operating system build should not rely strictly on modifications made by the firewall installation process. Firewall installation programs rely on the lowest common denominator approach; extraneous software packages or modules might not be removed or disabled during the installation process.

The hardening procedure used during installation should be tailored to the specific operating system undergoing hardening. Some often-overlooked issues include the following:

- Any unused networking protocols should be removed from the firewall operating system build. Unused networking protocols can potentially be used to bypass or damage the firewall environment. Finally, disabling unused protocols ensures that attacks on the firewall utilizing protocol encapsulation techniques will not be effective.
- Any unused network services or applications should be removed or disabled. Unused applications are often used to attack firewalls because many administrators neglect to implement default-restrictive firewall access controls. In addition, these services and applications are likely to run by using default configurations, which are usually much less secure than production-ready application or service configurations.
- Any unused user or system accounts should be removed or disabled. This however is operating system specific, since all operating systems vary in terms of which accounts are present by default as well as how accounts can be removed or disabled.
- Applying all relevant operating system patches is also critical. Since patches and hot fixes are normally released to address security-related issues, they should be

integrated into the firewall build process. Patches should always be tested on a non-production system prior to rollout to any production systems.

- Unused physical network interfaces should be disabled or removed from the server chassis.

Intrusion Detection Systems

After the perimeter controls, firewall, and authentication and access controls block certain actions, some users are admitted to use a computing system. Most of these controls are preventive, that is, they prevent known undesirable things from happening. Many studies, however, have shown that most computer security incidents are caused by insiders, people who would not be blocked by a firewall. And insiders require access with significant privileges to do their daily jobs.

Intrusion detection systems complement these preventive controls as the next line of defence. An intrusion detection system (IDS) is a device, usually another separate computer, which monitors activity to identify malicious or suspicious events. An IDS is a sensor that raises an alarm if specific things occur. The alarm can range from writing an entry in an audit log to something significant, such as messaging an alert to the system security administrator. An IDS receives inputs from sensors. It saves those inputs, analyzes them, and takes some controlling action. The functions performed by IDS are:

- Monitoring users and system activity
- Auditing system configuration for vulnerabilities and mis-configurations
- Assessing the integrity of critical system and data files
- Recognizing known attack patterns in system activity
- Identifying abnormal activity through statistical analysis
- Managing audit trails and highlighting user violation of policy or normal activity
- Correcting system configuration errors
- Installing and operating traps to record information about intruders
- Offering special considerations for auditing remote access and network security

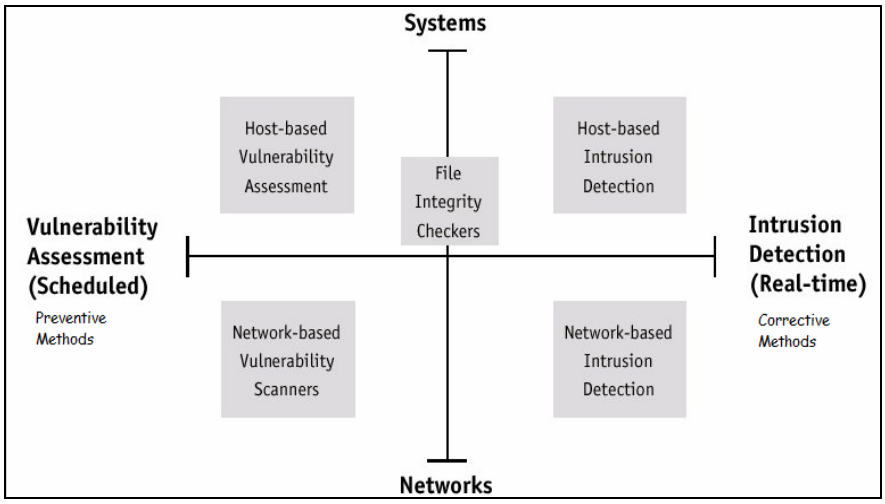


Fig.3.8: Intrusion Detection System

Many intrusion detection systems are also capable of interacting with firewalls and thus bring a reactive element to the provision of network security services. Firewalls that interact with intrusion detection systems are capable of responding to perceived remote threats automatically, and without the delays associated with a human response. For example, if an intrusion detection system detects a denial of service attack in progress, it can instruct certain firewalls to automatically block the source of the attack (although, false positives responses can occur).

The two general types of intrusion detection systems are signature based and heuristic. Signature-based intrusion detection systems perform simple pattern-matching and report situations that match a pattern corresponding to a known attack type. Heuristic intrusion detection systems, also known as anomaly based, build a model of acceptable behaviour and flag exceptions to that model; for the future, the administrator can mark a flagged behaviour as acceptable so that the heuristic IDS will now treat that previously unclassified behaviour as acceptable.

Intrusion detection devices can be network-based or host-based. A network-based IDS is a stand-alone device attached to the network to monitor traffic throughout that network; a host-based IDS runs on a single workstation or client or host, to protect that one host. For more details, please see the previous module.

Auditing Network Security

Auditing networked computing environments presents significant complexities. Networking enables several virtual machines to operate together using a limited set of systems resources, irrespective of the barriers of geographic location of the user and

systems infrastructure. For example, a customer can now access his bank account from anywhere in the world. This means that logical paths open up enabling access through insecure networks and diverse computing infrastructures.

Audit of network security requires the auditor to take special considerations into account and plan accordingly to achieve his audit objectives. These are:

- Locating logical access paths by reviewing network diagrams
- Identifying network topologies, virtual paths spanning across LANs, WANs and the open networks such as shared networks and the Internet.
- Recognizing logical access threats, risks and exposures in the networked environment.
- Identifying and controlling access paths used for distributed processing and distributed databases.
- Evaluating network management and change control with respect to technical components such as modems, switches, routers, firewalls, VPNs, network management and access control software, encryption, protocols, middleware controls and Internet security.
- Identifying information resource owners can be quite complex since in a distributed computing environment, an application process can span several systems and networks, including those outside the organisation's control.
- Evaluating logical network security policies and practices.

Evaluating effectiveness of logical access security with respect to network security components, such as:

- **Firewalls and filtering routers**: architecture, configuration setting as per firewall security policy, port services, anti-virus configuration, reporting and management controls
- **Intrusion detection systems**: architecture, configuration, interface with other security applications, reporting and management controls.
- **Virtual private networks:** architecture, devices, protocol, encryption process integration with firewall security, change management.
- **Security protocols:** selection of appropriate protocol, seamless security integration of protocols between devices running different protocols.
- **Encryption:** selection of appropriate encryption methods to various application processes
- **Middleware controls:** middleware design and access control with respect to identification, authentication and authorization, management of components and middleware change management.
- Network event logging and monitoring

Type of System		Intrusion Detection				Vulnerability		
System Control Features		Monitoring				Assessment		
Controls D- Detective P-Preventive C- Corrective S-Support		Application Based	Host Based	Target Based	Network Based	Host Based	Network Based	Password Assessment
Confidentiality	Unauthorized access to files and system resources		D			P	P	P
	Modification to files		D	D		P	P	P
	Violation of enterprise system access polices	D	D			P	P	
	Violation of security policies	D	D	D	D	P	P	
	Weak or non-existent passwords	D	D			D		D
Integrity	Placement of Trojan horse or malicious software		D	D		P	P	
	Presence of Trojan horse or malicious software			D				
	Attack Against network services				D		P	
	Script based attacks	D			D	P		
Availability	Denial of Services Attacks				D		P	
	Failure or Mis-configuration of firewalls	D			D	P	P	
	Attacks Happening Over Encrypted Links	D	D					
	Unusual activity or variation from normal data pattern		D		D			
Other	Errors in Network configuration		D			DPC	DPC	
	Liability Exposure associated with attacker using own resources to attack others	P	P	P	P	P	P	P
	Post incident damage assessment	S	S	S	S	S	S	S

Penetration Testing

Adequately protecting an organization's information assets is a business imperative, one that requires a comprehensive, structured approach. The purpose of this section is to explore an ethical hacking technique referred to in the IT community as Penetration Testing, which is being used increasingly by organizations to evaluate the effectiveness of information security measures.

As its name implies, penetration testing includes a series of activities undertaken to identify and exploit security vulnerabilities. The idea is to find out how easy or difficult it might be for someone to "penetrate" an organization's security controls or to gain unauthorized access to its information systems.

A penetration test typically involves a small team of people sponsored by the organization asking for the test. The team attempts to exploit vulnerabilities in the organization's information security by simulating an unauthorized user (or "hacker") attacking the system by using similar tools and techniques. Penetration testing teams typically comprise people from an organization's Internal Audit department or IT department, or from consulting firms specializing in these services. Their goal is to attempt to identify security vulnerabilities under controlled circumstances, so that they can be eliminated before unauthorized users can exploit them. Because penetration testing is an authorized attempt to simulate hacker activities, it is often referred to as "ethical hacking."

It is important to point out that a penetration test cannot be expected to identify all possible security vulnerabilities, nor does it offer any guarantee that an organization's information is secure. Penetration testing is typically conducted at a point in time. New technology, new hacker tools and changes in an organization's information system can create exposures not anticipated during the penetration testing. In addition, penetration testing is normally completed with finite resources, focused on a particular area, over a finite period of time. Hackers determined to break into an organization's information systems are often not bound by similar constraints. Penetration testing is also typically focused on a system's security vulnerabilities that can cause unauthorized access. It is not necessarily focused on security vulnerabilities that could result in the accidental loss or disclosure of the organization's information and information systems.

Many organizations have deployed sophisticated security mechanisms, such as firewalls or intrusion detection systems (IDS), to help protect their information assets and to quickly identify potential attacks. While these mechanisms are important, they are not foolproof. A firewall cannot protect against what is allowed through – such as online applications and allowed services. While an IDS can detect potential

intrusions, it can detect only what it has been programmed to identify, and it will not be effective at all if the company does not monitor or respond to the alerts. Also, firewalls and intrusion detection systems have to be continuously updated or they lose their effectiveness to prevent or detect attacks. Penetration testing can help validate and confirm the effective configuration of an organization's firewalls and its intrusion detection systems.

Penetration Testing Scope

The scope of a penetration testing project is subject to negotiation between the sponsor of the project and the testing team, and will vary, depending on the objectives to be achieved. The principal objective of penetration testing is to determine whether an organization's security vulnerabilities can be exploited and its systems compromised.

Conducting such a test involves gathering information about an organization's information systems and information security and then using it to identify and exploit known or potential security vulnerabilities. Penetration testing team's ability to exploit security vulnerabilities can vary from gathering "computer screen shots" to copying sensitive information or files to being able to create new user accounts on the system or being able to create and/or delete particular files on the organization's servers.

Penetration testing can have a number of secondary objectives, including testing the security incident identification and response capability of the organization, testing employee security awareness or testing users' compliance with security policies.

Penetration Testing Strategies

Various strategies for penetration testing, based on the specific objectives to be achieved, include:

i. **External vs. internal Testing**: External testing refers to attacks on the organization's network perimeter using procedures performed from outside its systems, that is, from the Internet or Extranet. To conduct the test, the testing team begins by targeting the company's externally visible servers or devices, such as the Domain Name Server (DNS), email server, web server or firewall.

   Internal testing is performed from within the organization's technology environment. The focus is to understand what could happen if the network perimeter were successfully penetrated or what an authorized user could do to penetrate specific information resources within the organization's network.

ii. **Blind testing**: In this strategy, the testing team is provided with only limited information concerning the organization's information systems configuration. The penetration team uses publicly available information (such as company web-site

and domain name registry, Internet discussion board. to gather information about the target and conduct its penetration tests. Blind testing can provide information about the organization that may have been otherwise unknown, but it can also be more time consuming and expensive than other types of penetration testing (such as targeted testing) because of the effort required to research the target.

iii. **Double-blind testing:** extends the blind testing strategy in that the IT and security staff of the organization are not informed beforehand about the planned testing activities, and are thus "blind" to them. Double-blind testing can test the security monitoring and incident identification of the organization, escalation and response procedures. It requires careful monitoring by the project sponsor to ensure that the testing procedures and the organization's incident response procedures are terminated when the objectives of the test have been achieved.

iv. **Targeted testing:** (often referred to as the "lights-turned-on" approach) involves both the organization's IT team and the penetration testing team who are aware of the testing activities and provided with information concerning the target and the network design. This approach is more efficient and cost-effective when the objective of the test is focused more on the technical setting, or on the design of the network, than on the organization's incident response and other operational procedures. A targeted test takes less time and effort to complete than blind testing, but may not provide as complete a picture of the security vulnerabilities and response capabilities of an organization.

Types of Penetration Testing

In addition to the penetration testing strategies, consideration should be given to the types of testing the testing team is to carry out. These could include:

i. **Application security testing**: Many organizations offer access to core business functionality through web-based applications. This type of access introduces new security vulnerabilities, because even with a firewall and other monitoring systems, security can be compromised, since traffic must be allowed to pass through the firewall. The objective of this testing is to evaluate the controls over the application and its process flow. Areas of evaluation may include the application's usage of encryption to protect the confidentiality and integrity of information, the mode of authenticating users, integrity of the Internet user's session with the host application, and use of cookies – a block of data stored on a customer's computer that is used by the web server application.

ii. **Denial of Service (DoS) testing**: The goal of DoS testing is to evaluate the system's susceptibility to attacks that renders it inoperable and it denies legitimate access attempts. Decisions regarding the extent of Denial of Service testing to be incorporated into a penetration testing exercise will depend on the

relative importance of ongoing, continued availability of the information systems and related processing activities.

iii. **War Dialing:** War dialing is a technique for systematically calling a range of telephone numbers in an attempt to identify modems, remote access devices and maintenance connections of computers that may exist on an organization's network. Well-meaning users can inadvertently expose the organization to significant vulnerability by connecting a modem to the organization's information systems. Once a modem or other access device has been identified, analysis and exploitation techniques are performed to assess whether this connection can be used to penetrate the organization's information systems network.

iv. **Wireless network penetration testing**: The introduction of wireless networks, whether through formal, approved network configuration management or the inadvertent actions of well-meaning users, introduces additional security exposures. Sometimes referred to as "war-driving," hackers have become proficient in identifying wireless networks simply by "driving" or walking around office buildings with their wireless network equipment. The goal of wireless network testing is to identify security gaps or flaws in the design, implementation or operation of the organization's wireless network.

v. **Social Engineering**: Often used in conjunction with blind and double blind testing, this refers to techniques using social interaction, typically with the organization's employees, suppliers and contractors, to gather information and penetrate the organization's systems. Such techniques include:

- Posing as a representative of the IT department's help desk and asking users to divulge their user account and password information;
- Posing as an employee and gaining physical access to restricted areas that may house sensitive information;
- Intercepting mail, courier packages or even trash to search for sensitive information on printed materials.

Social engineering activities can test a less technical, but equally important, security component: the ability of the organization to contribute to, or prevent, unauthorized access to information and information systems.

Risks associated with Penetration Testing

Though management sponsors testing activities for security reasons, such activities, in themselves, carry some element of risk. Some of the key risks are:

- The penetration test team may fail to identify significant vulnerabilities;
- Misunderstandings and miscommunications may result in the test objectives not being achieved;

- Testing activities may inadvertently trigger events or responses that may not have been anticipated or planned, ,such as notifying law enforcement authorities;
- Sensitive security information may be disclosed, increasing chances of external attacks to the organization.

During the course of penetration testing, significant security vulnerabilities can, and are likely to be, identified. Such information must be adequately protected, so that it does not fall into wrong hands.

Some questions to consider include:

- Will activities be conducted over the Internet or any other public network? If so, how is information protected while in transmission over such networks?
- How and where will the collected information, including working paper files, be stored? In electronic form? In physical form? Who has, or will have, custody of this information, including summaries of findings and observations?
- How much information will the final reports and executive summaries contain? How will the content and distribution of findings, observations and reports be controlled?
- How will notes, working papers and other forms of information be retained or destroyed?
- Do the terms of engagement include appropriate provisions to protect the confidentiality of the information collected, as well as the findings, observations and recommendations?

The activities or events that will trigger the conclusion of the penetration testing activities should be clearly described. These would, of course, depend on the specific objectives of the test, but could, for example, include collecting proof of the team's ability to exploit security vulnerabilities. This "proof" could take many forms, such as copying a target file, creating a file on a target server, adding a new user to a target system or capturing "screen shots" of a target application system. In some instances, it may be appropriate to define a time period within which the testing is to be completed.

Table: Network Vulnerabilities and Controls.

Target	Vulnerability	Control
Precursors to attack	Port scan	Firewall Intrusion detection system Running as few services as possible Services that reply with only what is necessary

	Social engineering	Education, user awareness Policies and procedures Systems in which two people must agree to perform certain security-critical functions
	Reconnaissance	Firewall Hardened"(self-defensive) operating system and applications Intrusion detection system
	OS and application fingerprinting	Firewall "Hardened" (self-defensive) applications Programs that reply with only what is necessary Intrusion detection system
Authentication failures	Impersonation	Strong, one-time authentication
	Guessing	Strong, one-time authentication Education, user awareness
	Eavesdropping	Strong, one-time authentication Encrypted authentication channel
	Spoofing	Strong, one-time authentication
	Session hijacking	Strong, one-time authentication Encrypted authentication channel Virtual private network
	Man-in-the-middle attack	Strong, one-time authentication Virtual private network Protocol analysis
Programming flaws	Buffer overflow	Programming controls Intrusion detection system Controlled execution environment Personal firewall
	Addressing errors	Programming controls Intrusion detection system Controlled execution environment

		Personal firewall Two-way authentication
	Parameter modification, time-of-check to time-of-use errors	Programming controls Intrusion detection system Controlled execution environment Personal firewall Two-way authentication
	Server-side include	Programming controls Personal firewall Controlled execution environment Intrusion detection system
	Cookie	Firewall Intrusion detection system Controlled execution environment Personal firewall
	Malicious active code: JavaScript, ActiveX	Intrusion detection system Controlled execution environment Signed code
	Malicious code:virus, worm, Trojan horse	Intrusion detection system Signed code Controlled execution environment Intrusion detection system
	Malicious typed code	Signed code Intrusion detection system Controlled execution environment
Confidentiality	Protocol flaw	Programming controls Controlled execution environment
	Eavesdropping	Encryption
	Passive wiretap	Encryption
	Misdelivery	Encryption
	Exposure within the network	End-to-end encryption

	Traffic flow analysis	Encryption Traffic padding Onion routing
	Cookie	Firewall Intrusion detection system Controlled execution environment
Integrity	Protocol flaw	Firewall Controlled execution environment Intrusion detection system Protocol analysis Audit
	Active wiretap	Encryption Error detection code Audit
	Impersonation	Firewall Strong, one-time authentication Encryption Error detection code Audit
	Falsification of message	Firewall Encryption Strong authentication Error detection code Audit
	Noise	Error detection code
	Website defacement	Error detection code Intrusion detection system Controlled execution environment Hardened host Honey pot Audit
	DNS attack	Firewall Intrusion detection system

		Strong authentication for DNS changes Audit
Availability	Protocol flaw	Firewall Redundant architecture
	Transmission or component failure	Architecture
	Connection flooding, e.g., echo-charges, , ping of death, smurf, syn flood	Firewall Intrusion detection system ACL on border router Honey pot
	DNS attack	Firewall Intrusion detection system ACL on border router Honey pot
	Traffic redirection	Encryption Audit
	Distributed denial of service	Firewall Intrusion detection system ACL on border router Honey pot

The layers of security controls on the network are depicted in the following table.

Security Level	**Applicable Security/Control measures**
Perimeter	Firewall Network-based anti-virus VPN encryption
Network	Intrusion detection /prevention system (IDS/IPS) Vulnerability management system Network access control Access control /user authentication
Host	Host IDS Host vulnerability assessment (VA.) Network access control

	Anti-virus Access control/user authentication
Application	Application shield Access control/user authentication Input validation
Data	Encryption Access control/user authentication

Table: Security Layers

Network Infrastructure Auditing Checklist

The following is a general illustrative checklist for the audit of Network infrastructure.

Network Server

- Obtain or prepare logical and physical diagrams of the network and attached local and wide area networks, including the systems' vendor and model description, physical location, and applications and data residing and processing on the servers and workstations.
- Using the information obtained in the prior steps, document the server and directory location of the significant application programs and data within the network; document the flow of transactions between systems and nodes in the network.
- Assess whether the trusted domains are under the same physical and administrative control and are logically located within the same sub-network.
- Determine that router filtering is being used to prevent external network nodes from spoofing the IP address of a trusted domain.
- Determine that the Administrator/super-user and Guest accounts have passwords assigned to them (by attempting to log on without providing a password.. Also ascertain that the Administrator account password is well controlled and used/known by only the system administrator and one backup person.
- Review the account properties settings active in each user's individual profile, which may override the global account policy.
- List the security permissions for all system directories and significant application programs and directories and ensure that they are consistent with security policy.
- Review and assess permissions assigned to groups and individual accounts, noting that Full Control (all permissions) and Change (Read,Write, Execute, and Delete) permissions are restricted to authorized users.
- Review the audit log for suspicious events and follow up on these events with the security administrator.

Router

- Determine the types of accounts that were used to access the routers.
- Determine what users had access to these accounts.
- Were access attempts to the routers logged?
- Determine if all accounts had passwords and also determine the strength of the passwords.
- Was simple network management protocol (SNMP) used to configure the network?
- Determine the version of SNMP employed by the Company. (Version one stores passwords in clear-text format. Version two adds encryption of passwords.)
- Determine if open shortest path first (OSPF) was defined on the router. Determine the authentication mechanism that was employed in the Company's implementation of OSPF.
- Determine whether directed broadcast functionality was enabled on the router. This setting, if enabled, could allow a denial-of-service (DoS) attack of the network (Smurf attack).
- Obtain population of routers with modems and obtain the telephone numbers of the routers.
- Determine if users were properly authenticated when remotely accessing the routers.
- Determine how changes to the router environment were made.
- Were there procedures for changing router configurations? If so, were these procedures well-documented and consistent with security policy?
- Determine if changes to the router configuration were documented.
- Was there a separation of duties within the change control of the router environment?

Firewalls

- Obtain background information about the firewall(s), in place, e.g., segment diagrams, software, hardware, routers, version levels, host names, IP addresses, connections, any specific policies for an overview of the firewall security.
- Determine that the firewall components, both logical and physical, agree with the firewall strategy.
- Determine whether the firewall components are the latest possible version and security patches are current.
- Determine that the root cannot telnet to the system.
- Ensure that the telnet OS banner and other banners such as FTP banner, etc. have been eliminated.
- Ensure that there are no compilers/interpreters on the firewall.

- Ensure that a lockdown rule has been placed at the beginning of the rule base. The lockdown rule protects the firewall, ensuring that whatever other rules are put in later, it will not inadvertently compromise the firewall.
- Obtain and review the connections table for time out limits and number of connections.
- Attempt to test the rule base by scanning secured network segments from other network segments.
- Identify accessible resources behind the firewall that are to be encrypted and determine that the connections are encrypted.
- Determine if there is a change control process in place for the rule base.
- Determine the use of the firewall's automatic notification/alerting features and archiving the detail intruder information to a database for future analysis.

Summary

An Information Systems Auditor's understanding of network security helps to check the adequacy of controls implemented in a distributed enterprise environment. A thorough understanding of the techniques of controls, such as cryptography, security protocols, firewalls and Intrusion Detection systems, facilitates an Auditor in recommending and questioning control features. The penetration testing technique is an important tool to justify his audit evidence.

Some examples of Application Control Techniques and their Suggested Audit Procedures.

Control Activity	Control Techniques	Audit Procedures
Control Connectivity to system resources	Connectivity, including access paths and control technologies between systems and their resources is documented and approved by the management with consistent risk identification. Using technological controls like routers, firewalls, the network access paths in the intranet and internet are to be adequately protected.	Review the approval by management and risk assessment done on access paths of network segments, interface policies along with technical documentation. Identify the key devices and controls on information flow, configuration settings and preventive policies of the network.

	Security policies and controls are to be implemented to authenticate access through specific network devices.	Perform intrusion detection testing and penetration testing to assess the controls within the network and external access to network resources.
	Remote-access using dial-up, internet, VPN, or wireless access methods are to be controlled.	Perform external access tests to resources like dial-up m LAN,WAN and access to internet, VPN, etc.
	Control network session to monitor a period of inactivity using session lock. Implementing effective identification and authentication procedures for users.	Interview network administrator and inspect compliance with standard security mechanisms followed for security on the enterprise network.
	Suitable logging notification with warning banners displaying penalties for unauthorized use.	Check if the system has an automatic session lock feature after a specific period of inactivity.
	Logging of previous successful and unsuccessful logons with date and time and duration of access.	Interview the network officials to ensure appropriate security procedure execution.

Master Checklist for Network Administration and Security Auditing

The following is a general checklist for the audit of Network Administration and Security.

S. No	Checklist
	Process
1.	Is there an Information Security guidelines document, which defines the minimum configuration for any device/link on the organisation's network, including levels of encryption?

2.	Are all platforms/links/devices in compliance with the guidelines? If not, has the appropriate level of management reviewed the non-compliant parts of the network to ensure that the risk levels are acceptable?
3.	For all items supported by external vendors, does the vendor or the manufacturer verify that all cryptographic functions in use by the product/service, such as encryption, message authentication or digital signatures, use approved cryptographic algorithms and key lengths.
4.	Wherever applicable, whether background and reference checks for both internal and outsourced vendor staff who perform security-related functions for the product/service under review are carried out.
5.	This includes job applicants who have accepted a job offer, temporaries, consultants, full time staff as well as the outsourced vendor who is involved in product/service management and operations.
Authentication	
6.	Does the product/service authenticate (verify) the identity of users (or remote systems) prior to initiating a session or transaction? Have these authentication mechanisms been approved by the organization's IT Department? (These include Passwords, Personal Identification Numbers (PINs), (static and dynamic., public keys and biometrics.
7.	Does the organization verify that the initial authentication has used a mechanism that is acceptable for the application? Has the approach been approved by IT Department and have the compensating controls been implemented?
8.	Does the organisation have a comprehensive password construction, implementation and management policy?
9.	Do the Products/Services utilizing biometrics authentication use only biometrics for local authentication?
Public Key Infrastructure (PKI)	
10.	Do the Products/services using Public key (or asymmetric. cryptography for authentication either on a session basis (peer authentication) or on a per-message/transaction basis (digital signatures) use approved security protocols to comply with the public key technology standard?
11.	For products/services that use PKI, private keys which are stored in hardware or software must be protected via an approved mechanism. The

	protection mechanism includes user authentication to enable access to the private key. Are these protection mechanisms adequate?
12.	For products/services that use PKI, an approved process for verifying the binding of a user identity to the public key (e.g., digital certificate) is required for any server relying on public key authentication. Is such a process in place?
Access Control	
13.	Is the access to highly privileged IDs (e.g., system administration access) strictly controlled, audited and limited in its use?
14.	Does the product/service support the need to perform a periodic entitlement review? A periodic entitlement review process should validate access privileges.
15.	Does the product/service support the requirement to limit individual user sessions to a maximum of X minutes of inactivity using either session time out or a password protected screen saver?
16.	Is there a process in place to ensure that access rights reflect changes in employee or job status within X hours of the change? This includes physical access tokens and dial-in capabilities as well as any systems or applications.
17.	For any products/services, which has been outsourced, is there a process in place to ensure that all platforms, services and applications are configured to meet the organisation's Information Security Standards?
18.	Does the product/service display the (a. date and time of last successful login and (b. the number of unsuccessful login attempts since the last successful login?
19.	Does the product/service support a periodic process to ensure that all user IDs for employees, consultants, agents, auditors, or vendors are disabled after X days and deleted after Y days from the day they were not used unless explicitly approved by the concerned business manager.
Cryptography	
20.	Is there a cryptography/encryption policy for various types of classified information that travels/gets stored within and outside the organization's network(s)?

Network Information Security	
21.	Is the approved Legal Affairs banner being displayed at all entry point where an internal user logs into the product/service? An automated pause or slow roll rate is in place to ensure that the banner is read. The Legal Affairs Banner usually carries the following kind of text: "You are authorized to use this system for approved business purposes only. Use for any other purposes is prohibited. All transactional records, reports, e-mail, software and other data generated or residing upon this system are the property of the Company and may be used by the Company for any purpose. Authorized and unauthorized activities may be monitored." NOTE: This is required for all mainframe, mid-range, workstation, personal computer, and network systems.
22.	Has the dial-in connectivity been prohibited on network-connected machine (server and workstation) except where documented and explicitly approved in writing by Business Management and the IT Department?. When explicitly approved, the modem must, as a minimum control, prohibit answer or pickup until after the 5th ring.
23.	Have the remote control products used in a dial-in environment been approved by the IT Department?
24.	Is it ensured that only software (applications /operating systems, etc.) supported by the vendors is being used? (Unsupported software could be vulnerable to attacks since the vendors would not come up with the relevant patches.)
Information Security Administration	
25.	Is there an approved document that clearly outlines the Security Administrator's (SA. responsibility?
26.	Are all the administrative actions (e.g., adding/deleting users, changes to entitlements/passwords) backed up by an independent review?
27.	Does the Security Administrator function review all security audit logs, incident reports, and on-line reports at least once per business day?
28.	In case of Wide Area Networks (WAN), are the router tables maintained securely in Routers?
29.	Are router login IDs and passwords treated as sensitive information and managed by authorised administrators? Are all changes to router table

	entries logged and reviewed independently?
30.	Are access violations taken note of, escalated to a higher authority and acted upon in a timely manner?
31.	Is there a process to report all unusual or suspicious activities? (Reporting to IT Department, investigating immediately, and bringing the case to closure without delay)?
32.	Does the Security Administrator function assess compliance with their security procedures quarterly and reports their results to the IT Department?
33.	Have all the all security related administrative procedures under the control of the Security Administrator been documented and approved by management (annual exercise)? The minimum procedures should include: • Information Ownership • Data Classification • User registration/Maintenance • Audit Trail review • Violation logging and reporting • Sensitive activity reporting • Semi-Annual Entitlement Reviews • Password resets • Escalation reporting
Microcomputer / PC Security	
34.	Do the LAN servers, mail servers, and microcomputers have IT department approved anti-virus products installed (in them?)
35.	Are all product/service specific microcomputers/PCs secured against removal and theft commensurate with the value of the computer and information it holds along with a process to report any thefts to the IT Department?
36.	Are microcomputers / PCs having sensitive information protected with power-on password to prevent unauthorized access?
37.	Are sensitive data in such microcomputers / PCs backed up and preserved properly to ensure recovery in case of failure?

Audit Trails	
38.	Does the audit trail associate with the product/service support the ability to log and review all actions performed by systems operators, systems managers, system engineers, system administrators, highly privileged accounts and emergency IDs?
39.	Do all the financial transactions as well as additions, changes and deletions to customer's and vendor's data, get recorded in the product/ service audit trail?
40.	Does the audit trail for product/service record all identification and authentication processes? Also, is there a retention period for the audit trails? Is it adequate?
41.	Does the audit trail associate with the product/service log all actions by the Security Administrator?
42.	Is there a process to log and review all actions performed by systems operators, systems managers, system engineers, system administrators, security administrators, and highly privileged IDs?
43.	Is there a process in place to log and review actions performed by emergency IDs associated with the product/service?
Violation Logging Management	
44.	Is the product/service capable of logging the minimum criteria specified to log and report specific security incidents and all attempted violations of system integrity?
45.	Are the product/service owners aware of their responsibilities with respect to Security incident reporting?
Information Storage and Retrieval	
46.	Has all the media (File/Floppy/Disks/Tapes etc. under the control of the product/service owner been marked (with the classification or have these been classified?) and securely stored with access restricted to authorized personnel only?
47.	Is there a process in place to ensure that all media under the control of the product/service owner containing critical information is destroyed in a manner that renders the data unusable and unrecoverable?
48.	Is there a procedure in place that enforces and maintains a clean desk

	program, which secures all critical information from unauthorized access?
Penetration Testing	
49.	Is it ensured that products/services that use the Internet for connectivity or communications have undergone a successful penetration test prior to production implementation?
50.	Is there a penetration test process that ensures whether modifications to the product/service that uses the Internet for connectivity or communication have been reviewed to determine whether a subsequent penetration test is warranted?
51.	Is there an intrusion detection system in place for all the external IP connections?

Questions

1. .....is a method used to gather information about the communication network.
   a. Port Numbering
   b. Port Scanning
   c. Port Listing
   d. Port Skipping

2. Congestion at network due to buffer overflows and packet dropping leads to a message confidentiality threat named……..
   a. Missing Traffic
   b. Communication delivery
   c. Misdelivery
   d. Late delivery

3. The Ping of death, connection flooding and traffic redirection are network vulnerabilities called ……..attacks that result in loss of network availability.
   a. Dumpster of Data
   b. Denial of signal
   c. Dumping of service
   d. Denial of service

4. The Link-Encryption network control provides ..... authentication and is used when the communication line is of high vulnerability.
   a. Node
   b. User
   c. Data
   d. Signal

5. …….is a control designed to address spoofing, eavesdropping, and session hijacking and uses a standard means for handling encrypted data. It is implemented at the IP layer.
   a. Internet Protocol Security (IPSec.
   b. Secure Socket Layer (SSL)
   c. Public Key Infrastructure (PKI)
   d. Secure Multi Purpose Mail Extensions (S/MIME)

6. Cryptographic checksums is a network control that ……
   a. Adds a parity bit after adding the data bits.
   b. Translates data in a file into a hash value used as checksum.
   c. Transmits the data after encryption.
   d. Translates the data into a parity checksum combination.

7. ……is one of the common methods used in user management control by an authentication server.
   a. Terminal Access Controller Access Control System
   b. Data Access Controlling System
   c. Terminal Access Contact Access Communication System
   d. Remote Terminal Access Connecting System

8. …….is a detective control that raises an alarm when a suspicious audit log entry is by an authorized access.
   a. Intermediate Detection System
   b. Intrusion Detection System
   c. Immediate Data Users
   d. Detection Signal Users

9. The Auditor checklist to check controls on network security requires to take special considerations on ……..
   a. Management and change controls on network devices
   b. Event logging and monitoring of logical access paths
   c. Only a
   d. Both a and b

10. The intrusion detection monitoring is a ……. on a host for data integrity attack by malicious software.
    a. Supportive control
    b. Corrective control
    c. Detective Control
    d. Preventive Control

11. The activity of testing undertaken by an organization with the help of teams to exploit the security vulnerabilities of its enterprise network is called…….
    a. Intrusion Detection Testing
    b. Preventing Detection Testing
    c. Post-Implementation Testing
    d. Penetration Testing

12. A testing team member when posing as a representative of the IT department's help desk and asking users to divulge their user account and password information is a …….type of penetration testing.
    a. Social engineering
    b. Team engineering
    c. User testing
    d. User engineering

13. Identify the control implemented to prevent the network vulnerability that monitors OS and application fingerprinting.
    a. Penetration Testing
    b. Cryptography methods
    c. Intrusion Detection Systems
    d. Immediate version System

14. ……vulnerability is an authentication failure which can be controlled by using encrypted channel and one-time authentication.
    a. Salami Technique
    b. Spoofing
    c. Buffer overflowing
    d. Brute-force Method

15. Intrusion detection /prevention system (IDS/IPS) are network vulnerability management systems implemented in the …….level.
    a. Application
    b. Data
    c. Perimeter
    d. Network

16. The ……..protects the firewall, ensuring that any change in the rules at a later, time will not inadvertently compromise it. .
    a. Lockdown
    b. Listdown
    c. OS banner
    d. Segmentation

17. "Does the organization have a comprehensive password construction, implementation and management policy?" This checks the ……. of the network security while auditing.
    a. Process
    b. Authentication
    c. Cryptography
    d. Assessment

18. Identify the checklist question used to check access control while auditing network security.
    a. Is there a process in place to ensure that all media under the control of the product/service owner containing critical information is destroyed in a manner that renders the data unusable and unrecoverable?
    b. Does the financial transactions as well as additions, changes and deletions to customer's and vendor's data, get recorded in the product/ service audit trail?
    c. Does the product/service support the requirement to limit individual user sessions to a maximum of X minutes of inactivity using either session time out or a password protected screen saver?
    d. Is there an Information Security guidelines document, which defines the minimum configuration for any device/link on the organization's network, including levels of encryption?

19. The network vulnerability of traffic re-direction can be prevented by using the ……..and …… controls.
    a. Awareness, Programming controls
    b. Error detection, Authentication
    c. Assessment, monitoring
    d. Encryption, Audit trail

20. War-Dialing is a type of ……..
    a. Firewall
    b. Denial of service
    c. Penetration testing
    d. Wire testing

21. The user authenticates with his …….in a strong authentication device working on challenge response system method.
    a. Logon time
    b. Personal Identification Number
    c. Host Identification address
    d. Network Identification Line

22. Which of the following is an advantage of using link encryption?
    a. Individual nodes in the network do not have to be protected.
    b. The exposure that results from compromise of an encryption key is restricted to a single user who owns the key
    c. It prevents messages from traffic analysis attacks
    d. The users of the network can bear the cost of link encryption.

23. ……..is a malicious code that can be used by a user to invoke/services on the server.
    a. Cookies
    b. Scripts
    c. Active Code
    d. Viruses

24. A message authentication code is used to protect against:
    a. Changes to the content of a message
    b. Traffic Analysis
    c. Release of message contents
    d. Password being transmitted.

25. …… is an attack that adds spurious entries to a table in the server that deals with the conversion of www.icai.org into network address like 202.54.74.130.
    a. Host Name Redirection
    b. Traffic Name Server
    c. Data Name Server Attacks
    d. Domain Name Server Attacks

Answers:

1. b	2. c	3. d	4. a	5. b	6. b
7. a	8. b	9. d	10. c	11. d	12. a
13. c	14. b	15.d	16. a	17. b	18. c
19. d	20. c	21. b	22. c	23. b	24. a
25. d					

4 Application Controls

Learning Objectives

- Application Boundary Controls
- Input Controls
- Data Processing Controls
- Datafile Controls
- Output Controls

Introduction

Over the last several years, organizations around the world have spent billions of dollars upgrading or installing new business application systems for reasons ranging from tactical goals, such as year 2000 compliance, to strategic activities, such as using technology to establish company differentiation in the marketplace. An application or application system is a software that enables users to perform tasks by employing a computer's capabilities directly. These applications represent the interface between the user and business functions. For example, a counter clerk at a bank is required to perform various business activities as part of his job and assigned responsibilities. From the point of view of users, it is the application that drives the business logic.

Application controls pertain to individual business processes or application systems, including data edits, separation of business functions, balancing of processing totals, transaction logging, and error reporting. Therefore, the objective of application controls is to ensure that:

- Input data is accurate, complete, authorized, and correct.
- Data is processed in an acceptable time period.
- Data stored is accurate and complete.
- Outputs are accurate and complete.
- A record is maintained to track the process of data from input to storage and to the eventual output.

From an organizational perspective, it is important that application controls:

- Safeguard assets
- Maintain data integrity

- Achieve organisational goals effectively and efficiently

Application Exposures

i. **Weak Security**: Increased end-user access to the enterprise-wide information system to critical information of an organization at varying levels lead to security risks. The primary source of security coverage involves end-user awareness.
ii. **Unauthorized access or changes to Data or Programs**: Segregation of access rights to an application for a user should follow the "least privileges" rule with respect to his/her job responsibilities. The access to production data and programs for end-user and IT staff are to be regulated within different layers.
iii. **Unauthorized remote access:** In today's global enterprise scenario the need for remote access to a company's resources plays a vital role in its marketability. Remote access should avoid unauthorized access with callback features to identify users along with mechanisms like encryption, smart cards, digital signatures, etc.
iv. **Inaccurate Information:** User access to information from an application is critical with reference to job requirements. Redundant or wrong information results in business discrepancies.
v. **Erroneous or falsified data Input:** Wrong data collected, manually processed or prepared for entry causes undesirable system performance.
vi. **Misuse by authorized end users**: Legitimate use of the system by end-users within the scope of their job requirements.
vii. **Incomplete processing:** Transaction or files not processed to completion due to an error during an online or an offline processing or failure of a trigger.
viii. **Duplicate Transaction Processing**: This is a wrong process trigger due to multiple or duplicate transactions.
ix. **Untimely processing**: When an online system posts transactions to a batch system it causes a delayed time of processing for say crucial financial process which are to take place on a daily closing report.
x. **Communication system failure**: Information exchange happening in a networked enterprise is vulnerable to accidental failures like intentional interception and modification of messages by unauthorized users.

Components of Application Controls

Application controls can be broadly classified as follows:

i. **Boundary Controls:** Controls to ensure that access to the application is restricted only to authorised users and that it protects systems from unauthorized access.
ii. **Input Controls:** Controls to ensure that only complete, accurate and valid data and instructions form an e input to the application.

iii. **Processing Controls:** Controls to ensure that there is only authorised processing and integrity of processes and data is ensured.
iv. **Datafile controls:** Controls to ensure that data resident in the files are maintained consistently with the assurance of integrity and confidentiality of the stored data.
v. **Output Controls:** Controls to ensure that outputs delivered to the users in a consistent and timely manner in the format prescribed/required by the user.

Application Boundary Controls

The objective of boundary controls is to prevent unauthorized access to applications and their data. Such data may be in any stage, in input, processing, transit or output. The controls restrict user access in accordance with the business policy of an organization and its structure; and protect other associated applications, systems software, database and utilities from unauthorized access.

Access controls may be implemented by using any of the logical security techniques embedded in the application software. Besides access security implemented at the operating system and/or database management systems level, a separate access control mechanism is required for controlling access to application. The application is to have boundary controls to ensure adequate access security to prevent any unauthorized access to:

- Applications themselves
- Application data during communication or transit
- Stored application data
- Resources shared with other processes

The above objectives can be achieved by adopting logical security techniques like:

- Using logon ids and passwords
- Providing access to application from specified terminals only
- Using **Cryptographic Controls, that is,**
    - Encrypting all data leaving and entering the application process.
    - Encrypting intermediary data either in input, processing or output stage stored in database.
    - Ensuring confidentiality, integrity and availability through encryption and authentication of all exchanges of data/processes between applications. The authentication of users and process usually occur at the network level since it is cumbersome to authenticate r each and every application. See the next chapter for authentication of users at the network layer.

- Uusing **audit trails,** i.e., logging all significant events that occur at the boundary of the application. These events include all or any of the following:
    - Identity of the user requesting access.
    - Authentication supplied by the user.
    - Number of incorrect login attempts
    - Terminal-id.
    - Time of login and log-out.
    - Actions permitted/denied.

Input Controls

Input controls are responsible for ensuring accuracy and completeness of data and instruction input into an application system. These controls are important since substantial time is spent on input of data, involving human intervention and therefore prone to error and fraud. Input controls address the following:

a. Source Document Design
b. Data entry screen design
c. Data code controls
d. Batch Controls
e. Data Input Validation Controls
f. Data Input Error Handling and Reporting
g. Instruction Input Controls

Source Document Design

Source documents are used as an intermediary medium to record data before being used for data entry (input) into the system, such as a sales invoice prepared manually by using a typewriter in a branch office for want of computer infrastructure. A well-designed source document helps improve control in the following ways:

- Reduces data entry errors
- Increases speed of data entry
- Ensures better control over the process
- Assists subsequent reference

Source document design begins with an analysis of the need and usage of the source document. Source document design includes the following:

- Material to be used for the source document.
- Layout and style of the source document.

For designing the layout of the source document, the following should be kept in mind:

- Include instructions for completing the form.
- Minimize the amount of handwriting.
- Data to be entered (keyed. be sequenced so that it can be read like a book: that is, top-to-bottom and left-to-right.
- Capture only variable data.
- Not to capture data that can be calculated or stored in computer programs as constants.
- Use organization-wide consistent business codes for appropriate attributes.

Data entry screen design

Data is keyed into online forms or tables. The system plays a role in the data entry process, guiding users who need help, checking data entries to detect errors, and providing other kinds of data processing aids. A designer of user interface software has to take into consideration computer processing logic as well as data input by the user. While data entry is heavily emphasized in clerical jobs, all jobs involve a bit of data entry. . The general objectives of designing good data entry screens are to establish consistency of data entry transactions, minimize input actions and resource requirements on the system, ensure compatibility of data entry with data display, and provide flexibility of user control of data entry. Auditors must be able to evaluate the data entry screens and come to a judgment about the possible extent of errors.. This judgment will affect the extent of their checking. The primary objective is to design the data-entry screen based on the respective source documentation.

The factors that should be considered while designing data entry screens are detailed below.

- Screen organization: Good screen designs are symmetric. Further, screens should be uncluttered and easy to follow the pattern of data flow. With increase in data elements, the complexity increases and therefore multiple screens with a logical changeover should be used. As much as possible where a source document is involved, the screen organization should be identical to the source document. Consistency of the screen layout should be ensured and the same design should be repeated wherever possible.
- Caption design: Captions guide the users about the nature of the data to be entered. Some of the points to be considered are as follows:
    - Make field captions consistent; always employ the same caption to indicate the same kind of data entry.
    - Ensure that captions are sufficiently close to be associated with their proper data fields, but are separated from data fields by at least one space.

  - Make captions for data fields distinctive, so that they will not be readily confused with data entries, labeled control options, guidance messages, or other displayed material.
  - The caption for each entry field should end with a special symbol(":"), signifying the start of the entry area.
  - Captions should employ descriptive wording, or else standard, predefined terms, codes and/or abbreviations; avoid arbitrary codes.

- Data entry field design: Data entry fields should either be to the right of the caption or exactly below it. Provide underscores in fields to indicate a fixed or maximum length specified for a data entry. Where source documents are used, the pattern should be based on the source document. Option buttons and check boxes should be used when the user has to choose from a small list of options. In case users have to choose from a long list of options, list boxes can be used.
- Tabbing and skipping: During data entry, the user moves from field to field by pressing the "Tab" key. It is important to ensure that the Tab order is consistent, so that the insertion point moves from the first field on the screen to the last without skipping fields. Incorrect tab order will not only frustrate users, but may cause them to enter data in the wrong field. Automatic tabbing should be avoided since fields may be skipped for data entry.
- Colour: Colours help reduce search time and make the screen interesting. However, bad usage of colour may distract or confuse the users. The following should be kept in mind when deciding the colours.

  - Bright colours should be avoided. Only soft or pastel colours or those that provide good contrast for the fields and captions should be used.
  - Uppercase and lowercase may be used to differentiate captions, if the display is monochrome or if the users have difficulty in distinguishing colours due to eye defects, poor or excessive lighting, etc.

- Display rate: Display rate is the rate at which characters or images are displayed. Data entry screens should have a fast and consistent display rate. If the rate is slow or inconsistent, the data entry is prone to errors.
- Prompting and help facilities: Descriptive help should be added wherever possible. Data entry forms should be as self-explanatory as possible but should also include help for each field. Prompting of actions by the user can be provided by using pop-up messages that appear on placing the cursor on a field. More prompting and help is required in the case of direct data entry where no source document is involved.

Data code controls

Data codes are used to uniquely identify an entity or identify an entity as a member of a group or set.

Types of data coding errors:

- **Addition:** Addition of an extra character in a code
- **Truncation:** Omission of characters in the code
- **Transcription:** Recording wrong characters
- **Transposition:** Reversing adjacent characters
- **Double transposition:** Reversing characters separated by one or more characters, i.e., 45123 is entered as 42153.

Factors affecting coding errors are:

- **Length of the code:** Long codes are naturally prone to more errors. Long codes should be broken using hyphens, slashes or spaces to reduce coding errors.
- **Alphabetic numeric mix:** The code should provide for grouping of alphabets and numerals separately. Inter-mixing the two can result in errors.
- **Choice of characters:** Certain alphabets are confused with numerals such as B, I, O, S, V and Z would be confused with 8,1,0,5,U, 2 when written on source document and entered into the system. Such as characters should be avoided
- **Mixing uppercase/lowercase fonts:** Upper case and lower case should NOT be mixed when using codes, since they delay the process of keying in due to usage of the shift key. Such codes are also prone to errors.
- **Sequence of characters:** Character sequence should be maintained as much as possible. Such as using ABC instead of ACB.

Check digits are redundant digits that help verify the accuracy of other characters in the code that is being checked. The program recalculates the check digits and compares with the check digit in the code when it is entered to verify if it is correct. Check digits may be prefixes or suffixes to the actual data. Since these take time to calculate, they should be used only on critical fields.

Batch Controls

Batch controls group input transactions into logical or physical batches. Physical batches are groups of transactions that constitute a physical unit such as a set of invoices pertaining to a branch. Logical batches are groups of transactions that are divided on a logical parameter, such as cut off date, or documents pertaining to division or branch.

Control over physical batches is ensured through batch header forms, which are data preparation sheets containing control information about the batch.

Types of batch controls are as follows:

- **Total financial amount**: Counter-checking that the total financial amount of items processed in that batch match with the total individual financial amounts in the fields. For example, if a batch of 100 invoices is processed, it should be verified that the gross total of individual item sale value in the invoices matches the total value of sales of all the invoices processed.
- **Total items**: Counter-checking that the total number of items included on each document in the batch matches the total number of items processed. For example, the total number of units sold as per each invoice should match the total number of items processed in the batch.
- **Hash totals:** Hash totals are totals of any numeric field in the batch such as serial number of invoices to counter check the total of the field after input of data.
- **Total documents**: Total number of documents processed should also be counterchecked for correctness.

Data Input Validation Controls

Input Authentication Controls: One of the important objectives of application controls is to ensure that the data that is input into the application is valid and authorised. This can be ensured in many ways; some of these are given below:

- Manual signatures on input document
- In case of online inputs, input menu is available for specified logins only to ensure that only the personnel having authority enter the document and are accountable for the transaction they enter. Further input may be restricted through predetermined specified terminals.
- Restricting certain types of input to be enabled on a unique password at the input menu/form level.
- Scanned Input using OCR (Optical Character Recognition), Barcode Readers, MICR (Magnetic Ink Character Recognition), etc.

  The input system should validate the data that is submitted for processing as close to the point of origin as possible. Data validation controls identify data errors, incomplete or missing data and inconsistencies among related data items. Normally data validation controls are preventive in nature since they ensure that incorrect data is not entered in the system.

- **Edit Controls**: Edit controls are the principal data validation controls and are used to validate data. They contain the following:

  - Sequence checks: Controls that verify if the data maintains a proper sequence or order. They are usually used to check serial numbers of documents or the date sequence of transactions.
  - Range and Limit check: Range check means fixing upper and lower limits for data values to ensure that they don't exceed them. Limit checks refer to only upper or lower limit of data values, but not both. For example, if the edit check requires that gross salary should not exceed Rs.100000/- per month, it is a limit check. If the check requires that the percentage of tax should be between 10% toof 30% on the gross total income, it is a range check.

- Missing data check: Ensures that certain key fields are not left blank during data entry.
- Duplicate check: Duplicate check ensures that the same data is not keyed twice. For example when entering invoices, the same invoice number is not repeated twice.
- Programmed Validity Check: Logical validations may be built in an application to check invalid input. E.g., there can be a validation to prevent a user from entering a non- existent account code.
- Dependency Match: Where certain fields depend on input values in other fields, programmed checks can be used for internally validating data before accepting input. For example, Date of joining may be compared to the date of birth to ensure that the latter is earlier than the former.
- Completeness check: Completeness of data may be checked before accepting data entered. For example, while creating a new e-mail user id, the input screen will not be accepted without date of birth, name, address etc. (marked as required fields).
- Reasonableness check: Reasonableness of value entered may be compared with an acceptable range within the system before accepting input, e.g. age entered as 140 may be rejected as unreasonable.
- Table lookups: Input entered is matched with a range of values in a table before acceptance. E.g.. a customer code entered by the operator is internally matched with a table of valid customers for a successful match, or else the transaction is rejected.

Data Input Error Handling and Reporting

Input error handling and reporting requires that controls that identify errors and correct them especially in a batch processing system where input and validation may take place at different points of time. Input errors can be handled in the following ways:

- **Rejecting only transaction with errors:** Only those transactions containing the errors would be rejected and the rest of the batch would be processed.
- **Reject the whole batch of transactions:** The whole batch of transactions is rejected if even a single error is found in it. This is usually done in cases where transactions have to be in a sequential order and data entry has not been made in that order.
- **Accepting batch in suspense:** The batch would be held in suspense till the corrections in transactions are made. However, the batch or the transactions are not rejected.
- **Accepting the batch and marking error transactions:** Erroneous transactions are separately marked for identifying later and the whole batch is processed. This is usually done in an online system.

Instruction Input Controls

Validating instructions is more difficult than validating data. This is because the instruction input requires more user interaction and decision discretion than in the case of data input. However, it has to be ensured that users are brought under a set of action privileges and their instructions are controlled and logged.

- **Instruction input methods:**There are different types of interfaces to give instructions depending upon the application system and the extent of control to be exercised on them. Some of the available instruction interfaces are:
- **Menu Driven Applications:** These applications provide a set of instructions that are fixed and the users only need to choose the actions to be done depending on the options available in the menu. Such instructions can be well controlled since the choices available to users can be restricted based on the content and context.
- **Question Answer dialogs:** These are mainly used for capturing user preferences, but occasionally data too. These interfaces ask a series of questions to the user and provide a set of options from which the user is required to make a selection. For example, a financial software may ask the user a series of questions regarding present value, interest rate, periodicity of payment, etc. and the then provide the requisite output.
- **Command Languages:** These are languages which require users to specify commands to the system to complete a set of processes. Each command may have several parameters that modify the behaviour of the command. The DOS operating system is an example of a command based operating system. Structured query language (SQL) which is used to interrogate databases is another example of command language.

Reporting Instruction Input Errors

Error messages and procedural instructions need to be communicated to users at the instance of a possible error occurrence. The error message must be complete and meaningful and help the user correct it immediately. Different error messages may be given based on the expertise of a user.

Processing Controls

Data processing controls perform validation checks to identify errors during the processing of data. They are required to ensure both the completeness and accuracy of the data being processed. Normally the processing controls are enforced through the database management system. However, adequate controls should be enforced through the front end application system also to ensure consistency in the control process.

i. Data processing controls

- **Run-to-run totals:** These help in verifying data that is subject to process through different stages. If the current balance of an invoice ledger is Rs.150,000 and the additional invoices for the period total Rs.20,000 then the total sales value should be Rs.170,000. A specific record (probably the last record. can be used to maintain the control total.
- **Reasonableness verification**: Two or more fields can be compared and cross verified to ensure their correctness. For example, the statutory percentage of provident fund can be calculated on the gross pay amount to verify if the provident fund contribution deducted is accurate.
- **Edit checks:** Edit checks similar to the data validation controls can also be used at the processing stage to verify accuracy and completeness of data.
- **Field initialization:** Data overflow can occur, if records are constantly added to a table or if fields are added to a record without initializing it, i.e. setting all values to zero before inserting the field or record.
- **Exception reports:** Exception reports are generated to identify errors in the data processed. Such exception reports give the transaction code and why a particular transaction was not processed or what is the error in processing the transaction. For example, while processing a journal entry if only debit entry was updated and the credit entry was not updated due to the absence of one of the important fields, then the exception report would detail the transaction code, and why it was not updated in the database.

Datafile Controls

- **Version usage**: Proper version of a file should be used for processing the data correctly. In this regard it should be ensured that only the most current file be processed.
- **Internal and external labelling**: Labelling of storage media is important to ensure that the proper files are loaded for process. Where there is a manual process for loading files, external labelling is important to ensure that the correct file is being processed. Where there is an automated tape loader system, internal labelling is more important.
- **Data file security**: Unauthorized access to data file should be prevented, to ensure its confidentiality, integrity and availability. These controls ensure that the correct file is used for processing and are not concerned with data's validity.
- **Before and after image and logging**: The application may provide for reporting of before and after images of transactions. These images combined with the logging of events enable re-constructing the datafile back to its last state of integrity, after which the application can ensure that the incremental transactions/events are rolled back or forward.
- **File updating and maintenance authorization**: Sufficient controls should exist for file updating and maintenance to ensure that stored data are protected. The access restrictions may either be part of the application program or of the overall system access restrictions.
- **Parity Checking**: When programs or data are transmitted, additional controls are needed. Transmission errors are controlled primarily by detecting errors or correcting codes.

Output Controls

Output controls ensure that the data delivered to users will be presented, formatted and delivered in a consistent and secured manner. Output can be in any form: either printed data report or a database file in a removable media such as a floppy disk, CD-ROM or removable hard disk. Whatever the type of output, confidentiality, integrity, and consistency of the output is to be maintained.

The following form part of the output controls:

- **Storage and Logging of sensitive, critical forms**: Pre-printed stationery should be stored securely to prevent unauthorized destruction or removal and usage. Only authorized persons should be allowed access to stationery supplies such as security forms, negotiable instruments, etc.

- **Logging of Output program executions**: When programs used for output of data are executed, these should be logged and monitored, otherwise confidentiality of the data could be compromised.
- **Spooling / Queuing:** "Spool" is an acronym for "Simultaneous Peripherals Operations Online". This is a process used to ensure that the user is able to continue working, while the print operation is getting completed. When a file is to be printed, the operating system stores the data stream to be sent to the printer in a temporary file on the hard disk. This file is them "spooled" to the printer as soon as the printer is ready to accept the data. This intermediate storage of output could lead to unauthorized disclosure and/or modification. A queue is the list of documents waiting to be printed on a particular printer; this should not be subject to unauthorized modifications.
- **Controls over printing**: Outputs should be made on the correct printer and it should be ensured that unauthorized disclosure of information printed does not take place. . Users must be trained to select the correct printer and access restrictions may be placed on the workstations that can be used for printing.
- **Report distribution and collection controls**: Distribution of reports should be made in a secure way to ensure unauthorized disclosure of data. It should be made immediately after printing to ensure that the time gap between generation and distribution is reduced. A log should be maintained for reports that were generated and to whom these were distributed. Where users have to collect reports the user should be responsible for timely collection of the report, especially if it is printed in a public area. A log should be maintained about reports that were printed and collected. Uncollected reports should be stored securely.
- **Retention controls**: Retention controls consider the duration for which outputs should be retained before being destroyed. Consideration should be given to the type of medium on which the output is stored. Retention control requires that a date should be determined for each output item produced. Various factors ranging from the need of the output, use of the output, to legislative requirements would affect the retention period.

Existence Controls in Application Systems

Existence controls ensure the continued availability of the application system and data in a consistent manner to the users. These form an integral part of the input, processing and output controls. Recovery of the application system from failures and restoration of both standing data as well as transaction data is very critical. Therefore, existence controls should include backup and recovery procedures of data. This requires secure storage of data files. Existence controls over processing of data should include adequate checkpoint/restart controls that recover the process from a

failure without having to repeat the entire process from the beginning. Existence controls should also be exercised over output to prevent loss of output of any form i.e., paper, spool files, output files, etc. Finally, recovering the application system accurately, completely and promptly can be critical to many organizations, especially if they are in an EDI environment.

Audit of Application Controls

The audit of application controls requires the IS auditor to ensure that the application system under audit

- Safeguards organizational assets
- Maintains data integrity
- Achieves organisational goals effectively
- Consumes resources efficiently

When reviewing the application controls, the IS auditor should ensure that the audit objectives with regard to confidentiality, integrity and availability of organisational information are orderly and complete. The process of application control review will include:

i. Obtaining preliminary information

- Based on general understanding of the business process and risks assessment, the IS auditor should determine the application owners and users.
- Determine the application objectives, functionality and features that map onto the business process(es) of the organisation to determine if functional design incorporates key control points.
- In case of applications developed in-house, determine if the control systems development methodology was adequate and in the case of bought-out applications determine if there were adequate controls with respect to escrows (insurance/monetary payments/contract), warranties and support.
- Determine the IT environment in which the application is installed and operated. This would include gaining an understanding of the hardware, networking, operating systems and other systems software. The technical and application administration manual, programmer manual and user manual must contain information relevant to control.

ii. Determining whether

- The application provides adequate level of access controls protecting the application data and application program.

- Besides providing adequate preventive access controls, does the application provide for detective control such as check sums, audit trails, and transaction logs.
- Controls over the preparation, collection, and processing of source documents ensure the accuracy, completeness and timeliness of data before it reaches the application.
- Manual and automated controls over data entry (batch or online), data validation, error identification and reporting are adequate and effective.
- Error correction and re-entry is effective and the data is completely and accurately entered into the application.
- Controls over application programs and related computer operations ensure the accuracy, completeness, and timeliness of data during batch or real time processing and are adequate and effective.
- Controls over file handling, file access, and backup and recovery are effective and adequate so as to ensure the completeness and accuracy of data during the process of data storage and retrieval.
- Controls over balancing and reconciliation, distribution of output, handling of negotiable documents, and output retention are effective to ensure that output is accurate and distributed to authorised personnel on a timely basis.

iii. Application Audit Best Practices

The auditor could follow these up to ensure the proper procedures were followed:

- Apply defense-in-depth.
- Use a positive security model.
- Fail safely.
- Run with the least privilege
- Avoid security by obscurity.
- Keep the security simple.
- Detect intrusion and keep logs.
- Never trust infrastructure and services.
- Establish secure defaults.
- Use open standards.

Summary

Application controls are specific control procedures over the applications which can provide assurance that all transactions are authorized and recorded, processed completely, accurately and on a timely basis. Application controls may consist of manual procedures carried out by users (user controls) and automated procedures or controls performed by the computer software.

Testing application controls is achieved by gaining evidence that a control has operated. For example, if a computer enforces segregation of duties between various finance functions the auditor may check the access control lists and the user permissions on those lists. Alternatively, if the computer has automated controls to ensure that purchase order clerks can only order goods from a predefined list of approved products from approved suppliers (regularity assurance), the auditor may check the access controls to the approved products list, together with a sample of new additions to the list.

Evidence collected about the operation of computer controls is obtained from a combination of observation, enquiry, examination and sampling. The auditor may also be able to use computer assisted audit techniques to assist in the examination of controls. For example, the auditor could download the audit log file and write a routine to extract unauthorized access attempts.

Some examples of Application Control Techniques and their Suggested Audit Procedures.

Audit procedures for application level controls are segregated into the following areas of critical evaluation.

i. Application security Management
ii. Application access controls
iii. Application configuration management
iv. Segregate user access to conflicting transaction activities and monitoring
v. Application Contingency planning

Control Activity	Control Techniques	Audit Procedures
Application security Management plan	The security plan would include: - Identification of risk level. - Interconnection/information sharing. - Implementation of controls and special considerations. - Process for developing security roles with administration policies, role maintenance and development. - Identification of sensitive transactions and high risk user roles.	Inspect and review the security plan procedures and documentation addressing the security issues and audit their appropriateness and completeness. Observe and inspect authorization of critical activities and user access.

	- Procedures for emergency access to production system including program or database update or modification and security parameter settings compliant with the organizations policies. - Identify critical user IDs and business process or sub-process with their appropriate access privileges. - User "least privileges" preventing execution of incompatible transactions from application menu interface.	
Application risk assessment and periodic monitoring	Periodic assessment of application and supporting systems. Documentation of risk assessment, validation and approvals and as part of the security plan.	Obtain the recent risk assessment documentation and check whether technology and business requirements have been appropriately taken into account.
Policies and procedures to assess the access to applications periodically.	Documentation of procedures and policies regarding business security needs and segregation of user activity from application/data administrator activity. Owner and user awareness of security policies.	Through interview determine the following: - Business security needs. - segregation of duties. - system administrator activity. - awareness of the security policy and authorizations.
Management responsibility on security policies and procedures and compliance.	Application security policy and procedure test plan is documented. Security controls related to each application activity is tested periodically.	Inspect the overall testing strategy for its efficiency and compliance with best practices. Inquire management on plans for corrective action plans, milestones and

	A mission on the frequency and scope of testing the security policies is in place Identify the weakness and initiation of corrective action plans, milestones and tested with a periodic monitoring plan. Compliance and report on the part of the application entity is being assessed.	resolution plans. Based on the application test plan assess the frequency and scope of testing aligned with the given risk and criticality of the application. Evaluate the corrective action plans and inspection of evidence.
Application boundaries and user access are appropriately authenticated and protected.	Adequate security issues are identified and is included as part of the security policies. Approval of user access to the application. Guidelines for assigning, changing and resetting and handling lost or compromised passwords.	Review the security plans with respect to application boundaries. Inquiry and observation and inspection to assess the procedures and policies on password management.
Controlled access to the application.	User access to the application is aligned with a valid business purpose and periodically reviewed for appropriateness. Authorization techniques, use of digital signatures and least privileges methods are used to control public access to the applications. Sensitive transactions, inactive accounts, questionable authorizations are monitored. Inbuilt procedures and monitoring to handle security violations.	Inquire and inspect the following: Identification and authentication Read, write, modify and delete privileges. Access prohibition to live or production databases. Interview and review the access to approved users to transactions and process responsibilities. Logging along with regular inspection carried on security violations and review of security

		parameters used within applications and support applications.
Policies and procedure for change management for application functionality.	Policies and procedures are established for current application configuration management and change requirements. A SDLC methodology is prepared to provide a structured outline of accepted concepts and practices with the preview of active user process involvement. Change requests are to be documented and approved by both users and IT staff.	Inspect documents identifying key transactions that provide user access to application functionality changes. Review the system documentation of SDLC methodology defined with the followed or system implemented methodology. Examine and inspect recent software modification request forms and the procedures followed.
Access to program libraries is restricted.	Segregation of libraries of programs containing production code, source code, support programs and their maintenance. Accesses to all libraries are to be protected by access control software and operating system features. Responsibility of a group of users to approve and log movement/change of programs and data among libraries with respective application revision approvals. Access to application assigned to users involving critical	Verify source code compile dates; compare module size to production load module size. For critical transactions process check if the access software control rules have been defined. Review procedures used for authorizing library program or data exchange and maintenance of before and after images. Test access to libraries by examining security system

	transactions in production data are to have the necessary approval.	parameters.
Access to application activities, processes, transactions, programs and tables are controlled.	To have a defined user role access approved by management with authorization for critical transactions, changes to application programs, codes and tables within the production environment. Security design for sensitive administration for critical transactions and restricted user access. Periodic monitoring of key programs and table changes and assessment of compliance with configurable objects and programs. Effective process to document, test and approve emergency application changes.	Inspect system reports, authorization forms and inquire key programs and table with preview of user roles who have access to configuration transactions. Select a sample of user access policies and procedures pertaining to administrative privileges and determine their activity log and diversion from management approved rules. Inspect documented assessments and review exceptional handling procedures.
Effective monitoring of controls in place to monitor incompatible transactions and user functions.	Prepare a duty matrix to identify the risk when segregation of duty conflicts in user roles. Controls by the application to prevent users from executing incompatible transactions, to input and/or approve transactions and appropriate access to transactions supported by business needs. Periodic review of application user	Observe and inspect the user roles from the application access policy listing. Compare the access policy and user profile with management approved procedures for them. Inspect the workflow document with the user

	access to remove conflicts and assign responsibility of access with respect to business requirements.	access roles and responsibilities from the application preview.
Steps to prevent potential damage and interruption with a business impact analysis plan.	Identify the critical functions of the application including IT resources, key programs and data used to perform them. Identify the outage times for the application and appropriate recovery strategies. Develop a time-based backup schedule of application data and programs stored offsite and easily retrievable for contingency implementation. Contingency plan includes - System recovery on an alternate platform. - Periodic tests for disaster simulations. - Test results are documented as lessons-learnt. - corrective measures are incorporated to address deficiencies identified during testing. - Co-ordination of recovery teams, alternate measures and normal restoration processes.	Review the policies and methodology of the recovery priorities and interview program, information technology and security officials. Examine the adequacy of the backup strategies in use. . Determine the efficiency of the control environment during a contingency operations plan and its scope. Review the testing of contingency plans and interview the management to find out their level of awareness. Observe and review a recovery plan and test results.

Master Checklist on Application Controls

The following is an illustrative questionnaire that could be used to review Application Controls within an information system.

S.No.	Checklist
	Each transaction is authorized, complete, accurate, and timely and input only once.
1.	Ensure that manual or operating procedures exist for system users.
2.	Check if appropriate input controls are established by the organization for input data. (*Verify the input verification procedures such as assigning Transaction ID, restriction on duplicate entry, range checks, validity checks, control totals, etc.*
3.	Make sure that transactions are from recognized sources. *(Determine the audit trail for documents prior to input to an application.* *Follow through a document to check that controls ensure that input is only accepted from recognised sources. E.g., a valid timesheet.)*
4.	See to it that transactions are explicitly authorized either by r manual or electronic means.(*Establish how input is authorized.)*
5.	Obtain from Payroll / Personnel a list of staff within the section. Request from the Systems Administrator a list of all users of the system. Ensure that all system users are valid employees and users.
6.	Make sure that password controls are effective in restricting access. *(Ensure that access to the system requires a unique ID and password. Ideally the password should be alphanumeric and changed periodically.)*
7.	Check if input and authorization functions are restricted and separated. *(Is there an effective segregation of duties to prevent data entry operators from authorizing transactions and vice versa?* *Can the system produce a system security report, which includes user access permissions?)*
8.	See to it that input of parameters for processing and other standing data is strictly controlled. *(What controls exist to prevent accidental / malicious changes to fixed data parameters, i.e., tax calculations, pay rise, etc.)* *Check the correctness of key values and data within the system.* *i.e. if a VAT calculation is required, is the standing VAT rate set to 17.5%. Does the system record a history of standing data changes?)*

S.No.	Checklist
9.	Subject data to validation for completeness and accuracy at input stage. *(Establish if key fields are validated, what the criteria is and who ensures this is carried out.)*
10.	Ensure that there are clear procedures for data items rejected on input. *(Ascertain how rejected inputs are treated and reported. From a sample of rejected records, ensure that they are amended and successfully re-input.)*
11.	Make sure that clear timetables exist for input and are adhered to. *(Ascertain who is responsible for authorizing the processing of jobs and what procedures are in place. Are they reviewed on a regular basis?)*
12.	Check if corrections are made to rectify differences, exceptions, duplicate transactions, missing transactions and rejected items, and if they are approved (e.g., maker/ checker, exception report, etc.),
13.	Check if there are any duplicate input records. *(Determine what checks for duplicate input are carried out by the application itself, and how they are reported / followed up. From a sample of 10 reported records, determine the action taken and the reason for making duplicates. .)*
14.	Review and evaluate date input controls.
15.	If the input of data is through batch upload, check if the software has controls to ensure that all the entries in the batch have been uploaded without any omission/ commission (e.g., reconciliation of control totals, etc.)?
16.	Review and evaluate the controls in place over data feeds to and from interfacing systems.
17.	Check whether the application prevents the same user from performing both the functions of entering a transaction and verifying the same.
18.	Find out if the application has adequate controls to ensure that all transactions input have updated the files.
19.	In cases where the same data are kept in multiple databases and/ or systems, check if periodic ' sync' processes are executed to detect any inconsistencies in the data.
	An appropriate level of control is maintained during processing to ensure completeness and accuracy of data.
20.	A clear processing schedule exists and is understood by users and operations staff.

S.No.	Checklist
	(Ask who is responsible for job scheduling. What are the procedures for scheduling jobs, and are they up to date and reviewed on a regular basis?)
21.	All data, including that transferred from other systems, is subject to appropriate validation during processing. *(Determine what consistency s checks exist to ensure that systems are in step. Ee.g., outstanding orders in creditors are consistent with outstanding commitments.)*
22.	Data is processed by the correct programs and written to the correct files. *(Confirm that logs / documentation exists, recording any systems updates that may affect validity of processes and any legislative changes.)*
23.	Programs provide confirmation that processing has been completed successfully. *(Examine job journals for evidence that jobs have been completed.* *Check if any failures have been logged / re-run.)*
24.	Assurance is provided that all records have been processed. *(From a sample of control reports, ensure that totals carried forward at the end of one run equal those brought forward at the start of the next.* *Investigate any errors or warnings.)*
25.	Procedures exist for handling rejected records. *(Ensure that procedures do exist.)*
26.	There are adequate procedures for investigation and correction of differences or exceptions identified by the controls over update for completeness and accuracy.
27.	Controls are adequate for the programmed procedure that generates the data. (*Verify the controls implemented like recalculations (Manual), Editing, Run-to-run totals, and Limit checks, etc.)*
28.	The system should provide a means to trace a transaction or piece of data from the beginning to the end of the process.
	Controls ensure the accuracy, completeness, confidentiality and timeliness of output reports and interfaces.
29.	Appropriate staff should carry out checks to ensure that output is reasonable, consistent and complete.

S.No.	Checklist
	(Review any records that are maintained to control output distribution and check that they are adequate and are sufficient to identify who has received the authorized outputs. What procedures exist for controlling the use of stationery? Observe storage arrangements for output. Is physical access controlled? Test a sample of output reports.)
30.	Output should be clearly identified and include information that demonstrates completeness. *(Ensure reports are easily identifiable as output markings such as* *Date / time stamps; End of Report messages; clear titles and headings, etc.)*
31.	Distribution of output should ensure that it goes to the correct location / users and that confidentiality is maintained. *(Determine and observe arrangements for distributing output and consider whether full precautions are taken when handling valuable or confidential output.)*
32.	The usefulness of output should be kept under review. *(Determine what reviews are in place to determine the usefulness of output reports. Are bulky printouts consistently required? Is an electronic copy sufficient?)*
33.	Confidential output should be disposed of securely. *(Ensure that guidelines exist for the disposal of confidential data. Review current procedures.)*
34.	Outputs viewed/ generated by users are only on need to know basis. *(Check whether outputs cannot be generated by all and sundry users in the system. Verify the procedure on the generation, distribution, authentication and preservation of outputs).*
35.	Unique source information should be retained for all transactions. *(Ensure that a unique identifier exists for all inputs to the system.* *I.e. invoice reference, date of entry, date last updated, feeder system, etc.)*
36.	Input documents and output reports should be filed to facilitate transactions through the system. *(Attempt to trace input documents to data held on computer files, and records on output reports to original documents.)*
37.	It should be possible to break down totals on control reports into transactions that form the totals.

S.No.	Checklist
	(Review application output and check whether listings are produced or can be generated which substantiate reported control totals)
38.	When records are posted from one financial system to another, those input to the second should agree with those output from the first. *(Follow up and reconciliations of data between financial systems)*
39.	Determine the need for error/exception reports related to data integrity, and evaluate whether this need has been fulfilled**.**
40.	Review and evaluate the application's authorization mechanism to ensure that no user is allowed to access any sensitive transactions or data without first being authorized by the system's security mechanism.
41.	The application should provide a mechanism that authenticates users on the basis of a unique identifier and confidential password.
42.	Ensure that the system's security/ authorization mechanism has an administrator who functions with appropriate controls.
43.	Review and evaluate the audit trails present in the system and the controls over those audit trails.
	Arrangements exist for creating back-up copies of data and programs, storing and retaining them securely, and recovering applications in the event of failure
44.	Any data and programs that have to be held on PC's / standalone systems are backed up regularly. *(Determine the backup arrangements for data processed. What short term data recovery / disaster recovery procedures exist? Is tape/disc back up media safely stored in a suitable fireproof container?)* *(Do procedures require media to be taken off site as an additional safety measure? Are responsibilities clearly defined for the backing up of important data?)*
45.	Database integrity checks are run periodically and back-up copies of the database are retained from one check to the next. (*Establish that database integrity checks are carried out, and , how are they documented?)*
46.	A general procedures or training manual for use by all staff involved with processes exists to follow in the event of an application failing during processing.

S.No.	Checklist
	(Ensure that a manual exists that details actions to be taken in the event of application failing.)
47.	Check if appropriate controls are established on data files such as Prior and Later Image, Labeling, Version and Check Digit/Parity Check.
48.	Find if appropriate controls are established on data integrity such as Domain Integrity, Relational Integrity, Entity Integrity and Referential Integrity.
49.	Ensure that appropriate backup and recovery controls are in place.
50.	Evaluate controls around the application's data retention and controls around data classification within the application.
51.	Evaluate the use of encryption techniques to protect application data.
52.	Ensure that the application software cannot be changed without going through a standard checkout/staging/testing/approval process after it is placed into production.
53.	Evaluate controls around code checkout, modification, and versioning along with testing of code (procedures/process activity) placed into the production environment.
54.	Evaluate user access levels to alter/update/read production data.
55.	Ensure that a mechanism or process has been put in place that suspends user access on termination from the company or on a change of jobs within the company.
56.	Verify that the application has appropriate password controls.
57.	Review and evaluate processes for granting access to users. Ensure that access is granted only when there is a legitimate business need.
58.	Ensure that users are automatically logged off from the application after a certain period of inactivity.

Questions

1. Transaction or files not processed to completion due to an error during an online or an offline processing or failure of a trigger is an …….. application exposure.
   a. Incomplete processing
   b. Inaccurate Information
   c. Changes to Data or Programs
   d. Unauthorized remote access

2. ……. to ensure that only complete, accurate and valid data and instructions are input to the application.
   a. Boundary Controls
   b. Input Controls
   c. Output Controls
   d. Processing Controls

3. The logging of events like user requesting access, incorrect login attempts and terminal-id at the boundary of an application are done by using ……..
   a. Cryptographic controls
   b. Transaction Trails
   c. Audit Trails
   d. Backup Trails

4. Which of the following is an input control?
   a. Login
   b. Range check
   c. Report distribution
   d. Digital signature

5. Which of the following best describes output control?
   a. Protects the confidentiality of output.
   b. Prevents data loss
   c. Ensures availability of data
   d. Ensure integrity of data while in transit

6. Which of the following is not an application Control?
   a. Input Controls
   b. Processing Controls
   c. Recovery Controls
   d. Output Controls

7. In the layout of a source document…….
   a. To avoid confusion for users keying information should not be present in the form.
   b. The form need not contain instructions.
   c. To capture data that is to be calculated on the system for verification.
   d. Data entry is to be sequenced from left to right and top to bottom.

8. ……..is the main objective to design the data-entry screens.
   a. The number of data to be collected.
   b. The approved source document of the screen

c. The experience of the data-entry operator with the screen
d. The periodic frequency of the screen being used.

9. If the product number B8597 is coded as B5987, it is an example of a
   a. Truncation error
   b. Double Transposition error
   c. Random error
   d. Transcription error

10. A data/field check edit control is
    a. A check for missing blanks during data entry.
    b. A check for the data upper and lower limit.
    c. A check for proper sequence of data.
    d. A check for data dependency.

11. Identify the best validating instruction input interface method that can avoid errors and can be controlled well:
    a. Processing language
    b. Manual entry Screen
    c. Menu Driven
    d. Command language

12. While processing a journal entry if only debit entry was updated and the credit entry was not updated due to absence of an important field the …….report gives details of the respective transaction code.
    a. Edit Report
    b. Exception Report
    c. Run-to-run Total Report
    d. Verification Report

13. The existence control over processing of data should include a ……. control to recover a process from a failure without having to repeat the entire process from the beginning.
    a. Checkpoint
    b. Backup
    c. Monitoring
    d. Validation

14. Storage of critical forms, logging programs executed for report generation and print job monitoring are ……..controls implemented in an application.
    a. Boundary
    b. Output

c. Input
d. Data file

15. Identify the controls that can ensure loading and execution of the most current data file or program.
    a. Version and internal labeling
    b. Verification and validation
    c. Value and internal blocks
    d. Internal segmentation labels

16. Which of the following is not a design guideline for using color on a data-entry screen?
    a. Use colors scarcely
    b. Use bright colors so differences are highlighted
    c. Use similar colors
    d. Do not use red for error messages

17. Identify the audit procedure that assesses the access to program libraries within an application.
    a. Verify source code compile dates; compare module size to production load module size.
    b. Observe and inspect the user responsibilities within the organization.
    c. Examine the adequacy of the backup strategies adopted.
    d. Review the system documentation of SDLC methodology followed.

18. …….is a logical batch control based on the cut off date parameter.
    a. A group of invoices for a table
    b. A set of invoices that have a similar error
    c. A set of invoices that constitute a division/branch
    d. A set invoices made by a group of users

19. ……control ensures that data resident in the files are maintained consistently with the assurance of integrity and confidentiality of the stored data.
    a. Output
    b. Input
    c. Boundary
    d. Data file

20. Identify the control that is not a factor of data-entry screen design.
    a. Caption design
    b. Colour design
    c. Tabbing design
    d. Check digit design

21. If a data entry requires that gross salary should not exceed Rs.100000/- per month this can be done by using …….control.
    a. Sequence check
    b. Range check
    c. Total check
    d. Field check

22. The ……..control check is used to prevent unauthorized access to the intermediate storage of output before printing.
    a. Ranging
    b. Table lookup
    c. Spooling
    d. Segmenting

23. If data entry is done directly on screen, the design organization should be
    a. Tabbing and skipping design is a critical factor
    b. Use asymmetry to reduce the number of screens required for data input
    c. Screen organization should be synchronized to data capture method
    d. Screen organization should be with the preferences of the data entry operator.

24. Identify the audit procedure to verify that transactions are from recognized sources:
    a. To compare the application with a valid source document timesheet.
    b. To check the password of the data-entry operators
    c. To interview the personnel maintaining rejected records
    d. To check logging of data exchange methodology

25. "Check if appropriate controls are established on data files such as Prior and Later Image, Labeling, Version and Check Digit/Parity Check." This checklist question can be used by an auditor to check……..controls.
    a. Boundary
    b. Data process
    c. Data Backup
    d Data storage

Answers :

1. a	2. b	3. c	4. b	5. a	6. c	7. d	8. b
9. b	10. a	11. c	12. b	13. a	14. b	15. a	16. b
17.a	18. c	19. d	20. d	21. b	22. c	23. c	24. a
25. c							

5 Information Assets and Their Protection

Learning Objectives

- The need to protect Information Assets
- Difference between Information Asset and Information Systems Asset
- Classification of Resources
- Classification of Users
- Access Control Models
- Information Security Policy
- Components of Policies
- Tools to Implement Policies
- Policy Implementation

Introduction

The need to protect Information Assets is a tight rope walk for every enterprise as it sets its business objectives and strives to achieve them, while managing the risks to the external and internal environment. The successful performance of business processes is significantly dependent on information technology, because it helps in handling them effectively and efficiently. Increasingly information is being recognized as a key business asset that is critical to the success and survival of today's enterprise. But this technology, including the Internet, is a double edged sword, for the benefits come along with risks. Hence it is all the more necessary that business information resources and the information technology that processes it are protected. While the former i.e. business information is generally referred to as Information Resources, the latter i.e. the business processes, the technology, the IT processes and people are referred to as Information Systems Resources.

The first major process undertaken in protecting information assets is information recording and information classification.

Information recording takes stock of the kinds of information used for business. This includes various kinds of information used in application processing on handheld and portable devices, etc.

All information in the company is classified according to its intended audience, and handled accordingly. This includes paper documents, computer data, faxes and letters, audio recordings, and any other type of information. Classifying this information and labeling it clearly helps employees understand how management expects them to handle it, and to whom they should expose it.

The next process is the classification of users. It is based on the roles played by users in the organization.

Lastly, access control models describe what kind of users can have access to what kind of data.

Information Classification

The information classification process focuses on business risks and data valuation. Not all data has the same value for an organization. Some data is valuable to the people who have to make strategic decisions, because it aids them in making long-range or short-range business direction decisions. Some data, such as trade secrets, formulas, and new product information, are valuable because its loss can create a big problem for the enterprise in the marketplace. Apart from creating public embarrassment, it can also affect its credibility.

For these reasons, information classification has an enterprise-level benefit. Some of the other benefits are:

- An organization demonstrates commitment to security protection
- Helps identify information that is t sensitive and vital to an organization
- Supports the tenets of confidentiality, integrity, and availability of to data
- Helps identify protections to different types of information
- Is required for regulatory, compliance, or legal reasons

Classification of Information Assets

Access Controls enable authorized users or processes (commonly called subjects) to gain access to information and information systems (“IS”) assets or resources (commonly called objects).

Different kinds of IS assets require different degree of access controls. Further, within an asset (say, a database), different elements or components of the asset (individual fields or records meeting a certain criteria. require more stringent controls. For example, a bank manager may have rights to modify the borrower’s database with regard to credit limit of a borrower but no access to modify cash transactions. Hence information and computer processes in an organization have different degrees of criticality.

Information systems resources require classifying or categorizing according to their sensitivity. This depends upon the risks affecting these resources and their impact resulting from exposure. Such classification makes for administrative convenience for implementing and maintaining access control systems and optimizing the cost of security.

Some kinds of information are more critical to a business than the others, such as production process information, formulas, strategic plans, and research information. In the financial services industry, for example, information assets are very close to financial assets. The loss of such information can jeopardize the existence of business or lead to loss of goodwill and trust among stakeholders and also loss of brand equity.

Compared to this, organizations may also have information which is of less consequence in the event of their loss, such as a list of customers, details of employee pay, etc. Thus, in order to ensure cost-effective controls, it is beneficial to classify the entire organizational information. It also enables fine tuning of access control mechanisms and avoids the cost of over-protecting and under protecting information.

IS resource classification also helps in determining the degree of access to be authorized to a user with regard to various classes (sensitivity) of resources. For example, access to production or live programs and data should be restricted to a limited set of users. Similarly, access to test data and programs in the development area, must be restricted to an identified group of programmers and analysts. The scheme of classification may vary according to the type of organization. However, some standard or popular classifications are also available. Data in high security organizations such as defense establishments may be classified on the following lines:

- **Top secret:** This indicates information of the highest degree of importance; any compromise of its confidentiality, integrity and availability can endanger the existence of the organization. Access to such information may be restricted to either a few named individuals in the organization or to a set of identified individuals.
- **Secret:** Information in this category is strategic to the survival of the organization. Unauthorized disclosure can cause severe damage to the organization and its stakeholders.
- **Confidential:** Information in this category also needs high levels of protection, because its unauthorized disclosure can cause significant loss or damage. Such information is highly sensitive and is to be well protected.

- **Sensitive:** Such information is more important than unclassified information. It s disclosure can have a serious impact on the organization.
- **Unclassified:** It is information that does not fall in any of the above categories. Its unauthorized disclosure will not t cause any adverse impact on the organization. Such information may also be made freely available to the public.

Another type of classification, popular in commercial organizations, can be: Public, Sensitive, Private and Confidential.

Data Privacy and Data Protection

Many advanced nations have enacted legislation concerning “Data Privacy”. That is, prohibiting say the release of medical or other information to third parties without a court order. Regulatory bodies also require restriction on dissemination of data collected by certain industries, notably banks and financial services.

The Data Protection Act, 1988, of UK, for example, has the following key features and is not limited specifically to data held electronically:

- It now applies to all personal information held in “relevant fling systems”, which may be in any medium (paper, database, spreadsheet, word-processing folder, etc.). The criteria for labeling something as an “filing system” in terms of the Act relate to whether the information is held in a structured way, and indexed by individual identifiers.
- There is an emphasis on giving data subjects advance notification about collecting data and what will be done with it (how it is to be 'processed'). In this context, data subjects must have the opportunity to give consent to the collection and processing of their data.
- That consent should be given freely, and (wherever possible) be obtained explicitly and in advance.
- There are “Fair Processing” principles. The personal data that is collected and processed must be for specified, explicit and legal purposes; it must be accurate, relevant and not exceed those purposes. Personal data must be kept secure, up-to-date and not longer than what is actually necessary.

There are strict controls on the processing of 'sensitive personal data' (i.e. race, ethnicity, gender, health), even where it is processed only for research purposes.

The Act also prescribes compliance audits.

The aims of data protection compliance audits go beyond the basic requirements of Data Security and address wider aspects of data protection including:

- Mechanisms for ensuring that information is obtained and processed fairly, lawfully and properly.
- Quality Assurance: ensuring that information is accurate, complete and up-to-date, adequate, relevant and not excessive.
- Retention: that there is appropriate weeding and deletion of information.
- Documentation on authorized use of systems, e.g. codes of practice, guidelines etc.
- Compliance with individual's rights, such as subject access.
- Compliance with the data protection legislation in the context of other pieces of legislation.

Did you know?

Privacy deals with personally identifiable information such as date of birth, address, phone no., spouse name etc. and is therefore different from confidentiality

Classification of Users

Entitlement of access to an information resource and degree of access is determined according to the job description, functions and role of the user in the organization. The "rights of access" is to ensure that the user does not gain access to undesired information resources or an undesired mode of access to the information. This is governed by the "need to know and need to do basis" or the principle of least privileges. The principle of least privilege (also known as the principle of least authority) is an important concept in information security. It advocates minimal user profile privileges on computers, based on users' job necessities. It can also be applied to processes on the computer; each system component or process should have the least authority necessary to perform its duties. This is also called the "default deny" principle. All these terms imply that access is available only to users after it has been specifically granted to them, only to perform the job function that has been granted to them.

These principles help reduce the "attack surface" of the computer by eliminating unnecessary privileges that can result in network exploits and computer compromises. At the same time, the authority for access should be adequate and not adversely affect the functioning of the user.

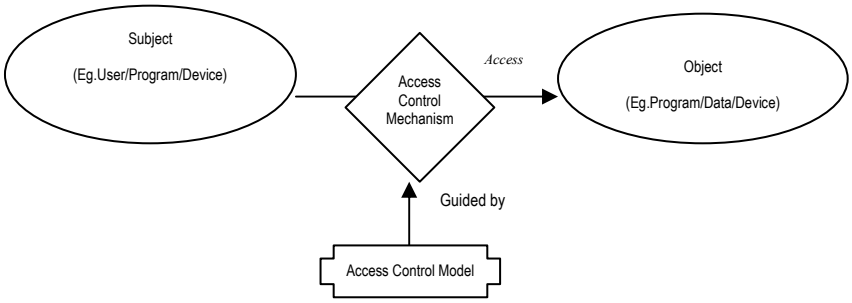
Based on these principles, users can be broadly classified as:

i. **Data Owners:** In an organization, ownership is established for every data resource. This enables assignment of accountability and responsibility for every data resource. The owner is responsible for protection of the data resource and

accountable for any loss or damage to it. . For example, the Sales Director may be the designated owner of the sales transaction database. The data owner has the rights of determining the classification of the data resource or changes thereto and rights of delegation of responsibilities to users and custodians. However, custodians may be responsible for day to day protection of the data resource, under delegated authority from owners.

ii. **Data Users:** Users require access to the data for their day to day functioning. For example, employees other than designated data owners, data entry operators, customers etc. Data users derive their rights of access from the data owners. Users are governed by the security policy and procedures of the organization and have the responsibility to use their authority for organizational purposes and protect the resources.

iii. **Data Custodians:** The custodians like the IT Department are delegated with the responsibility of administering and protecting data resources. The custodians also derive their rights from the owners and their actions are governed by the security policy framework, policies and procedures.



iv. **Naming Conventions:** The classification governs the security administration in determining which class of information can be accessed by which class of users and to what degree or type of access. For example, in a bank, the access to information about internal credit rating of borrowers contained in some of the central servers will be restricted to managers of particular departments, possibly spread geographically across the country. However, only a designated subset of these managers may have access to modify the information. For purposes of facilitating administration and efficient access control, this information would need to be identified by a specified name, e.g. “Class A” and the category of users can be classified as “Group One”. The access control policy could clearly define the access rights to each of the category of resources with regard to the user groups. Based on the granularity of access control desired, the naming conventions can be applied. These conventions simplify the access rule, which makes it easy to maintain them.

v. **Access Control Models:** One of the critical challenges in implementing access control systems is to configure the appropriate access rules when a subject seeks to access an object or objects. This is often problematic since the access control protects the IS resources and at the same time ensures that eligible persons and processes have rightful access to resources.

Managements evolve policy for systems, sub-systems, applications or group of systems. Logical access controls are a technical means of implementing policy decisions. The development of an access control policy is often challenging since it requires balancing of interests of security, operational requirements, and user friendliness.

There are many access control models. These are discussed in the chapter on Logical Access Controls.

Information Security Policy

A security policy is the statement of intent by the management about how to protect a company's information assets. It is a formal statement of the rules which give people access to an organization's technology and information assets, and which they must abide.

In its basic form, a security policy is a document that describes a company's or organization's security controls and activities. The policy does not specify technologies or specific solutions; it defines a specific set of intentions and conditions that help protect a company's assets and its ability to conduct business.

A security policy is the essential foundation for an effective and comprehensive security program. It is the primary way in which management's security concerns are translated into specific measurable and testable goals and objectives. It provides guidance to the people who build, install, and maintain computer systems. Security policies invariably include rules intended to:

- Preserve and protect information from any unauthorized modification, access or disclosure.
- Limit or eliminate potential legal liability from employees or third parties.
- Prevent waste or inappropriate use of the resources of an organization.

A security policy should be in written form. It provides instructions to employees about what kinds of behavior or resource usage are required and acceptable, and about what is forbidden and unacceptable. It gives clear instructions to IT staff and security professionals about how to restrict authority and enact access controls, authentication methods, privacy practices, and accounting techniques. A security policy also provides information to all employees about how to help protect their

employer's assets and information, and instructions regarding acceptable (and unacceptable) practices and behavior.

Definition

Security policy is defined broadly as the documentation of computer security decisions. In making these decisions, managers face hard choices involving resource allocation, competing objectives, and organizational strategy related to protecting both technical and information resources as well as guiding employee behavior.

Managers at all levels make choices that can result in policy, with the scope of the policy's applicability varying according to the scope of the manager's authority. In this chapter we use the term policy in a broad manner to encompass all the types of policy described above regardless of the level of manager who sets the particular policy. Managerial decisions on computer security issues vary greatly.

To differentiate among various kinds of policy, this chapter categorizes them into three basic types:

- **Program policy** is used to create an organization's computer security program.
- **Issue specific policy** addresses specific issues of concern to the organization.
- **System specific policy** focuses on decisions taken by the management to protect a particular system.

Familiarity with various types and components of policy will aid managers in addressing computer security issues of the organization. Effective policies result in the development and implementation of an efficient computer security program and a proper protection of systems and information.

It is not important not only to categorize organizational policies into these three categories, but also to focus on their functions.

Procedures, standards, and guidelines are used to describe how these policies will be implemented within an organization. Further, while policies are approved at the topmost level, procedures, standards and guidelines are approved at lower levels of management. What is included in a policy may vary from organization to organization. It may also be that an item covered in a policy in one organization may be documented at a procedure or standard level in another organization.

Tools to Implement Policy: Standards, Guidelines, and Procedures

Because policy is in the form of a broad general statement, organizations also develop standards, guidelines, and procedures that offer users, managers and others a clearer approach to implementing policy and meeting organizational goals.

Standards and guidelines specify technologies and methodologies to be used to secure systems. Procedures are more detailed steps to be followed to accomplish particular security related tasks. Standards, guidelines, and procedures may be promulgated throughout an organization through handbooks or manuals.

Organizational standards specify uniform use of specific technologies, parameters, or procedures whose uniform use benefits an organization. Standardization of organization-wide identification badges is a typical example, providing ease of employee mobility and automation of entry/exit systems. Standards are normally compulsory within an organization. Guidelines assist users, systems personnel, and others in effectively securing their systems. The nature of guidelines, however, immediately recognizes that systems vary considerably, and imposition of standards is not always achievable, appropriate, or cost-effective. For example, an organizational guideline may be used to help develop system-specific standard procedures. Guidelines are often used to help ensure that specific security measures are not overlooked, although they can be implemented, and correctly so, in more than one way.

Procedures normally assist in complying with applicable security policies, standards, and guidelines. They are detailed steps to be followed by users, system operations personnel, and others to accomplish a particular task (e.g., preparing new user accounts and assigning appropriate privileges). Some organizations issue overall computer security manuals, regulations, handbooks, or similar documents. These may mix policy, guidelines, standards, and procedures, since they are closely linked.

While manuals and regulations serve as important tools, it is often useful if they clearly distinguish between policy and its implementation. This can help in promoting flexibility and cost-effectiveness by offering alternative implementation approaches to achieving policy goals.

Components of a security policy

i. Introduction
ii. Purpose of Security Policy
iii. Security Policy Scope
iv. Security Policy Exemptions
v. General Security Policy
vi. Maintenance of Policies, Standards, Guidelines and Recommendations
vii. Security Policy: Review, Schedule and Updates
viii. Security Officers: Role and Responsibilities
ix. Auditing and Compliance; State Auditor's Role

Program Policy

The senior management of an organization issues program policy to establish (or restructure) its computer security program and its basic structure. This high-level policy defines the purpose of the program and its scope assigns responsibilities (to the computer security organization) for direct program implementation, as well as other responsibilities to related offices, and addresses compliance issues.

Program policy sets organizational strategic directions for security and assigns resources for its implementation.

Components of Program Policy

The following is a list of the components found in a typical Information security program policy.

i. **Purpose:** Program policy normally includes a statement describing why the program is being established. This includes defining the goals of the program. Security related needs, such as integrity, availability, and confidentiality, form the basis of organizational goals.. The primary purpose of program-level policy is to establish the IT security program. This includes defining the program management structure, reporting responsibilities, roles of individuals and groups throughout the organization, and organization-wide goals of the security program. For instance, in an organization responsible for maintaining large mission-critical databases, reduction in errors, data loss, data corruption, and recovery might be specifically stressed. In an organization responsible for maintaining confidential personal data, however, goals might emphasize stronger protection against unauthorized disclosure.

   Additionally, program-level policy should serve the purpose of impressing upon all employees the importance of IT security and clarifying the individual employee's role and responsibilities. IT security policy may be met with a degree of skepticism unless given appropriate visibility and support by the top management, and that visibility and support should be clearly and energetically reflected in the program level policy and in its emphasis on employee participation. The program level policy should thus firmly establish individual employee accountability. Employees should be made aware that even if they are not designated IT security program personnel, they nonetheless have significant IT security responsibilities.

ii. **Scope:** Program level policy should have sufficient breadth of scope to include all of the organization's IT resources, including facilities, hardware, software, information, and personnel. In some instances, it may be appropriate for a policy

to name specific assets, such as major sites, installations, and large systems. In addition to such specified assets, it is important to include an overview of all the types of IT resources for which the organization is responsible, such as workstations, Local Area Networks (LANs), standalone microcomputers, etc. In some instances, it may be appropriate for an organization's computer security program to be limited in scope.

iii. **Responsibilities:** Once the security program is established, its management is normally assigned to either a newly created or existing office. The responsibilities of officials and offices throughout the organization also need to be addressed; these include line managers, applications owners, users, data processing centre manager, and computer systems security group.

This section of the policy statement, for example, distinguishes between the responsibilities of computer services providers and those of the managers of applications using the provided services. The policy could also establish operational security offices for major systems, particularly those at high risk or most critical to organizational operations. It can also serve as the basis for establishing employee accountability. In assigning responsibilities, it is necessary to be specific; such statements as "computer security is everyone's responsibility," means that no one has a specific responsibility.

At the program level, responsibilities should be specifically assigned to those organizational elements and officials responsible for the implementation and continuity of the computer security policy. Overall, the program-level assignment of responsibilities should cover activities and personnel that are e integral to the implementation and continuity of the IT security policy.

The following are some of the personnel that are involved.

- **IT Security Program Manager: IT security** program managers are responsible for developing enterprise standards for IT security (e.g., single sign-on, remote access, user-id creation). They play a leading role in introducing an appropriate, structured methodology to help identify, evaluate, and minimize IT security risks to the organization. They coordinate and perform system risk analyses, analyze risk mitigation alternatives, and build a business case for the acquisition of appropriate IT security solutions that will enable mission accomplishment in the face of real threats. These IT security program managers also support senior management in ensuring that security management activities are conducted as required to meet the needs of the organization.
- **Chief Information Officer:** The CIO is responsible for the organization's IT planning, budgeting, investment, performance and acquisition. He provides advice and assistance to senior organization personnel in procuring the most

efficient and effective security product to fit the IT security architecture.

- **IT Investment Board:** The IT Investment Board (or its equivalent) is responsible for managing capital planning and investment control process. This Board can set the investment criteria for security product selection from qualitative and quantitative perspectives. The Board reviews benefits and risks for procuring a particular product and examines alternative approaches.
- **IT System Security Officer:** The IT System Security Officer is responsible for ensuring the security of an information system throughout its life cycle. He is required to:
    - Develop security policies and keep them up-to-date to meet changing business requirements, technology and threats.
    - Help in developing guidelines and procedures for the implementation of security policy.
    - Ensure that the risk acceptance process is followed when an exception to a security policy becomes necessary.
    - Remain current and up-to-date in technology and threats and security, so that he can meet the everyday security challenges and plans the security solutions in a better way.
    - Understand the business processes so that appropriate security mechanisms can be applied.
    - Work closely with systems administrators, security auditors, legal experts, insurance experts, etc to implement the security programs effectively.
    - Co-ordinate or assist in the investigation of security threats.
    - Define the key threats to each asset.
    - Help the management in the formulation of response to security audit.
    - Help in disaster recovery.

iv. Compliance:

Without a formal, documented information security policy it is not possible for the management to proceed with the development of enforcement standards and mechanisms. Program-level policy serves as the basis for enforcement by describing penalties and disciplinary actions that can result from failure to comply with the organization's IT security requirements. Discipline commensurate with levels and types of security infractions should be discussed.

Program policy typically will address two compliance issues:

General compliance to ensure meeting the requirements to establish a program and the responsibilities assigned therein to various organizational components. Often an oversight office is assigned responsibility for monitoring compliance, including how

well the organization is implementing management's priorities for the program. The use of specified penalties and disciplinary actions are consolidated in the policy document. Since the security policy is a high-level document, specific penalties for various infractions are normally not detailed here; instead, the policy may authorize the creation of compliance structures that include violations and specific disciplinary action(s).

Consideration should also be given to the fact that non-conformance to policy may be unintentional on the part of employees. For example, non-conformance can often be due to lack of knowledge or training. It can also be the result of inadequate explanation of the policy. For these stated reasons, it is desirable that, along with enforcement, program-level policy needs to make provisions for orientation, training, and compliance within a realistic timeframe.

Issue-Specific Policy

Whereas program policy is intended to address the broad organization-wide computer security program, issue-specific policies focus on areas of current relevance and concern to an organization.

Management may find it appropriate, for example, to issue a policy on how the organization will approach contingency planning (centralized vs. decentralized) or the use of a particular methodology for managing risk to systems. A policy could also be issued, for example, on the appropriate use of a leading-edge technology, whose security vulnerabilities are still largely unknown within the organization. Issue specific policies may also be appropriate when new issues arise, such as implementing a recently passed law requiring additional protection of particular information. Program policy is usually broad enough to require much modification over time, whereas issue-specific policies are likely to require more frequent revision with changes in technology and related factors.

Components of Issue-Specific Policy

A useful structure for issue-specific policy is to break the policy into its basic components.

i. **Issue Statement:** To formulate a policy on an issue, managers first must define the issue with any relevant terms, distinctions, and conditions. It is also often useful to specify the goal or justification for the policy which can be helpful for gaining compliance with the policy.
ii. **Statement of the Organization's Position:** Once the issue is stated and related terms and conditions are defined, this section is used to clearly state the organization's position (defined management's decision) on the issue. To

continue the previous example, this would mean stating whether the use of unofficial software is prohibited in all or some cases, whether there are further guidelines for approval and use, or whether case-by-case exceptions will be granted by whom and on what basis.

iii. **Applicability:** Issue-specific policies also need to include statements of applicability. This means clarifying where, how, when, to whom, and to what a particular policy applies. For example, it could be that the hypothetical policy on unofficial software is intended to apply only to the organization's own on-site resources and employees and not to contractors with offices at other locations. Additionally, the policy's applicability to employees travelling among different sites and/or working at home who need to transport and use disks at multiple sites might need to be clarified.

iv. **Roles and Responsibilities:** The assignment of roles and responsibilities is also usually included in issue-specific policies. For example, if the policy permits unofficial software privately owned by employees to be used at work with the appropriate approvals, then the approval authority granting such permission would need to be stated. The Policy would lay down, who, by position, has such authority. Likewise, it would need to be clarified who would be responsible for ensuring that only approved software is used on organizational computer resources and, perhaps, for monitoring users in regard to the use of unofficial software.

v. **Compliance:** For some types of policy, it may be appropriate to describe, in some detail, the infractions that are unacceptable, and the consequences of such behavior. Penalties should be explicitly stated and be consistent with organizational personnel policies and practices. When used, they should be coordinated with appropriate officials and offices and, perhaps, employee unions. It may also be desirable to task a specific office within the organization to monitor compliance.

vi. **Points of Contact:** For any issue-specific policy, the appropriate individuals in the organization to contact for further information, guidance, and compliance should be indicated. Since positions tend to change less often than the people occupying them, specific positions may be preferable as the point of contact. For example, for some issues the point of contact might be a line manager; for other issues it might be a facility manager, technical support person, system administrator, or security program representative. Using the above example once more, employees would need to know whether the point of contact for questions and procedural information would be their immediate superior, a system administrator, or a computer security official.

Areas Appropriate for Issue-specific Policies

Some of the areas in which management today needs to consider issue-specific IT security policies are covered in this section. These topics are intended to be illustrative rather than exhaustive. An organization would necessarily need to tailor its policies relating to them to meet its own unique needs.

i. **Physical security:** The physical protection of and access to IT resources and facilities will generally need to be addressed in one or more specific policies. In organizations with extensive IT systems and equipment, this may mean developing policies that address issues like who has access to what sites/locations; how often risks to installations are be analyzed and by whom; what types of physical access controls and monitoring equipment should be put in place; what responsibilities will be assigned to trained security officials and what activities and responsibilities will be required of all employees.

ii. **Personnel Security:** Depending on the types of activities being performed, the degree of data sensitivity to be encountered, and the number of personnel involved, specific security policies related to personnel screening, requirements, hiring, training, evaluating, and firing may need to be developed and administered. It may be appropriate that trained personnel security specialists initiate, review, approve, and perform all security-related personnel actions.

iii. **Communications Security:** Communications security is a complex technical specialty. . In organizations where day-to-day business relies on communicating routinely with remote locations, the security of the communications transmissions and lines is usually an issue that needs to be addressed according to a stated policy. If the data being transmitted is highly sensitive, then this concern is magnified and issue-specific security policies may need to be developed for a number of activities. Issues associated with the use of cryptography and its related options and procedures, the use of modems and dial-in lines, and precautions against wiretapping are some of the issues to be addressed.

iv. **Administrative Security:** Administrative security as it applies to IT system management and oversight activities comprises many potential security policy issues. It includes such topics as input/output controls, training and awareness, security certification/accreditation, incident reporting, system configurations and change controls, and system documentation.

v. **Risk Management**: Risk management involves assessing IT resources in terms of potential threats and vulnerabilities and planning the means for counteracting them. Issues that will need to be addressed by policies include how, by whom, and when the assessments should be performed; and what type of documentation should result.

vi. **Contingency Planning:** Contingency Planning means planning for the emergency actions required in the event of damage, failure, and/or other disabling events that could occur to systems. Issues that need to be addressed by policies include determining which systems are most critical and therefore of highest priority in contingency planning; how the plans will be tested, how often, and by whom; and who will be responsible for approving plans.

Examples of Issue-Specific Policies

Identity Management Policies

i. **Account/Password Authentication:** A unique account and password combination must authenticate all users of information systems. The account name must be used only by a single individual, and the password must be known only to that individual.
ii. **Account Changes:** The manager responsible for the end user must request changes in access privileges for corporate information systems for a system account. End users may not request access-privilege changes to their own accounts. The request must be recorded and logged for the record.
iii. **New Account Requests:** The manager responsible for a new end user must request access to corporate information systems through a new account. End users may not request their own accounts. The new account request must be recorded and logged for the record and disabled when no longer needed.
iv. **Login Message:** All computer systems that connect to the network must display a message before connecting the user to the network. Its intent is to remind users that information stored on information systems belongs to the company and is not private or personal. The message must also direct users to the corporate information system usage policy for more detailed information. It must state that by logging on, the user agrees to abide by the terms of the usage policy. Continuing to use the system indicates the user's willingness to adhere to the policy.
v. **Failed Login Account Disabling:** After five successive failed login attempts, a system account must be automatically disabled to reduce the risk of unauthorized access. Any legitimate user whose account has been disabled in this manner can have it reactivated by providing both proof of identity and management approval..
vi. **Employee Account Lifetime:** Permanent employee system accounts will remain valid for a period of 12 months, unless otherwise requested by the employee's manager. The maximum limit on the requested lifetime of the account is 24 months. After the lifetime of the account has expired, it can be reactivated for the same length of time upon presentation of both proof of identity and management approval.

Network Policies

i. **Extranet Connection Access Control:** All extranet connections to and from other companies' networks outside of the corporation, either originating from the external company's remote network into the internal network, or originating from the internal network going out to the external company's remote network must limit external access to only those services that are authorized for the remote company. This access control must be enforced by IP address and TCP/ UDP port filtering on the network equipment used to establish the basic network connection.
ii. **System Communication Ports:** Systems communicating with other systems on the local network must be restricted only to authorized communication ports. The ports must be blocked by firewalls or router filters.
iii. **Unauthorized Internet Access Blocking:** All users must be automatically blocked from accessing Internet sites that are inappropriate for company use. This restriction must be enforced by automated software, which has to be updated regularly.
iv. **Virtual Private Network and Dial-Up:** All remote access to the corporate network is to be provided by virtual private network (VPN) or direct landline connections. Dial-up access to the corporate network is not allowed.

Data Privacy Policies

i. **Copyright Notice:** All information owned by the company and considered intellectual property, whether written, printed, or stored as data, must be labeled with a copyright notice in the following format: Copyright © 2003 [Company Name], Inc. All Rights Reserved.
ii. **E-mail Monitoring:** All e-mail must be monitored for the following activity: Employees using the company mail account for personal use giving inflammatory, unethical, or illegal content disclosure of company confidential information file attachments or messages.
iii. **Customer Information Sharing:** Corporate customer information may not be shared with outside companies or individuals.
iv. **Encryption of Data Backups:** All data backups must be encrypted.
v. **Encryption of Extranet Connection:** All extranet connections must use encryption to protect the privacy of the information traversing the network.
vi. **Data Access:** Access to corporate information, hard copy, and electronic data is restricted to individuals with a need to know for a legitimate business reason. Each individual is granted access only to those corporate information resources that are required by them to perform their job.

Data Integrity Policies

i. **Virus-Signature Updating:** Virus signatures must be updated immediately after they are made available by the vendor.
ii. **Software Testing:** All software must be tested in a suitable test environment before installation on production systems.
iii. **Division of Environments:** The division of environments into Development, Test, and Production is required for critical systems.
iv. **Version Zero Software:** Version zero software (1.0, 2.0, and so on) must be avoided whenever possible to avoid undiscovered bugs.
v. **Offsite Backup Storage:** Backups older than one month must be sent offsite for permanent storage.
vi. **Quarter-End and Year-End Backups:** Quarter-end and year-end backups must be done separately from the normal schedule, for accounting purposes.
vii. **Disaster Recovery:** A comprehensive disaster-recovery plan must be used to ensure continuity of corporate business in the event of an outage.

System Administration Policies

i. **Application Monitoring:** All servers containing applications designated for monitoring must be constantly monitored during the hours the application operates. The following activities must be monitored:

- Application up/down status
- Resource usage
- Non-standard behavior of application
- Addition or change of the version, or application of software patches

ii. **Intrusion-Detection Monitoring:** All critical servers must be constantly monitored at all times for intrusion detection. This monitoring must cover at least the following categories:

- Port scans and attempts to discover active services
- Non-standard application connections
- Non-standard application behavior
- Multiple applications
- Sequential activation of multiple applications
- Multiple failed system login attempts

iii. **Firewall Monitoring:** All firewalls must be constantly monitored, 24x7, by trained security analysts. This monitoring must include at least the following activities:

- Penetration detection (on the firewall)
- Attack detection (through the firewall)
- Denial of service detection
- Virus detection
- Attack prediction
- Penetration blocking
- Attack blocking
- Virus blocking
- Intrusion response

iv. **Network Security Monitoring:** All internal and external networks must be constantly monitored, 24x7, by trained security analysts. This monitoring must involve at least the following activities:

- Unauthorized access attempts on firewalls, systems, and network devices
- Port scanning
- System intrusion originating from a protected system behind a firewall
- System intrusion originating from outside the firewall
- Network intrusion
- Unauthorized modem dial-in usage
- Unauthorized modem dial-out usage
- Denial of services
- Correlation between events on the internal network and the Internet

Usage Policies

i. **Personal Use of Information Systems:** Personal use of company computer systems may be allowed on a limited basis to employees provided that it does not interfere with the company's business, expose the company to liability or damage, compromise the company's intellectual property, or violate any laws. Employees should be advised that the company may at any time be required by law to print or copy files, e-mail, hard copy, or backups and provide this information to the government or law enforcement agencies.

ii. **Personal Use of Telephones:** Corporate phone systems may be used for limited, local, personal calls, as long as this usage does not interfere with the performance of the corporate business.

Physical Security Policies

i. **Room Access Based on Job Function:** Room access must be restricted based on employee job function.
ii. **Position of Computer Monitors:** Computer monitors must be faced away from windows to discourage "eavesdropping."
iii. **Badges on Company Premises:** All corporate employees on the controlled premises must display badges with picture identification in plain view.

System-Specific Policy

Program policy and issue-specific policy both address policy from a broad perspective, usually encompassing the entire organization. However, they do not provide sufficient information or direction that can be used , for example, for establishing an access control list or training users on what actions are permitted. System specific policy fills this need. It is very focused, since it addresses only one system.

Many security policy decisions may apply only at the system level and vary from system to system within the same organization. While these decisions may appear to be too detailed to be policy, they are extremely important, for their significant impact on system usage and security.

These types of decisions can be made by a management official and not by a technical system administrator. The impact of these decisions, however, is often analyzed by technical system administrators. To develop a cohesive and comprehensive set of security policies, officials may use a management process that derives security rules from security goals. It is helpful to consider a two-level model for system security policy: security objectives and operational security rules, which together comprise the system specific policy. Closely linked and often difficult to distinguish, however, is the implementation of the policy in technology.

Security Objectives

The first step in the management process is to define security objectives for the specific system. Although, this process may start with an analysis of the need for integrity, availability, and confidentiality, it should not stop there. A security objective needs to be specific, concrete and well defined. It also should be stated so that it is clear that the objective is achievable. This process will also draw upon other applicable organization policies. Security objectives consist of a series of statements that describe meaningful actions about explicit resources. These objectives should be based on system functional or business requirements, but should state the security actions that support the organizational requirements.

Development of system-specific policy will require management to make tradeoffs, since it is unlikely that all desired security objectives can be fully met. Management will face cost, operational, technical, and other constraints.

Operational Security Rules

After management determines the security objectives, the rules for operating a system can be laid out to define authorized and unauthorized modification. The rules specify who by job category, organization placement can do what (e.g., modify, delete) to which specific classes and records of data, and under what conditions.

The degree of specificity needed for operational security rules varies greatly. Generally, the more detailed the rules are, the easier it is to know when one has been violated. It is also easier to automate policy enforcement. However, inordinately detailed rules may make the job of instructing a computer to implement them difficult or computationally complex.

In addition to deciding the level of detail, the management should decide the degree of formality for documenting a system-specific policy. Once again, the more formal the documentation, the easier it is to enforce and to follow policy. On the other hand, a too detailed policy can also prove an administrative burden. In general, good practice suggests a reasonably detailed formal statement of the access privileges for a system. Documenting access controls policy will make it substantially easier to follow and to enforce. Another area that normally requires a detailed and formal statement is the assignment of security responsibilities. Other areas that should be addressed are the rules for system usage and consequences of non-compliance.

Policy decisions in other areas of security are often documented in the risk analysis, accreditation statements, or procedural manuals. However, any controversial, atypical, or uncommon policies will also need formal statements. Typical policies are required for areas where the system policy is different from organizational policy or digression from normal practice within the organization. The documentation for a typical policy contains a statement explaining the reason for deviation from the organization's standard policy.

Example of System-Specific Policies: Payroll System

i. **Security Objectives:** Only individuals in the accounting and personnel departments are authorized to provide or modify information used in payroll processing.
ii. **Operational Security Rule:** Personnel clerks may update fields for weekly attendance, charges to annual leave, employee addresses, and telephone numbers. Personnel specialists may update salary information. No employees should be allowed to update their own records.

Policy Implementation

A Policy implementation is a process. It cannot merely be pronounced by the upper management as a one-time statement or a directive with high expectations of it being readily accepted and acted upon. The implementation process actually begins with the formal issuance of the policy.

Policy Documentation

Once a security policy has been approved and issued, it may be initially publicized through memorandums, presentations, staff meetings, or any other means, which may be incorporated into the formal policy documentation. The policy documentation needs to be updated with feedbacks from the organization stakeholders.

i. **Existing Documentation:** Information security will need to be integrated into many existing activities and practices at many levels of the organization. This integration will be facilitated by revising any existing applicable documentation to reflect new procedures, rules, and requirements. Included may be the modification of various existing documents, forms, and plans at all levels of the organization.

   For example, if IT equipment purchases and/or upgrades have been reviewed and approved based on documented criteria such as cost, productivity, maintainability, etc., then security considerations may need to be introduced into that criteria. Also, if it has previously been the documented policy to review the progress and status of internal IT systems under development, then security-related concerns should be introduced into that review process.

ii. **New Documentation:** Additionally, the development of many new documents, such as guidelines, standards, and procedures, may be required.

This is often true of large organizations performing different activities and having many levels of management. In such environments, different functional elements may have widely differing IT systems and needs to accommodate. It is therefore generally more practical to tailor the policy to meet their needs. This can be accomplished through the development of documents containing detailed procedures and practices for specific kinds of systems and activities within the functional elements.

For example, organizations will want to issue policies to decrease the likelihood of data loss due to technology failures and/or operator errors. A program-level policy will state something like “It is the policy of the organization to ensure against data loss due to accidents or mishaps.” In an area where extensive writing and editing of lengthy documents is performed, such as a word processing or technical publications unit, security documentation might be developed on saving work in-progress much

more often than would usually be done, and/or utilizing automatic "save" features on IT systems and software. In a different type of functional area, however, where, for example, databases are maintained that do not undergo significant changes very often, the security documentation might focus on procedures for the database administrator to use in performing periodic (daily, weekly, etc.) backups of the system.

Policy Visibility

Polices are generally public documents. However, in many cases, parts of the policies may be kept private. High visibility should be given to the formal issuance of the information security policy. This is due to a combination of factors, including the following:

- Nearly all employees at all levels are affected;
- Major organizational resources are being addressed;

Many new terms, procedures, and activities are introduced. Information security policy should be provided visibility through management presentations, discussions, question/answer forums, and newsletters. Including information security as a regular topic at staff meetings at all levels of the organization can also be a helpful tactic. As an aspect of providing visibility for security policies, information should also be included regarding the applicable high level directives and requirements to which the organization is responding. Educating employees about requirements specified by various laws and regulations will help emphasize the significance and timeliness of computer security, and it will help provide a rational basis for the introduction of information security policies.

Information security policy should be given visibility by using all applicable documentation. A more integral security policy shall cover all aspects of daily routines with the associated actions and practices and will become natural to doing business. Ultimately, among the goals of policy are the assimilation of a common body of knowledge and values and the demonstration of appropriate corresponding behavior.. Those goals will be expedited by making the information security policy integral to the organization through all avenues.

System-Specify Policy Implementation

Technology plays an important but not the sole role in enforcing system-specific policies. When technology is used to enforce policy, it is important not to neglect non-technology-based methods. For example, technical system-based controls could be used to limit the printing of confidential reports to a particular printer. However, corresponding physical security measures would also have to be in place to limit access to the printer output. .

Frequently used technical methods to implement system-security policy are likely to include the use of logical access controls. But there are other automated means of enforcing or supporting security policy that typically supplement logical access controls. For example, technology can be used to block telephone users from calling certain numbers.

Intrusion detection software can alert system administrators to suspicious activity or take action to stop its activity. Personal computers can be configured to prevent booting from a floppy disk. Technology based enforcement of system security policy has both advantages and disadvantages. A computer system, properly designed, programmed, installed, configured, and maintained, consistently enforces policy within the computer system, although no computer can force users to follow all procedures. Management controls also play an important role and should not be neglected. In addition, deviations from the policy may sometimes be necessary and appropriate; such deviations may be difficult to implement easily with technical controls. This situation crops up if the security policy is implemented too rigidly.

Interdependencies

Information Security Policy is related to many other areas:

Program Management Policy is used to establish an organization's computer security program, and is therefore closely tied to program management and administration. Both program and system specific policy may be established in any area. For example, an organization may wish to have a consistent approach to incident handling for all its systems and issue appropriate program policy to do so. On the other hand, it may decide that its applications are sufficiently independent of each other and that application managers should deal with incidents on an individual basis.

Access Controls System specific policy is often implemented through the use of access controls. For example, it may be a policy decision that only two individuals in an organization are authorized to run a check-printing program. Access controls are used by the system to implement this policy.

Other Areas Information security policy can be related to nearly every topic covered in this course material. . This is because all the topics discussed in the material have associated issues that organizations may need to address through policies. The topics most directly related, however, are: security program management and administration; personnel; security training and awareness; contingency planning; and physical and environmental security.

Awareness, Training and Education

People are usually recognized as one of the weakest links for securing systems. The

purpose of computer security awareness, training, and education is to enhance security by:

- improving awareness of the need to protect system resources;
- developing skills and knowledge of the computer users so that they can perform their jobs securely; and
- building the required in-depth knowledge to design, implement, and operate security programs for organizations and systems.
- Making computer system users aware of their security responsibilities and teaching those correct practices to help users change their behavior.

It also supports individual accountability, which is important for improving computer security. Without knowing the necessary security measures and how to use them, users cannot be truly accountable for their actions.

Cost Considerations

A number of potential costs are associated with developing and implementing computer security policies. Overall, the major cost of policy is the cost of implementing the policy and its impact upon the organization. Establishing an information security program, through a properly framed policy, does not come at a negligible cost.

Other costs may be those incurred through the policy development process. Numerous administrative and management activities may be required for drafting, reviewing, coordinating, clearing, disseminating, and publicizing policies. In many organizations, successful policy implementation may require additional staffing and training and can take time. In general, the cost to an organization for computer security policy development and its implementation will depend upon how extensive is the change needed to achieve a level of risk acceptable to the management.

Audit of IS Security Policy

Standards such as ISO 17799 provide benchmarks for reviewing security policies. Security policies are reviewed for three broad purposes: benchmarking against external standards, changes in the risk perception, and implementation of new information system within the firm.

Summary

The information asset is the organizational data and the mediums and devices of storage, processing and communications constitute the information system asset. Inadequate protection to either information asset or the information system assets may result in computer crimes.

Information systems resources are required to be classified or categorized according to their sensitivity. This depends upon the risks affecting such resources and their impact resulting from the exposure. The classifications help the security administration to determine which class of information can be accessed by which class of users and to what degree of access. The scheme of classification may vary depending on the type of organization. In commercial organizations, data may be classified as public, sensitive, private or confidential. Once users and resources are classified, it then becomes possible to specify which users can access what data. The two broad frameworks available are the mandatory access control and discretionary access control.

The term "information security policy" generally refers to important information asset and computer security related decisions taken by the management. There are three basic types of policies: Program policy, Issue-specific policies, and system-specific policies. A computer security program policy is a strategic document used to create an organization's computer security program. Issue-specific policies, as the name implies, addresses specific issues of concern to the organization. System specific policies focus on decisions taken by the management to protect a particular system. Since the policy is written at a high level, most organizations also develop standards, guidelines and procedures that offer users, managers, and others the details necessary to implement the policy. Policies and other supporting documents are highly structured documents that have a format that is usually consistent organization-wide. Policies are ineffective unless implemented. Apart from documenting the policy, management must undertake a training exercise in order to inculcate desirable security procedures and practices. Technology plays a significant role in enforcing policies, but non-technical methods are also effective in many situations.

Few examples of Security Policy Control Techniques and their Suggested Audit Procedures

Control Activities	Control Techniques	Audit Procedures
The security policy is adequately documented, approved and updated.	The key elements of the security policy are documented and include the following: - Periodic risk assessments - Document policies and procedures.	Review the documentation of the enterprise-wide security policy and discuss the key security management issues with the management and staff.

	- A hierarchical security plan covering all levels of access to IS assets. - Security awareness training - Management level testing and evaluation. - Corrective actions and incident procedures and emergency plans. To document an enterprise-wide structure of the security management with details of authority, expertise and resources.	Check audit evidence and its successful implementation. Audit security policies, budget documentation and security structure of the organization.
IS security responsibilities are clearly assigned.	The security program identifies those responsible for managing access to IS resources and their expected behavior. The resource owners and users are identified and responsibilities of senior management, security administrators and staff are specified. The policy is to cover all major IS facilities, applications; weakness and key affected stakeholders of the system.	Review the security structure of responsibilities implemented and documented within the organization. Determine if the policies reflect the conditions documented. Review the relevant security plans and critical control points.
Risk assessments and sustaining activities are	IS assets are categorized and based on the potential impact of loss of	Check the consistency of the system with the tested results and cross

thoroughly conducted.	confidentiality, integrity or availability would have on the operations on these assets. Risks are to be periodically assessed for application changes, system configuration changes and documented, based on their security categorizations.	check the critical control points and determine if risk assessments are up-to-date. Review the criteria of risk assessments and compliance with the best practices or guidelines.
Awareness of security policies.	Continuous security awareness briefings and training that is monitored for all stakeholders of the system. The security policies are to be distributed to all stake holders involved in the access to IS assets or application.	Observe security briefing and interview data owners, system administrators and users. A selected number of users are examined with respect to their security awareness, mechanisms by attempting to talk to them as a network staff and try to make them reveal their password.
Employee hiring, transfer, termination and performance policies addressing security.	Periodic investigation of regulations implemented in the preview of authorization to information assets within the organization. To have appropriate transfer, termination procedure consisting of: - Return of property keys,	Inspect investigation policies with respect to sensitive positions, confidentiality or security agreements. Review compliance with security policies and compare with system generated list of active

	identification cards etc. - Notification to security managers of terminations to prompt revocation of IDs and passwords. - staff compliance with security policies	users. Review training record and documentations on the level of awareness and subsequent monitoring procedures.
Information security weakness is identified and corrective action is taken.	Management initiates prompt action to correct weakness and documents action plans and milestones. Corrective actions are tested and monitored after implementation.	This part of the audit is to identify the accuracy of the information of the IS control weakness and the scope of the corrective actions. Check that remedial actions are being tested and monitored.
Vendor activities are secure, documented and monitored.	The activities of third party vendors are monitored with the help of: - Clearances. - confidentiality agreements. - security roles and responsibilities documented. - Corrective/preventive actions. - monitoring/audit procedures and reporting - security awareness and training.	Audit the controls and contract documents that determine the enterprise –wide application level access and responsibilities. Inspect the risks identified and managed or controlled for systems operated or accessed by third party staff.

Master Checklist to AUDIT IS Security Policy.

Sr. No.	Check points
1.	Security Department and Legal Department have been formed to provide appropriate direction to formulate, implement, monitor and maintain IS security in the organization.
2.	Whether a well-documented security policy is available.
3.	Whether an Inventory of IT assets is a part of the policy.
4.	Whether policies related to IT activities are listed in the security policy.
5.	Whether the policy takes into account the business strategy for the next 3 - 5 years.
6.	Whether the policy takes into account the legal requirements,
7.	Whether the policy takes into account the regulatory requirements.
8.	Whether the policy is approved and adopted by the Board of Directors / Senior Management,
9.	Whether the policy is communicated to all concerned and is understood by them.
10.	Whether the security awareness is created not only in IS function but also across the organization.
11.	Whether the following major security areas are covered in the policy.
12.	PC and LAN, MAN and WAN security.
13.	Whether disciplinary procedures for non-compliance to IS policies are communicated to the employees. (Violations of security policies and remedial action taken should be reviewed).
14.	Whether the physical security of IS establishments is continually addressed.
15.	Handling of security incidents
16.	Handling of confidential information

17.	Whether critical assets are identified and appropriate security is available.
18.	Privacy related issues for outside entities
19.	E-mail security
20.	Application security
21.	Interface Security
22.	Password Security
23.	Operating system security, web site security
24.	Database security
25.	Antivirus and piracy policy
26.	Archived and Backed up data security
27.	Procedures for handling incidence of security breach
28.	Whether the Incident Management Policy is documented. (Incident Management policy and procedures should be established to ensure an appropriate, effective and timely response to security incidents)
29.	Disaster Recovery Plan
30.	Use of cryptology and related security
31.	Whether a formally designed security framework is in place which covers - Internal security - Non-repudiation of transactions - Confidentiality of information - Positive customer authentication - Application development - End-user (browser security standards and settings.
32.	Persons responsible for implementing security policy and consequence for willful violation of the Security Policy
33.	Whether a review process is in place for reviewing the policy at periodic intervals and / or on any other major event

Questions

1. The information resources like paper documents, computer data, faxes and letters are business information assets which are to be protected. The first major step/process in protecting these assets is ……..
   a. information policy
   b. information stocking
   c. information controlling
   d. information classification

2. Standard scheme of classification of data in a commercial organization is……..
   a. Private, Secret, Confidential and Unclassified
   b. Public, Sensitive, Private and Confidential
   c. Top Secret, Secret, Public and Unclassified
   d. Public, Secret, Internet and Intranet

3. When data is stored/displayed/exchanged through any medium in a structured manner and indexed by individual identifiers, it is called a ……system.
   a. Relevant
   b. Filed
   c. Fling
   d. Structured

4. Quality Assurance, Retention, Documentation, individual rights and fair process of data along with data security come within the scope of data ……..
   a. Process Audit
   b. Compliance Audit
   c. System Audit
   d. Tax Audit

5. The rights of access to information authorized to users on the need to know and need to do basis is called the principle of ……..
   a. Need privileges
   b. Full privileges
   c. Least process
   d. Least privileges

6. The personnel within the organization delegated with the responsibility of administration and protection of data resources are called ……..
   a. Data protectors
   b. Data custodians
   c. Data committers
   d. Data owners

7. ……..is the statement of intent and conditions set up by the management about how to protect the company's information assets.
   a. Data policy
   b. Access policy
   c. Security policy
   d. Process policy

8. Identify the components of Information security program policy.
   a. Program, data access, controls and scope
   b. Purpose, scope, responsibilities and compliance.
   c. Purpose, scope, storage and coverage.
   d. Program security, Program scope, Program execution and results.

9. The…….component of the issue-specific policy states where, how, when, to whom and to what a particular policy applies.
   a. Issue statement
   b. Roles and Responsibilities
   c. Applicability
   d. Point of contact

10. …….is a issue specific security policy which deals with input/output controls, incident reporting, system configurations and change controls and system documentation.
    a. Communication Security
    b. Personnel Security
    c. Physical security
    d. Administrative security

11. The password authentication, account changes, employee account lifetime, login message are components of the ……..issue specific policy.
    a. Identity Management
    b. Information Management
    c. Network Management
    d. Data Management

12. Virus signature updating is the component of the …….policy.
    a. Network integrity policy
    b. Data integrity policy
    c. Data privacy policy
    d. Administrative policy

13. Copyright notice, encryption of data backups, customer information sharing and e-mail monitoring are components of the issue specific …….policy.
    a. Data integrity
    b. Data storage
    c. Data privacy
    d. Data exchange

14. ……..is a component of the identity management security policy that sets a maximum limit of 24 months validity for a user account.
    a. Login account lifetime
    b. Employee access account
    c. Access Account lifetime
    d. Employee Account lifetime

15. Extranet connection access control, restricted communication ports and blocking of unauthorized internet access are components of the …….. security policy.
    a. Personnel policy
    b. Logic policy
    c. Network policy
    d. System policy

16. Prevention of unauthorized access on a network, Intrusion response, penetration blocking are activities of …….
    a. Intrusion-Detection Monitoring
    b. Firewall Monitoring
    c. Signal Monitoring
    d. Port Monitoring

17. The process of documenting policies usually requires updating the…… and also creating ……..
    a. Existing documentation , New documentation
    b. Existing programs , New programs
    c. Existing personnel, Old personnel
    d. Networks documentation ,communication documentation

18. The control activity of observing security briefings along with interview of data owners, system administrators and users is an audit procedure to check the..……. of the security policy.
    a. Documentation
    b. Risk Assessment
    c. Awareness
    d. Responsibilities

19. The security control audit procedures ……..and ……. are used to monitor vendor activities.
    a. monitoring clearances, confidentiality agreements
    b. monitoring systems, monitoring documents
    c. monitoring personnel, monitoring data
    d. monitoring network, monitoring communication

20. …….monitoring covers multiple failed logins, activation of multiple applications, port scans and active services in the critical servers of an organization.
    a. Firewall
    b. Application-detection
    c. Network-detection
    d. Intrusion-detection

Answers :

1. D	2. B	3. C	4. B	5. D
6. B	7. C	8. B	9. C	10. D
11. A	12. B	13. C	14. D	15.C
16. B	17. A	18. C	19. A	20. D

Sources:

[1]. Ron Weber, Information Systems Audit Control, Pearson Education Inc, 1999.

[2]. Chris Davis, Mike Schiller, Kevin WheelerIT Auditing: Using Controls to Protect Information Assets (Paperback) McGraw-Hill Osborne Media; 1st edition, 2006.

[3]. Sandra Senft, Frederick Gallegos, Information Technology Control and Audit, Third Edition (Hardcover) AUERBACH; 3 edition , 2008.

[4]. Committee On Information Technology, Technical Guide On IS Audit, ICAI,Second Edition, 2009.

[5]. Government Accountability Office (GAO), Federal Information System Controls Audit Manual (Fiscam), Exposure Draft, 2008.

Module – III

System Development Life Cycle & Application Systems

1 Business Application Development Framework

Learning Goals

A clear understanding of:

- The concept of a system and its characteristics,
- The need for Structured System Development,
- The various phases of Software Development Life Cycle - SDLC and their interrelationship in brief,
- Feasibility Study,
- System Requirements Analysis, and
- Hardware and Software Acquisition.

System

A system is an interrelated set of elements that function as an integrated whole. The concept of an 'integrated whole' can also be stated in terms of a system embodying a set of relationships which are differentiated from relationships of the set to other elements, and from relationships between an element of the set and elements not a part of the relational regime. Thus, a system is composed of parts and subparts in orderly arrangement according to some scheme or plan. For example; human body is a system, consisting of various parts such as head heart, hands, legs and so on. The various body parts are related by means of connecting networks of blood vessels and nerves and the system has a main goal of "living".

A business is also a system where economic resources such as people, money, material, machines, etc are transformed by various organizational processes (such as production, marketing, finance etc.) into goods and services. A computer based information system is also a system which is a collection of people, hardware, software, data and procedures that interact to provide timely information to authorized people who need it. Each of these can be further divided into its sub-systems and this process can go on till we decide to stop with respect to our study context. Just like living animals, business systems also have a life span. After this life span is over, the

systems will have to be retired and be replaced with new systems. Some of the reasons for systems having a life span are listed below:

- Technology may become outdated e.g. writing instruments evolved from ink pen to ball point pen to gel pen and so on.
- People using the system may change e.g. new generation people may not be exposed to old technology and therefore systems will have to undergo change.
- Government or other regulatory change may render the systems obsolete.
- Business needs are expanded due to expansion of business, mergers, take-overs etc.

Characteristics of System

A system has mainly nine characteristics (as shown in Fig. 1.1), which are given as follows:

1. **Components:** A system consists of several components. A component is either an irreducible part or an aggregate of parts, also called a 'subsystem'. For example: in any automobile system, we can repair or upgrade the system by changing individual components without changing the entire system.

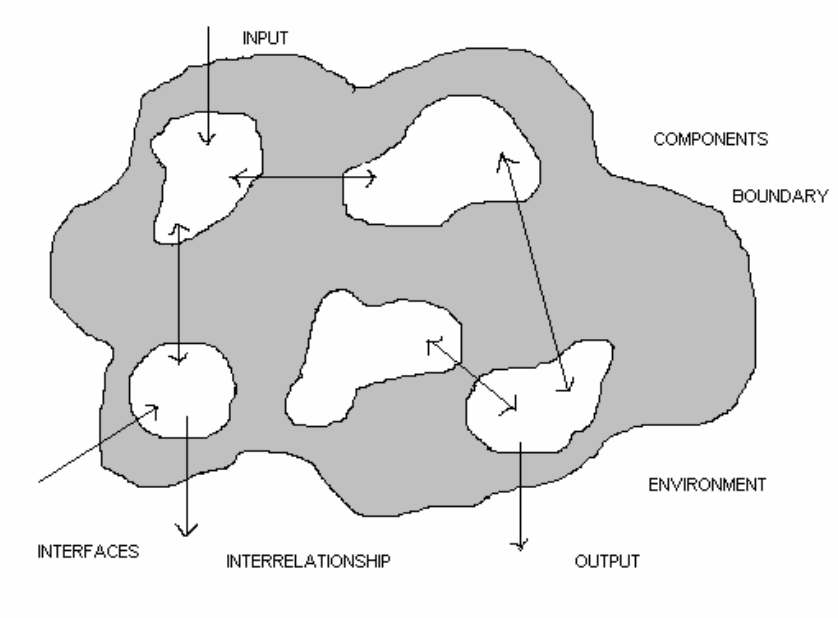


Fig. 1.1: Characteristics of a System

2. **Interconnected Components:** The components of a system become interconnected; that means the function of one component is somehow tied to the functions of the others.
3. **Boundary:** A system has a boundary; within which all of its components are contained and this established the limitations of s system, separating it from other systems. Components of a system within the boundary can be changed whereas systems outside the boundary can not be changed. These boundaries are of two types; Physical and Logical.
4. **Purpose:** In a system, all of the components work together to achieve overall purpose; which is the main reason for the existence of a system.
5. **Environment:** A system exists within an environment; everything outside the system's boundary that influences the system.
6. **Interfaces:** The points, at which the system meets its environment, are called interfaces. Similarly, interfaces also exist between subsystems.
7. **Input:** A system takes input from its environment in order to function.
8. **Output:** Finally, a system returns output to its environment as a result of its functioning, and hence achieves its purpose.
9. **Constraints:** A system must have some limitations e.g. capacity, speed etc., which are termed as constraints in its functioning. Some of the constraints are imposed inside the system, and others are imposed by the environment. Some people refer constraints as business rules for the system so that it functions, as desired.

For any system, an IS auditor must understand all the above listed features but especially the business rules will enable auditor to prepare audit scope, audit plan and audit tests etc.

Types of System

As shown in Fig. 1.2, we can distinguish systems on the basis of following parameters:

i. Elements
ii. Interactive behavior
iii. Degree of human intervention
iv. Working/Output

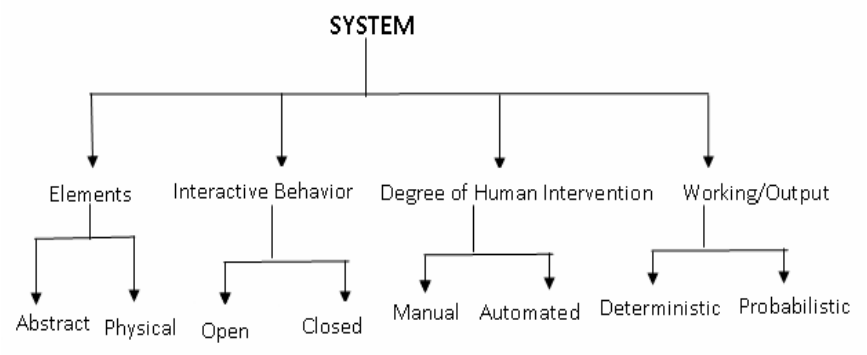


Fig. 1.2: Classification of Systems

According to Elements

i. **Abstract System:** It is also known as Conceptual Systems or Models and can be defined as an orderly arrangement of interdependent ideas or constructs. For example, a system of theology is an orderly arrangement of ideas about GOD and the relationship of humans to GOD.

ii. **Physical System**: A physical system is a set of tangible elements which operate together to accomplish an objective. Some of its examples are shown in Table 1.1.

Physical System	Description
Circulatory system	The heart and blood vessels which move blood through the body.
Transportation system	The personnel, machines, and organizations which transport goods.
Weapons system	The equipment, procedures, and personnel which make it possible to use a weapon.
School system	The buildings, teachers, administrators, and textbooks that function together to provide education to students.
Computer system	The equipment which function, together to accomplish computer processing.

Table 1.1: Few Examples of Physical System

The examples illustrate that a system is not a randomly assembled set of elements; it consists of elements, which can be identified as belonging together because of a common purpose, goal, or objective. Physical systems are more

than conceptual construct; they display activity or behavior. The parts interact to achieve an objective.

According to Interactive Behavior

A system may be composed of a number of components that work together in a cascade to achieve a goal for which the system is designed. All systems work in a specific environment and based on how they perform within an environment, systems can be categorized in two classes:

i. **Open System –** A system that interacts freely with its environment by taking input and returning output is termed as an open system. With change of environment an open system also changes to match itself with the environment. For example, the education system or any business process system will quickly change when the environment changes. To do this an open system will interact with elements that exist and influence from outside the boundary of the system.

   Information systems are open systems because they accept inputs from environment and sends outputs to environment. Also with change of environmental conditions they adapt themselves to match the changes.

ii. **Closed System –** A system that does not interact with the environment nor changes with the change in environment is termed as a closed system. Such systems are insulated from the environment and are not affected with the changes in environment. Closed systems are rare in business area but often available in physical systems that we use in our day to work. For example, consider a 'throw-away' type sealed digital watch, which is a system, composed of a number of components that work in a cooperative fashion designed to perform some specific task. This watch is a closed system as it is completely isolated from its environment for its operation. Such closed systems will finally run down or become disorganized. This movement to disorder is termed on increase in entropy.

   Organizations are considered to be relatively open systems. They continuously interact with the external environment, by processes or transformation of inputs into useful output. However, organization behaves as a relatively closed system in certain respects so as to preserve their identity and autonomy. They may ignore many opportunities so as to maintain their core-competence. Organizations are open systems, because they are input-output systems. The input consists of finance, physical & mental labor and raw material.

Organizations perform several operations on these inputs and process out products or services. The process of exchange generates some surplus, in the form of profit, goodwill experience and so on, which can be retained in the organization and can be used for further input output process. Organizations are dependent upon their external environment for the inputs required by them and for disposing of their outputs in a mutually beneficial manner.

According to Degree of Human Intervention

i. **Manual System** – Where data collection, manipulation, maintenance and final reporting are carried out absolutely by human efforts.
ii. **Automated systems** – Where computers or microprocessors are used to carry out all the tasks mentioned above. However it will be wrong to say that a business system is 100% automated; rather, to some extent, it depends on manual intervention, may be in a negligible way.

Computers made it possible to carry out processing which would have been either too difficult or too much time-consuming or even impossible to do manually. A system may be even 100% manual. In earlier days, all accounting procedures and transactions, production details or sales data used to be maintained in different ledgers that were created by human efforts only and all these were *manual systems*. With introduction of computers and complexity in business procedures, these manual jobs were passed to computers in a major way and now a business system inherently involves a close man-machine interaction. The reasons for using computer in business area are as follows:

- Handling huge volume of data that is not manageable by human efforts.
- Storing enormous volume of data for indefinite period without any decay.
- Quick and accurate processing of data to match the competitive environment.
- Quick retrieval of information on query.
- Quick and efficient transportation of data/information to distant places almost at no cost.
- Availability of software tools for quick decision making in a complex situation.

According to Working/Output

i. **Deterministic System:** A deterministic system operates in a predictable manner. The interaction among the parts is known with certainty. If one has a description of the state of the system at a given point in time plus a description of its operation, the next state of the system may be given exactly, without error. An example is a correct computer program, which performs exactly according to a set of instructions.

ii. **Probabilistic System**: The probabilistic system can be described in terms of probable behavior, but a certain degree of error is always attached to the prediction of what the system will do. An inventory system is an example of a probabilistic system. The average demand, average time for replenishment, etc, may be defined, but the exact value at any given time is not known. Another example is a set of instructions given to a human who, for a variety of reasons, may not follow the instructions exactly as given.

Business Application System and its Development

Business application system, also called application software, is designed to support a specific organizational function or process, such as inventory management, payroll, or market analysis. The goal of application system is to turn data into information. For example, software developed for the inventory department at a bookstore may keep track of the number of books in stock for the latest bestseller. Application system for the payroll department may keep track of the changing pay scales of the employees.

System development involves developing or acquiring and maintaining application systems which are used for various day-to-day business activities. These business activities are called as Business Processes and they process data. The effective management and control of this development result into Business Systems which control Information Assets. Organizations use a standard set of steps, called a Systems Development Methodology, to develop and support their business applications.

Project Initiation

Whenever a business entity decides (i.e. stakeholders in the business or senior management) to undertake computerization, a Project will have to be initiated. This process is called as Project Initiation. Some examples of a formal Project Initiation are given as follows:

- A new business application is required to be developed to address a new or existing business process e.g. a billing system.
- Adoption of a new technology invented or available becomes advantageous to the business e.g. Internet based advertising for an advertising company.
- The application software to be developed, is expected to rectify the present problem related to existing business e.g. computerization of college admissions.
- The application software to be developed, is expected to rectify the present problem related to existing technology e.g. migrating from text-based computerized system to GUI based system as in case of old COBOL / XBASE based distributed banking to RDBMS based Core Banking system.

During project initiation, the project manager performs several activities that assess the size, scope, and complexity of the project, and establishes procedures to support subsequent activities. Depending on the project, some initiation activities may be unnecessary and some may be very useful. The major activities to be performed in the project initiation are as under:

- **Establishing the project initiation team:** This activity involves organizing an initial core of project team members to assist in accomplishing the project initiation activities.
- **Establishing a relationship with the customer:** A through understanding of the customer builds stronger partnerships and higher levels of trust.
- **Establishing the project initiation plan:** this step defines the activities required to organize the initiation team while it is working to define the scope of the project.
- **Establishing management procedures:** Successful projects require the development of effective management procedures.
- **Establishing the project management environment and project workbook:** The focus of this activity is to collect and organize the tools that we will use while managing the project and to construct the project workbook. For example, most diagrams, charts, and system descriptions provide much of the project workbook contents. Thus, the project workbook serves as a repository for all project correspondence, inputs, outputs, deliverables, procedures, and standards established by the project team.

The main outcome of Project Initiation is a formal Project Initiation Report which is presented to senior management or BOD. This will be accepted with or without modifications and then the next phases of SDLC will be rolled out. In case of SMEs or very small organizations, a formal Project Initiation Report may not be prepared. However, there may a 1 or 2 pages Concept Note or e-mail circular may be issued by the stakeholders of the business. Having a Project Initiation (formal or informal) helps to identify an objective of the organization for undertaking computerization and will form an initial document. If this report is circulated to all concerned personnel, then it will help in making people aware about the project.

Systems Development Methodology

System development methodology is a formalized, standardized, documented set of activities used to manage a system development project. It refers to the framework that is use to structure, plan and control the process of developing an information system. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses. Each of the available methodologies is

best suited to specific kinds of projects, based on various technical, organizational, project and team considerations. It should be used when information systems are developed, acquired, or maintained. The methodology is characterized by the following:

- The project is divided into a number of identifiable processes, and each process has a starting point and an ending point. Each process comprises several activities, one or more deliverables, and several management control points. The division of the project into these small, manageable steps facilitates both project planning and project control.
- Specific reports and other documentation, called deliverables, must be produced periodically during system development to make development personnel accountable for faithful execution of system development tasks. An organization monitors the development process by reviewing the deliverables that are prepared at the end of each key step. Many organizations rely on this documentation for training new employees; it also provides users with a reference while they are operating the system.
- Users, managers, and auditors are required to participate in the project. These people generally provide approvals, often called signoffs, at pre-established management control points. Signoffs signify approval of the development process and the system being developed.
- The system must be tested thoroughly prior to implementation to ensure that it meets users' needs.
- A training plan is developed for those who will operate and use the new system.
- Formal program change controls are established to preclude unauthorized changes to computer programs.
- A post-implementation review of all developed systems must be performed to assess the effectiveness and efficiency of the new system and of the development process.

An organization's system development methodology should be documented in the form of a system development standards manual. Among other items, such manuals often indicate:

- Methods for requesting systems development.
- Procedures to be followed, techniques to be used, and documentation to be prepared during system development.
- Reviews to be performed and signoffs to be obtained.

Need for Structured Systems Development Methodology

The following are the basic reasons for the need of Structured Systems Development Methodology:

- Software is a logical intangible product which makes all the difference in estimating, measuring and managing it without a methodology.
- Software products are not manufactured by machines (unlike most tangible products) and they are developed by developers. Therefore, their quality heavily depends on the quality of people carrying out system development. This mandates organized development efforts.
- Software products depend heavily on the quality of people carrying out system development, therefore, clear procedures for the development are required to be communicated and followed by the concerned employees. This helps in a better planning and control by the project managers on cost and delivery time and compliance to the standards and quality by the developers.
- Developing software products in an organized manner means organizations should have:
    - Planning and control i.e. software development should be treated as a Project,
    - Schedules of completion and deliverables in a time line for various phases,
    - Resources and cost estimation required for all the phases, and
    - Quality standards for comparing products of every phase.

This gives rise to apply Project Management techniques and tools for entire Software Development process which is discussed in Chapter 4 of the same module.

The Systems/Software Development Life Cycle (SDLC. is a common methodology for systems development in many organizations. This lifecycle approach i.e. SDLC involves defined phases and it is an incremental process of going towards next phase in building and operating business application systems. The phases may be undertaken in a serial manner (i.e. one after the other) or in a parallel manner. This constitutes a Model of SDLC.

Approaches to Systems Development

Since organizations vary significantly in the way they automate their business procedures, and since each new type of system usually differs from any other, several different system development approaches are often used within an organization. All these approaches are not mutually exclusive, which means that it is possible to perform some prototyping while applying the traditional approach.

Every software development methodology has more or less it's own approach to software development. There is a set of more general approaches, which are developed into several specific methodologies. These approaches are:

- Waterfall: Linear framework type,
- Prototyping: Iterative framework type,
- Incremental : Combination of linear and iterative framework type,
- Spiral : Combination linear and iterative framework type, and
- Rapid Application Development (RAD.: Iterative Framework Type.

These Systems Development Models are discussed in chapter 3 of the same module.

Risks associated with SDLC

The SDLC framework provides system designers and developers to follow a sequence of activities. It consists of a set of steps or phases in which each phase of the SDLC uses the results of the previous one. The SDLC is document driven. This means that at crucial stages during the process documentation is produced. A phase of the SDLC is not complete until the appropriate documentation or artifact is produced. These are sometimes referred to as deliverables. A deliverable may be a substantial written document, a software artifact, a system test plan or even a physical object such as a new piece of technology that has been ordered and delivered. This feature of the SDLC is critical to the successful management of an IS project.

The SDLC can also be viewed from a more process oriented perspective. This emphasizes the parallel nature of some of the activities and presents activities such as system maintenance as an alternative to a complete re-design of an existing system. The advantages of this system are:

- Better planning and control by project managers.
- Compliance to prescribed standards ensuring better quality.
- Documentation that SDLC stresses on is an important measure of communication and control.
- The phases are important milestones and help the project manager and the user for review and signoff.

Any methodology whether organized or unorganized has some element of Risk. It is necessary to know these risks prior to undertaking SDLC projects. The objective is to

- Identify risks
- Discovering methods to eliminate or mitigate them
- Accepting residual risk and going ahead with the project

Some of the SDLC risks are enlisted below:

- Generally, the development team finds it cumbersome, as structured methodology involves lot of documentation. E.g. a programmer may develop a program but may not document it later on or during program development. This is a natural human tendency.
- The users may find that the end product is not visible for a long time. This is because each SDLC phase process may go thru a cycle of doing work-document-test-quality accept-change-release work product.
- As the duration of project may be longer (due to formal structured methodology), it may not be suitable for small and medium sized organizations.

A well structured systems and methodology for the development of software has the following distinct processes:

- Detailed documentation identifying the need or problem for the development of software (Project Initiation, Feasibility Studies)
- Specifying the system to be developed (Requirements Analysis)
- The potential benefits resulting from the development of the new system. (Feasibility Study)
- Identification and evaluation of both external and internal factors affected by the problem and their effect on the business. (Project Initiation, Feasibility Studies)
- Designing of the system (System Design)
- Programming (Developing source code)
- Program testing
- Implementation of the resulting software

Whether an organization adopts a Structured Methodology or not, the above distinct processes or combination of them are generally observed in any system development effort. In industry, there may be lot of variations and different models which are used for above listed distinct processes.

Advantages of Structured Methodology for IS Auditors

If an organization simply buys a software product off-the-shelf without any methodology (e.g. adopting SDLC phases), an IS auditor cannot do any value-addition to the process. However, if a structured methodology is adopted, an IS auditor can participate in all the phases and value-add to each work product of the phases. Moreover, any audit demands an independent view of the system under scope. Auditor also requires evidence of what s/he has reviewed. Therefore, there are certain advantages to IS auditors, if a structured methodology is adopted. Some of these are listed below:

- IS auditor can have a clear understanding of SDLC phases due to documented processes.
- IS auditor can give report about compliance by the IS management about various SDLC phases. This gives senior management assurance about resulting system.
- Technically competent IS auditor can involve himself in various phases as an independent entity for expressing opinion and directing teams to proper outcome of the phase. E.g. an IS auditor can express his opinion and vet whether database design is adequately normalized or not.

Overview of Phases in Structured Methodology of SDLC

A brief introduction of various phases of SDLC is given below:

i. **Preliminary Investigation/ Systems Planning/Feasibility Study:** A preliminary investigation is undertaken when users come across a problem or opportunity and submit a formal request for a new system to the MIS department. This activity consists of three parts: request clarification, feasibility study and request approval. Generally the requests which are submitted to the MIS department are not clearly stated. Hence, before any system investigations can be considered, the system request must be examined to determine precisely what the originator wants. Thereafter, the analyst tries to determine whether the system requested is feasible or not in various terms like - technical, economical, time, operational etc. The third part of the investigation relates to approval of the request. Not all requested systems are desirable or feasible. Based on the observations of the analyst, the management decides which system should be taken up for development.

ii. **Requirements Analysis or Systems Analysis:** If, after studying the results of preliminary investigation, management decides to continue the development process, the needs of the users are studied. Analysts work closely with employees and managers of the organization for determining information requirements of the users. Several fact-finding techniques and tools such as questionnaires, interviews, observing decision-maker behavior and their office environment etc. are used for understanding the requirements. As details are gathered, the analysts study the present system to identify its problems and shortcomings and identify the features, which the new system should include to satisfy the new or changed user application environment. This step is also referred to as “systems analysis”.

iii. **System Design:** During system design, the user requirements that arose from analyzing the user applications environment are incorporated into a new systems design. The design of an information system produces the details that state how

a system will meet the requirements identified above. The analyst designs various reports/outputs, data entry procedures, inputs, files and database. He also selects file structures and data storage devices. These detailed design specifications are then passed on to the programming staff so that software development can begin.

iv. **Programming / Development/Construction:** After the system design details are resolved, such resources needs as specific type of hardware, software, and services are determined. Subsequently, choices are made regarding which products to buy or lease from which vendors. Software developers may install (or modify and then install, purchased software or they may write new, custom-designed programs. The choice depends on many factors such as time, cost and availability of programmers. The analyst works closely with the programmers if the software is to be developed in-house. During this phase, the analyst also works with users to develop worthwhile documentation for software, including various procedure manuals.

v. **Systems Testing:** Before the information system can be used, it must be tested. Systems testing is done experimentally to ensure that the software does not fail i.e. it will run according to its specifications and in the way users expect. Special test data are input for processing, and results examined. If it is found satisfactory, it is eventually tested with actual data from the current system.

vi. **Implementation:** After the system is found to be fit, it is implemented with the actual data. Hardware is installed and users are then trained on the new system and eventually work on it is carried out independently. The results of the development efforts are reviewed to ensure that the new system satisfies user requirements.

vii. **Post-Implementation Maintenance:** After implementation, the system is maintained; it is modified to adapt to changing users and business needs so that the system can be useful to the organization as long as possible.

The system development life cycle should be viewed as a continuous iterative process that recycles through each stage for many applications. Thus, even when a system is fully specified, designed, purchased and running, it is continually being enhanced or maintained. Enhancement and maintenance may require returning to one of the earlier stages of system development life cycle. *SDLC can be implemented by the Waterfall model which is a traditional model* and a sequential software development process, in which progress is seen as flowing steadily downwards (like a waterfall) through the different phases, as indicated above.

The following Table 1.2 enlists seven different phases of SDLC, along with the activities of each phase:

Phase No.	Phase Name	Deliverable/s	Activities undertaken
1.	Preliminary Investigation/ Systems Planning/ Feasibility Study	• Feasibility Study Report	• Determining strategic benefits of the system • Priority list of systems to be taken for computerization (unless it is ERP) • Cost-benefit analysis of the proposed system
2.	Requirements Analysis	• Requirements Analysis Report	• Gathering user (users at various levels) requirements for all processes in the proposed system.
3.	Systems Design	• Systems Design Report	• Designing user interface (screens and dialogue boxes), database and table designs, depiction of business processes and business rules (validations etc. thru various graphical representations such as System flow charts, data flow diagrams, screen and report layouts etc
4.	Programming / Development/ Construction	• Source programs and executable programs	• Writing programs using programming languages such as Visual Basic, Java etc. • If object oriented technology is used,

			sometimes, programming is called as Construction (as programmer is "constructing" system by collecting and joining various pre-developed components
5.	Testing	• Test results and suggestions for modifications if test results do not comply with test objectives	• Various kinds of testing are conducted such as Unit testing, interface testing etc.
6.	Implementation	• Final approval by QC / Management / Auditors	Following activities are undertaken in this phase • Data entry and/or migration of data from earlier computerized system • Setting up of system parameters for live working • Users training and hand-holding • Getting accreditation for system
7.	Post-Implementation Maintenance	• Evaluation and improvement changes to system thru users feed-back • Help desk support for users	• Solving day-to-day problems of users thru hand-holding and help-desk support • Improving operational system performance • Modifications and changes to the system as per feed-back from users

Table 1.2: SDLC Phases

Phase I: Preliminary Investigation/ Systems Planning/ Feasibility Study

The main objective of this phase is to determine and analyze the strategic benefits in implementing the system through evaluation and quantification of:

- Productivity gains,
- Future cost avoidance,
- Cost savings, and
- Intangible benefits like improvement in morale of employees.

A preliminary investigation is normally initiated by some sort of system request. The following activities are typically addressed in the Feasibility Study:

- To determine whether the solution is as per the business strategy,
- To determine whether the existing system can rectify the situation without a major modification,
- To define the time frame for which the solution is required,
- To determine the approximate cost to develop the system, and
- To determine whether the vendor product offers a solution to the problem.

Steps involved in feasibility study

The steps involved in the feasibility study phase are given as follows:

1. Identification of Problem
2. Identification of Objective
3. Delineation of Scope
4. Feasibility Study

Major document /deliverable of this phase is a Preliminary Investigation Report/ Feasibility Study for Management.

Identification of Problem

The first step in application development is to define the problem clearly and precisely. This can be done only after several rounds of discussions with the user group. Then its prevalence within the organization has to be assessed. A problem that has a considerable impact on the organization is likely to receive immediate management attention. User involvement will also be high, if they are convinced that the proposed solution will resolve the problem.

For instance, personnel in a functional area may feel that an existing system is outdated or a manager might want access to specific, new information that he claims will lead to better decisions. Shifting business requirements, changing organizational

environments, and evolving information technology may render systems ineffective or inefficient. Some reviews can be conducted to determine whether we should adopt new information technology like -

- A system still satisfies users' information needs. For example, do the system's reports provide sufficient information to solve the problems presently facing management?
- New design ideas can be incorporated. For example, should the organization develop an online transaction processing (OLTP) system?
- Evolving environmental changes, such as competition, require system changes. For example, should the organization provide direct computer links with its customers for online, direct customer ordering, perhaps using EDI?
- New types of business by the firm require system changes. For example, has a traditional wholesaler added a service or retail oriented branch requiring that consideration be given to an Internet site for customer sales and service?
- The user requests systems development when the system no longer efficiently and effectively meets the goals of the system. For example, a change in government regulations may require new or modified reports.

Other examples in the user-request category are:

- Excessive time spent on correcting errors.
- Current reports not meeting user decision-making needs.
- Escalating customer or vendor complaints.
- Erroneous system outputs causing problems
- Information system-caused delays slowing the operations system.

Whatever may be the reason, managers and users may feel compelled to submit a request for a new system to the IS department. If the need seems genuine, a system analyst is assigned to make a preliminary investigation. It is advisable for all proposals to be submitted to the steering committee for evaluation to identify those projects that are most beneficial to the organization. A preliminary investigation is then carried out by systems analyst, working under the direction of the steering committee.

Thus it can be concluded that the purpose of the preliminary investigation is to evaluate the project request. It is neither a designed study, nor it includes the collection of details to completely describe the business system. Rather it relates to collection of information that permits committee members to evaluate the merits of the project request and make an informed judgement about the feasibility of the proposed project.

The analyst working on the preliminary investigation should accomplish the following objectives:

- Clarify and understand the project request: What is presently being done? What is required and why? Is there an underlying reason different from the one the user has identified?
- Determine the size of the project: Does a request for a project call for new development or for modification of the existing system? The investigation to answer this question will also gather the details useful in estimating the amount of time and number of people required to develop the project.
- Determine the technical and operational feasibility of alternative approaches.
- Assess costs and benefits of alternative approaches: What is the estimated cost for developing a particular system? Will the proposed system reduce operating costs? Will the proposed system provide better services to customers, etc?
- Report findings to the management with recommendation outlining the acceptance or rejection of the proposal.

Identification of Objective

After the problem has been identified, it is easy to work out the objectives of the proposed solution For instance; inability to provide a convenient reservation system, for a large number of intending passengers was the problem of the Railways. So its objective was "to introduce a system wherein intending passengers could book a ticket from source to destination, faster than in real-time."

Delineation of Scope

The scope of a solution defines its boundaries. It should be clear and comprehensible to the user management stating what will be addressed by the solution and what will not. Often the scope becomes a contentious issue between development and user organizations. Hence, outlining the scope in the beginning is essential.

The following questions should be answered while stating the scope:

i. **Functionality requirements:** What functionalities will be delivered through the solution?
ii. **Data to be processed**: What data is required to achieve these functionalities?
iii. **Control requirements:** What are the control requirements for this application?
iv. **Performance requirements:** What level of response time, execution time and throughput is required?
v. **Constraints:** What are the conditions the input data has to conform to? For example, what is the maximum number of characters that a name can have in a database?

vi. **Interfaces:** Is there any special hardware/software that the application has to interface with? For example-Payroll application may have to capture from the attendance monitoring system that the company has already installed. Then the solution developer has to understand the format of data, frequency mode of data transfer and other aspects of the software.

vii. **Reliability requirements:** Reliability of an application is measured by its ability to remain uncorrupted in the face of inadvertent / deliberate misuse. The reliability required for an application depends on its criticality and the user profile. For instance-an AM application should protect the dataset against any misuse.

While eliciting information to delineate the scope, few aspects need to be kept in mind:

- Different users will represent the problem and required solution in different ways. The system developer should elicit the need from the initiator of the project alternately called champion or executive sponsor of the project, addressing his/her concerns should be the basis of the scope.
- While the initiator of the project may be a member of the senior management, the actual users may be from the operating levels in an organization. An understanding of their profile helps in designing appropriate user interface features.
- While presenting the proposed solution for a problem, the development organization has to clearly quantify the economic benefits to the user organization. The information required has to be gathered at this stage. For example- when you propose a system for Road tax collection, data on the extent of collection and defaults is required to quantify benefits that will result to the Transport Department.
- It is also necessary to understand the impact of the solution on the organization- its structure, roles and responsibilities. solutions, which have a wide impact, are likely to meet with greater resistance. ERP implementation in organizations is a classic example of change management requirement. Organizations that have not been able to handle this have had a very poor ERP implementation record, with disastrous consequences.
- While economic benefit is a critical consideration when deciding on a solution, there are several other factors that have to be given weightage too. These factors have to be considered from the perspective of the user management and resolved. For example- in a security system, how foolproof it is, may be a critical a factor like the economic benefits that entail.

The two primary methods with the help of which the scope of the project can be analyzed are:

- **Reviewing internal documents:** The analysts conducting the investigation first try to learn about the organization involved in, or affected by, the project. For example, to review an inventory system proposal, the analyst will try to know how the inventory department operates and who are the managers and supervisors. Analysts can usually learn these details by examining organization charts and studying written operating procedures.
- **Conducting Interviews:** Written documents tell the analyst how the systems should operate, but they may not include enough details to allow a decision to be made about the merits of a systems proposal, nor do they present users' views about current operations. To learn these details, analysts use interviews. Interviews allow analysts to know more about the nature of the project request and the reasons for submitting it. Usually, preliminary investigation interviews involve only management and supervisory personnel.

Feasibility Study

After possible solution options are identified, project feasibility - the likelihood that these systems will be useful for the organization – is determined. A feasibility study is carried out by the system analysts for this purpose. Feasibility study refers to a process of evaluating alternative systems through cost/benefit analysis so that the most feasible and desirable system can be selected for development. The feasibility study of a system is undertaken from three major angles: technical, economic and operational feasibility. The proposed system is evaluated from a technical view point first and if technically feasible, its impact on the organization and staff is assessed. If a compatible technical and social system can be devised, it is then tested for economic feasibility.

Dimensions of Feasibility

For computerization projects, a Feasibility Study will cover the following aspects of a project:

i. Operational – How will the solution work?
ii. Technical - Is the technology needed available?
iii. Economic – Return on Investment?
iv. Schedule /Time – Can the system be delivered on time?
v. Legal – Is the solution valid in legal terms?
vi. Financial – Is the solution viable financially?
vii. Behavioral –
viii. Resources –

Technical Feasibility: It is concerned with hardware and software. Essentially, the analyst ascertains whether the proposed system is feasible with existing or expected computer hardware and software technology. The technical issues usually raised during the feasibility stage of investigation include the following:

- Does the necessary technology exist to do what is suggested (and can it be acquired.?
- Does the proposed equipment have the technical capacity to hold the data required to use the new system?
- Can the proposed application be implemented with existing technology?
- Will the proposed system provide adequate responses to inquires, regardless of the number or location of users?
- Can the system be expanded if developed?
- Are there technical guarantees of accuracy, reliability, ease of access, and data security?

Some of the technical issues to be considered are given in the Table 1.3, which is as under:

Design Considerations	Design Alternatives
Communications channel configuration	Point to point, multidrop, or line sharing
Communications channels	Telephone lines, coaxial cable, fiber optics, microwave, or satellite
Communications network	Centralized, decentralized, distributed, or local area
	Independent vendor or in-house
Computer programs	Tape, floppy disk, hard disk, or hard copy
Data storage medium	Files or database
Data storage structure	Direct access or sequential files
File organization and access	Keying, OCR, MICR, POS, EDI, or voice recognition
Input medium	In-house or outsourcing
Operations	Instantaneous, hourly, daily, weekly, or monthly
Output frequency	CRT, hard copy, voice, or turnaround document
Output medium	Predetermined times or on demand
Output scheduling	Preprinted forms or system-generated

Printed output	forms
Processor	Micro, mini, or mainframe
Transaction processing	Batch or online
Update frequency	Instantaneous, hourly, daily, weekly, or monthly

Table 1.3: Technical Issues

Due to tremendous advancements in computer field these days, the technology is available for most business data processing systems but sometimes not within the constraints of the firm's resources or its implementation schedule. Therefore, trade offs are often necessary. A technically feasible system may not be economically feasible or may be so sophisticated that the firm's personnel cannot effectively operate it.

ii. **Financial Feasibility:** The solution proposed may be prohibitively costly for the user organization. For example – Monitoring the stock through VSAT network connecting multiple locations may be acceptable for an organization with high turnover. But this may not be a viable solution for smaller ones.

iii. **Economic Feasibility:** Also known as "Cost-Benefit Analysis", it includes an evaluation of all the incremental costs and benefits expected if the proposed system is implemented. After problems or opportunities are identified, the analysts must determine the scale of response needed to meet the user's requests for a new system, as well as the approximate amount of time and money that will be required in the effort. The analysts then determine just how much management wants to spend or change. Possible solutions are then examined in the light of the findings. Because of the myriad possibilities involved in most business situations every problem is different, and may require a solution different from that used in the past. Thus, common sense and intuition are key ingredients in the solution development process and is considered as the most difficult aspect of the study. The financial and economic questions raised by analysts during the preliminary investigation are for the purpose of estimating the following:

- The cost of conducting a full systems investigation.
- The cost of hardware and software for the class of applications being considered.
- The benefits in the form of reduced costs or fewer costly errors.
- The cost if nothing changes (i.e., the proposed system is not developed.

Estimating costs and benefits: After possible solution options are identified, the analyst should make a primary estimate of each solution's costs and benefits.

Cost : System costs can be sub divided into development, operational and intangible costs.

Development costs for a computer based information system include costs of the system development process such as

i. Salaries of the system analysts and computer programmers who design and program the system,
ii. Cost of converting and preparing data files and preparing systems manual and other supportive documents,
iii. Cost of preparing new or expanded computer facilities,
iv. Cost of testing and documenting the system, training employees, and other start up costs.

Operating costs of a computer based information system include:

Hardware/software rental or depreciation charges,

i. Salaries of computer operators and other data processing personnel who will operate the new system,
ii. Salaries of system analysts and computer programmers who perform the system maintenance function,
iii. Cost of input data preparation and control,
iv. Cost of data processing supplies,
v. Cost of maintaining proper physical facilities including power, light, heat, air conditioning, building rental or other facility charges and equipment and building maintenance charges, overhead charges of the business firm.

Intangible costs are costs that cannot be easily measure. For example, the development of a new system may disrupt the activities of an organization and cause a loss of employee productivity or morale. Customer sales and goodwill may be lost by errors made during the installation of a new system. Such costs are difficult to measure in rupees but are directly related to the introduction and operation of information system.

Benefits: The benefits which result from developing new or improved information systems that utilize EDP can be subdivided into tangible and intangible benefits. Tangible benefits are those that can be accurately measured and are directly related to the introduction of a new system, such as decrease in data processing cost. Intangible benefits such as improved business image are harder to measure and define. Benefits that can result from the development of a computerized system are summarized below:

1. Increase in sales or profits (improvement in product or service quality).

2. Decrease in data processing costs (elimination of unnecessary procedures and documents).
3. Decrease in operating costs (reduction in inventory carrying costs).
4. Decrease in required investment (decrease in inventory investment required..
5. Increased operational ability and efficiency (improvement in production ability and efficiency; for example, less spoilage, waste, and idle time).
6. New or improved information availability (more timely and accurate information, and new types and forms of information)
7. Improved abilities in computation and analysis (mathematical simulation).
8. Improved customer service (more timely service).
9. Improved employee morale (elimination of burdensome and boring job tasks).
10. Improved management decision making (better information and decision analysis)
11. Improved competitive position (faster and better response to actions of competitors)
12. Improved business and community image ("progressive" image as perceived by customers, investors, other businesses, government and the public..

Intangible though it is important to put a rupee and paise tag to each benefit for purposes of profit and loss statement, which can be done with diligence on the part of operating managers. The operating manager has homework to do here. The analyst can estimate for him the proportion of time the computer would save for the Chief Buyer, for example, by taking over his routine tasks and programmable decision making. It is now for the chief buyer to deliberate and come out with how best to utilize this time. He can, for example, undertake more extensive sourcing, spend more time on negotiations, participate more heavily in the value analysis effort and so on; therefore, it is he who has to quantify the benefits in monetary terms. This has led to dual advantage that subsequently during systems audit his performance can be appraised i.e., to what extent the payoffs he anticipated were actually realized upon computerization. However, it must be mentioned in passing that not all benefits are so hard to quantify. A computerized payroll system, for example, would likely displace some clerks and it is easy enough to measure benefits in that case.

v. **Schedule or Time Feasibility:** Schedule feasibility involves the design team's estimating how long it will take a new or revised system to become operational and communicating this information to the steering committee. For example, if a design team projects that it will take 16 months for a particular system design to become fully functional, the steering committee may reject the proposal in favor of a simpler alternative that the company can implement in a shorter time frame.
vi. **Resources Feasibility:** This focuses on human resources. Implementing sophisticated software solutions becomes difficult in non-metro locations. This is

because of the reluctance of skilled personnel to move to such locations.

vii. **Operational Feasibility:** It is concerned with ascertaining the views of workers, employees, customers and suppliers about the use of computer facility. The support or lack of support that the firm's employees are likely to give to the system is a critical aspect of feasibility. A system can be highly feasible in all respects except the operational and fails miserably because of human problems. Some of the questions which help in testing the operational feasibility of a project are stated below:

- Is there sufficient support for the system from management and from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for a change, there may be resistance.
- Are current business methods acceptable to users? If they are not, users may welcome a change that will bring about a more operational and useful system.
- Have the users been involved in planning and development of the project? Early involvement reduces the chances of resistance to the system and changes in general and increases the likelihood of successful projects.
- Will the proposed system cause harm? Will it produce poorer results in any respect or area? Will loss of control result in any areas? Will accessibility of information be lost? Will individual performance be poorer after implementation than before? Will performance be affected in an undesirable way? Will the system slow performance in any areas? Will it work when installed?

This analysis may involve a subjective assessment of the political and managerial environment in which the system will be implemented. In general, the greater the requirements for change in the user environment in which the system will be installed the greater the risk of implementation failure.

v. **Behavioral Feasibility:** Systems, which will be designed to process data and produce the desired outputs. However, if the data input for the system is not readily available or collectable, then the system may not be successful. This factor too must be considered.

vi. **Legal Feasibility:** Legal feasibility is largely concerned with whether there will be any conflict between a newly proposed system and the organization's legal obligations. Any system, which violates the local legal requirements, should also be rejected. For example, a revised system should comply with all applicable federal and state statutes about financial reporting requirements, as well as the company's contractual obligations.

This study is conducted by the team (either in-house or outsourced. through meetings with users at various levels. The team will collect all the data necessary for the above listed aspects (documents, forms, register photocopies, reports etc.. The team may take demonstrations of ready-made software solutions to quickly understand various aspects of the system. This will enable the team to discuss Making of the software or buying readymade solution in this report. The team generally consists of at a minimum a technical expert and a functional expert. Depending upon complexity of the organization additional team members such as documentation expert may be included in the team.

As mentioned above, Feasibility Study Report is the deliverable / outcome of this phase. This report should be presented to the senior management (or BOD. and the management should accept it with or without modifications, after which the project will enter the next phase viz. Requirements Analysis. Sometimes, it may be possible that Feasibility Study is merged into Requirements Analysis and a combined Requirements Analysis Report may be prepared.

Phase II – Requirements Analysis

This is the phase where a detailed elicitation of user's requirements to the maximum possible level (atomic level) is done. The Feasibility Study will give overall direction to the team conducting this study. However, this direction will be fine tuned and finalized to maximum possible detail. For example, in Feasibility Study a Payroll System is decided as scope for computerization, then, in this phase each sub-system (or module) of the payroll process with it's breakdown processes will be studied and described.

Techniques for requirements gathering

This will be done by undertaking the following indicative activities:

1. Understanding Requirements

The following are the major activities under this section:

- Identify and consult the stakeholders to determine their expectations and resolve their conflicts, if any. Many-a-times stakeholders give "one-liner" requirements for the system. For example, for a payroll system, stakeholders may also specify requirements of employee appraisals. Generally, employee appraisal requirements are classified under Personnel Management system and if scope is not to cover Personnel system then this is a conflicting area.
- Analyze requirements to detect and correct conflicts and determine priorities.

- Verify the requirements are complete, consistent, unambiguous, verifiable, modifiable, testable and traceable.
- Resolve conflicts between the requirements set and the resources that are available
- Identify how the system should interact with its environment. For example, in a financial accounting system, is the system going to scan the supporting bills / documents and attach it to the financial transaction?

2. Study of History, Structure and Culture

The study of the history of systems in an organization gives an idea of the types of systems that have been extremely useful, issues that have not been addressed over a period and new issues that require attention.

The organizational structure gives an idea of the power equations within all organization. An understanding of the organizational culture is possible by talking to users. Sometimes they may be guarded, but experienced systems analysts can read between the lines. Solutions that are not consistent with the culture often fail. To cite an example, one of the benefits of ERP systems is empowerment of employees. However, since many Indian organizations have pyramidal structures, users do not construe this as a benefit.

3. Study of Information Flows

In order to design a new system, it is necessary to study the existing system. Recall that, a system consists of sub-systems, sub-sub-systems and so on. In today's industry, a system is also considered as a collection of various processes which are done one after the other or simultaneously. Study of existing system, involves study of the business processes. For example, placing a purchase order or opening of a bank account can be considered as processes which can be further broken down into sub-processes. These are studied and steps are written down. Sometimes, this is also called as Work Flow, indicating how work flows from one person to another person or from one department to another department.

The next section gives the methods of eliciting the user requirements, collected for above given activities. The entire above information gathering happens through series of following activities:

- Meetings with users at various levels
- Observation of users at work
- Forms, documents, registers being handled in the system
- Verbal interactions done by users to collect and disseminate the information

Once user requirements are collected at all levels, the team documents all these requirements in an organized manner and presents it to the user management for their modification, approval and endorsement. The commitments are obtained from the user departments and system developers to contribute the necessary resources to complete the project.

Eliciting user requirements using Structured Analysis

In present days, industry follows either Structured Analysis or Object Oriented Design (through UML design – please refer next chapter) or mostly a hybrid of both the methodologies. Traditional Structured Analysis methodology is presented below in a very concise manner. Under this (this is also termed as one of SDLC models), the following elements of eliciting requirements are prepared:

1. Context Flow Diagrams / Data Flow Diagrams
2. Decision Tables / Decision Trees / Structured English (also called as psuedocode)
3. Entity-Relationship Diagrams
4. Data Dictionaries
5. State-Transition Diagrams
6. System and Program Flowcharts
7. Interface in form of data entry screens and dialogue boxes
8. Report Layouts

The above technique can be used for both eliciting user requirements as well as giving design of the system. This is somewhat similar to a building architect who prepares blue print for showing and getting approval of users (the builder, govt. authorities etc. as well as uses it as a design to explain to the people who are going to build the structure of building.

1. Context and Data Flow Diagrams (DFD.

Developers prepare a textual description of various business processes along with Context and Data Flow Diagrams (DFD.. A data flow diagram uses a few simple symbols to illustrate the flow of data among external entities (such as people or organizations, etc.), processing activities and data storage elements. A data flow diagram (DFD. graphically describes the flow of data within an organization. It is used to document existing systems and to plan and design new ones. There is no ideal way to develop a DFD; different problems call for different methods. Context diagram is nothing but a summary diagram depicting context in which system will operate.

A sample DFD is given below. The DFD gives flow of data through the system. It is very easy to comprehend as only four symbols are used for depict the flow of data,

e.g. process, source of data or sink of data, arrow indicating direction of flow, data store. Each is represented on a DFD by one of the symbols shown in Table 1.4.

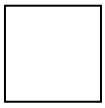
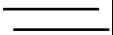
Symbol	Name	Explanation
	Data Sources and Destinations	The people and organizations that send data to and receive data from the system are represented by square boxes. Data destinations are also referred to as data sinks.
	Data Flows	The flow of data into or out of a process is represented by curved or straight lines with arrows.
	Transformation Process	The processes that transform data from inputs to outputs are represented by circles. They are often referred to as bubbles.
	Data stores	The storage of data is represented by two horizontal lines.

Table 1.4: Data Flow Diagram Symbols

Consider the following example:

Process description for Applicant selection process:

1. Department submits vacant posts in the department for approval of personnel department
2. Personnel department approves or rejects the vacant posts
3. Applicant applies for a job post
4. Personnel department verifies the credentials for candidature
5. A call letter for interview is sent to applicant
6. An intimation memo is sent to department for interview

The DFD for above activities is presented below in Fig. 1.3:

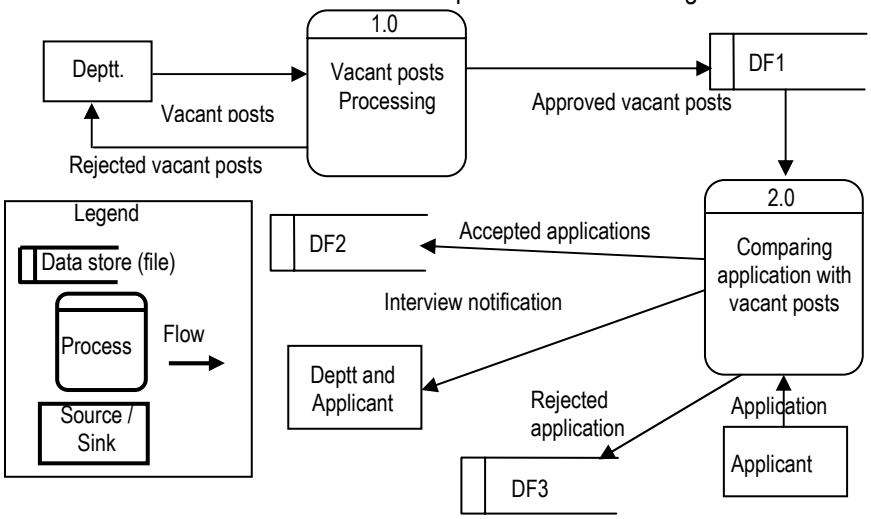


Fig. 1.3: DFD

From the above description and diagram, a developer can gain correct understanding of the data flow as well as user is ensured that developer has correctly understood his requirements. Like this, for all minute level business processes, a running description and graphical representation is given, through which developer understands the flow of data in various business processes.

However, one can easily see that, even though DFD tells us flow of data thru the system processes, it does show us structure and relationship amongst the data items. For example, in above DFD what is the data structure of application? What is that of vacant post? How they are related? All this missing information is given by next element of structured analysis viz. Entity-Relationship diagram.

2. Entity-Relationship (ER) Diagrams

An entity can be any of the following types:

- Person e.g. Employee, Student, Patient etc.
- Place e.g. State, Region, Branch
- Object e.g. Machine, Building, Automobile
- Event e.g. Sale, Registration, Renewal
- Concept e.g. Account, Course, Work Centre, Desk

An E-R diagram gives us structure and relationship between entities, as shown in Fig. 1.4. For example:

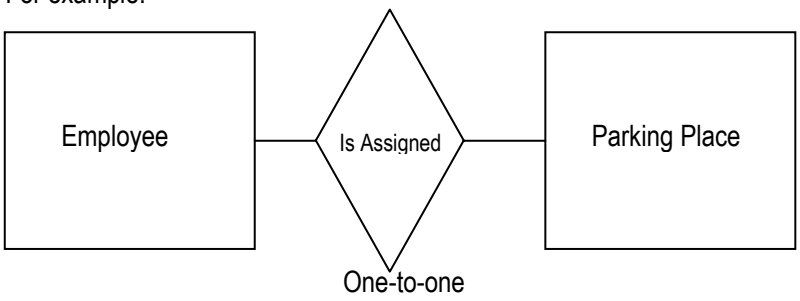


Fig. 1.4: ER Diagram

In above diagram, employee (person) is represented as one entity and parking place (place) as another. They are related to each other in one-is-to-one relationship (diamond indicates relationship). Since there are two entities this is called as binary one-to-one relationship. Similarly, we can have unary, ternary relationships within entities. For the entities described above, their structure is also given. E.g. in above case employee will have structure (also called as attributes or fields or columns) as Employee Identification, Employee name, Designation and so on. Parking place can have attributes such as Parking Place ID, description, area, status indicator, type of vehicle to be parked etc. Any further discussion is beyond the scope of this material.

3. Data Dictionaries

In an organization, there are many hundreds or thousands of data items which are used across various systems. It becomes humanly impossible to keep track of different data items, their structure and relationships. Data dictionaries are nothing but repository of data items handled in an organization or in a system. Through these data dictionaries, developers can find out what data is being used in which system and whether it has been already defined in some other system. If so, developer can simply use that data structure instead of defining it afresh.

So, now, we have flow of data and structure and relationship of data. But how we process the data? What is the logic of processing data? E.g. in the above example, how parking place is allotted for a new employee? Or what is to be done if all parking is full and still there is persons awaiting parking. For this, missing logic description, we will resort to either of the following: Decision table or decision tree or structured English (psuedocode)

They are given below:

Case: In payroll system, there are 2 kinds of employees, Salaried and Hourly Paid. The business rule is that Salaried employees must be paid Base Salary even they are absent. In case of Hourly Paid employees, pay them base salary for 40 hours of work, overtime for hours worked more than 40 hours and produce absence report for hours worked for less than 40 hours. The following Decision Table depicts this business rule

4. Decision Table / Decision Tree / Structured English

A decision table is a table which may accompany a flowchart, defining the possible contingencies that may be considered within the program and the appropriate course of action for each contingency. Decision tables are necessitated by the fact that branches of the flowchart multiply at each diamond (comparison symbol) and may easily run into scores and even hundreds. If, therefore, the programmer attempts to draw a flowchart directly, he is liable to miss some of the branches.

When a set of complex conditions needs to be evaluated for selecting appropriate action, then the decision table is the preferred option. The decision table provides conditions and actions in a tabular form. The decision table is divided into four quadrants. The upper left quadrant contains all conditions. The lower left quadrant contains actions that are possible for each combination of conditions. Values for all the possible combinations of a given set of conditions and the corresponding event (or action to be taken) are stored in the right quadrants. Each column in the right quadrant can be interpreted as a processing rule. A decision table is divided into four parts:

Condition Stub - (which comprehensively lists the comparisons or conditions);

Action Stub- which comprehensively lists the actions to be taken along the various program branches;

Condition entries - which list in its various columns the possible permutations of answer to the questions in the conditions stub.; and

Action entries - (which lists, in its columns corresponding to the condition entries the actions contingent upon the set of answers to questions of that column).

Conditions / Courses of Action	Rules			
	1	2	3	4
Employee Type	Salaried	Hourly	Hourly	Hourly
Hours Worked	-	< 40	40	> 40
Pay base salary	✔			
Calculate hourly wage		✔	✔	✔
Calculate overtime				✔
Produce absence report		✔		

Table 1.5: Decision Table

The same business rule can be depicted in the form of Decision Tree instead of Decision Table as given below in Fig. 1.5:

Decision Tree

A decision tree (or tree diagram) is a support tool that uses a tree-like graph or model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. Major benefits of Decision Trees are given as follows:

i. Decision trees are commonly used in operations research, specifically in decision analysis, to help identify a strategy most likely to reach a goal.
ii. Decision trees are descriptive means for calculating conditional probabilities.
iii. It is a common method used in data mining wherein each interior node corresponds to a variable; an arc to a child represents a possible value of that variable. A leaf represents a possible value of target variable given the values of the variables represented by the path from the root.

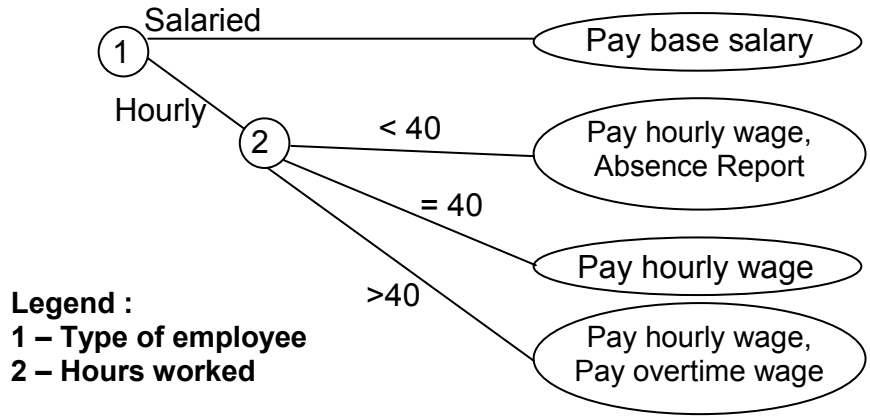


Fig. 1.6: Decision Tree

1 & 2 indicate nodes of the decision tree and lines indicate branches. Oval shapes represented leaves of tree indicating actions to be taken.

The same business rule can also be represented as a Structured English or also called as Psuedocode.

Structured English (Psuedocode)

Structured English is the use of the English language with the syntax of structured programming. Thus structured English aims at getting the benefits of both the programming logic and natural language. Program logic helps to attain precision while natural language helps in getting the convenience of spoken languages.

Structured English consists of the following elements:

i. Operation statements written as English phrases executed from the top down,
ii. Conditional blocks indicated by keywords such as IF, THEN, and ELSE, and
iii. Repetition blocks indicated by keywords such as DO, WHILE, and UNTIL

The following guidelines are generally used when writing Structured English:

i. Statements should be clear and unambiguous
ii. Use one line per logical element
iii. All logic should be expressed in operational, conditional, and repetition blocks
iv. Logical blocks should be indented to show relationship
v. Keywords should be capitalized.

Examples of keywords that may be used

START, BEGIN, END, STOP, DO, WHILE, DO WHILE, FOR, UNTIL, DO UNTIL, REPEAT, END WHILE, END UNTIL, END REPEAT, IF, IF THEN, ELSE, IF ELSE, END IF, THEN, ELSE THEN, ELSE IF, SO, CASE, EQUAL, LT, LE, GT, GE, NOT, TRUE, FALSE, AND, OR, XOR, GET, WRITE, PUT, UPDATE, CLOSE, OPEN, CREATE, DELETE, EXIT, FILE, READ, EOF, EOT

Example of Structured English

A bank will grant loan under the following conditions:

1. If a customer has an account with the bank and had no loan outstanding, loan will be granted.
2. If a customer has an account with the bank but some amount is outstanding from previous loans then loan will be granted if special approval is needed.
3. Reject all loan applications in all other cases.

In Structured English, the above conditions can be written as:

IF Customer has a Bank Account **THEN**

IF Customer has no dues from previous account **THEN**

Allow loan facility

ELSE

IF Management Approval is obtained **THEN**

Allow loan facility

ELSE

Reject

ENDIF

ENDIF

ELSE

Reject

ENDIF

Here we notice that, the Structured English is really very close to a programming language.

Developer will decide, based on complexity of the situation whether to adopt Decision Table, Decision Tree or Structured English.

5. State Transition Diagram

These diagrams help in enabling developer to know through how many states a particular entity goes during data processing. This is not given by any of the above listed diagrams. For example, in case of an ATM machine, it can go through the following state-transitions and depending on the current state of the entity, some action is initiated.

- Idling state (i.e. ATM machine is idling as no user is using it). So program residing in ATM machine's memory is awaiting for user to come and insert an ATM card.
- Card acceptance state – As soon as user inserts a card, a program module is activated and the machine sucks the card
- Card validation state – Now ATM machine transits to card validation wherein, the program will read the card number and validate it against database for existence
- Denying state – if card is found invalid, the machine will go in denying state.

And so on…

A correct State-transition diagram will link program modules to the states and actions thereupon of an entity.

6. System Charts / Program Flow Charts

System charts present various system modules in a block diagram. Whereas DFD gives data flow through a system, a System Chart gives flow of sub-systems or modules.

Program flow charts depict logic of a program in a very great detail. Now-a-days, developer prefer Structured English to program flow charts as drawing these diagrams require diagramming tools and is a cumbersome process.

It must be noted that, under Structured Analysis, various researchers have listed many other elements of design apart from listed above.

7. Interface in form of data entry screens and dialogue boxes

User has to enter his business data as well as other data in the system through some interface. This interface is provided by designing screens and dialogue boxes. Generally, user's business data is entered through data entry screens whereas other supporting data (such as user ID, passwords, user confirmations etc. is entered via dialogue boxes. Under this element, a developer designs screen and dialogues

boxes layouts with the help of computer software. These can be shown to users for getting their approvals, modifications and endorsement. Generally, developers design standards for these such as background and foreground color schemes for screens, position on the screens, colors for data items, graphics etc.

8. Report layouts

Report layouts give output of processed data or information. This is the element of system which will have a direct impact on users and success of developer largely depends upon the way in which he presents the reports (either on-screen or printed.. Reports can be final reports for user consumption or intermediate reports. They can be for internal use or to be circulated to outsiders. The following are some of the important considerations, which should be taken into account for layout and contents of reports.

- Page numbering style and position e.g. Page : 2 of 50
- Report title – will appear once at the beginning of report
- Processing date – will indicate when the report was processed
- Report period date – will indicate report for the period
- Control totals – on each page as page totals and on next page as B/f totals
- Input/output record counts – for control purpose
- Report grand totals – for control purpose and giving total amounts
- Report criteria – to indicate if report is on filtered out data

All the above mentioned seven elements of Requirements Analysis (Systems Analysis) are also used for specifying designing of the system. In fact, in earlier days, all these were considered as part of a Systems Analysis and Design Report. Due to increasing complexity and user awareness, these Structured Analysis elements are also used for eliciting systems requirements.

Requirements Analysis Report is the deliverable of this phase which is presented to the user management. User management has to go through this report critically and suggest modifications. Once accepted, this document becomes the Blueprint for the further development efforts.

Requirement analysis can be conducted by in-house team or can be outsourced. Here again, the team generally consists of at a minimum a technical expert and a functional expert. Depending upon complexity of the organization additional team members such as documentation expert may be included in the team. Functional expert helps the technical team members in understanding the functionality of the system and “tricks of the trade”. The technical team will convert functional understanding into technical specifications. For example, in a system if there is a tax calculation to be done, then functional consultant will understand the tax calculation

methods, processes and other details from the user's functional person. Being a functional expert he will quickly understand tax calculation process and will be able to ask right questions. The technical team member then understands the same process from functional expert and converts each process into his understanding for eliciting in textual description and graphical representation as mentioned above.

In the industry, the Requirement Analysis is known by different names such as Systems Requirements Specifications (SRS), Business Requirements Specifications (BRS), Users Requirements Specifications (URS) or Users Requirement Document (URD.. Strictly speaking all these will give different aspects of requirements, however, industry calls classifies all as SRS or BRS or URS or URD for simplicity.

If this is accepted, the project goes into the next phase which is Design Phase. However, this phase and the next phase viz. Programming / Construction will not be present if it has been decided to acquire a readymade solution (as a Feasibility Study outcome), since design and programming would have been taken care of by the readymade solution vendor. However Requirements Analysis is must because later on the software vendor will be configuring the readymade solution as per the specific requirements of the organization.

Software Acquisition

Software Acquisition is a very important activity, even though not considered as a Phase in SDLC. The following steps shall be relevant in acquisition of the software:

1. The feasibility study in relation to the software acquisition should contain documentation that supports the decision to acquire the software.
2. The decision is based upon various factors like Cost difference between development and acquisition, availability of the required software readily in the market, the time saved between development and acquisition etc.
3. A project team comprising of technical supports staff and key users should be crated to prepare Request for Proposal (RFP).
4. The project team should carefully compare and examine the various responses of the Vendor to the RFP, prepare the gap analysis and shortlist the vendors on the basis of such evaluation.
5. The project team should invite the short listed vendors to have presentation of the product and the processes. The user's participation with feedback must be ensured in such presentation. The project team may also visit some of the customers to have live demonstration of the software. The discussion with the customers should focus on:

    i. Reliability of the Vendor's deliverables
    ii. Vendor's commitment to service to sort out the system problems

iii. Vendor's commitment to user training and system documentation

6. Once the final vendor is confirmed, the vendor should be invited to present a pilot session so as to enable the project team to understand the system, have a hands-on experience and suggest for the areas of customization.
7. On the basis of the above activities, the vendor's presentations and final evaluation, the project team can make the final selection of the product.
8. The last step in the acquisition process is the negotiation and signing of a legal contract with the vendor. Contract should contain the following items:

   i. Specific description of the system deliverables and their respective costs
   ii. Commitment dates for their delivery in the order of priority decided by the purchaser
   iii. Commitments for delivery of documentation, fixes, upgrades, new release notification and training to the purchaser
   iv. Allowance of escrow agreement if the deliverables do not include source code.
   v. Description of the support to be provided during installation on site.
   vi. Provision for a reasonable period of acceptance, before purchase is made
   vii. Allowance for changes to be made by the customer company
   viii. Maintenance agreement with the specified period and the maintenance cost
   ix. Payment schedule for the software should be linked to actual delivery dates

9. Managing and control on the implementation of the system is required with the regular status reports.
10. IS auditors job in the software acquisition process would be to determine whether adequate level of security controls has been considered prior to making an agreement. This is required to ensure data integrity of the information processed and controls like audit trails, password controls and overall security of the application. The above procedures should be more elaborate and systematic in case where the business is implementing ERP systems giving fully integrated corporate solutions like SAP, Oracle Financials, BAAN, PeopleSoft, JD Edwards etc.

Following Table 1.6 lists out certain parameters for preparing RFP:

Item	Description
Product v/s System requirements	1. Comparison between the required system and the vendor developed product to include the volume of transactions that the software can handle and the database size 2. Acceptance or otherwise from the Users for the

	deficiencies in the vendor product. 3. Possibility of the customization by the Vendor to address the deficiencies
Customer References	Validate the Vendor's claims about their product performance and timely completion of work by the vendor from the vendor's customers.
Vendor viability and financial stability	Evaluate the vendor's viability with reference to period for which the vendor is in operation, the period for which the desired product is being used by the existing customers and the Vendor's financial stability on the basis of the market survey and the certification from the customers and on certain supporting documentation from the Vendor
Availability of complete and reliable documentation about the new software	Review the complete set of system documentation provided by the Vendor prior to acquisition of the software. The documentation should in detail and precise.
Vendor support	Evaluate what kind support the vendor provides for the software like onsite maintenance, online updation of upgrades, onsite training to users, automatic new version notifications, 24 x 7 helpline etc.
Response time	Time taken for the vendor to respond and fix in case a problem is reported.
Source code availability	1. If the software is developed only for the concerned business, the vendor has no right to further sale. The source code must be given by Vendor. 2. In other case, the source code should be deposited with a third party so that the same would be available if the vendor goes out business. The source with the program updates and programs fixes should be included as a part of escrow agreement.
Vendor's experience	Vendor having longer experience in the desired software is more acceptable
A list of recent or planned	This will ensure the vendors effort to keep the product

enhancements to the product with dates	current.
List of current customers	More the number of customers, more is the accept-ability of the product.
Acceptance testing of product	The vendor should allow the acceptance testing by the users to ensure that the product satisfies the system requirements of the business. This should be done before purchase commitment.

Table 1.6: Parameters for preparing RFP

Roles involved in SDLC

There are certain standard functions (not designations) during the development process. This means that, these roles may be combined, especially for small organizations and may be performed by the same individual. Under such cases, IS Auditor has to remember to evaluate conflicts between the roles (segregation of duties) as detailed out in Module-II. The roles are enlisted here in brief:

- **Steering Committee**

Wherever large-scale application development is undertaken, a steering committee is set up to supervise and direct the projects. This committee provides funding and overall direction to the projects.

- **Project Manager**

A project manager is normally responsible for more than one project and liaisons with the client or the affected functions. This is a middle management function, and he is responsible for delivery of the project within the time and budget.

- **Systems Analyst**

The systems analyst is also referred to as a business analyst. His main responsibility is to conduct interviews with users and understand their requirements. He is a link between the users and the programmers. He converts the users requirements in the system requirements. He plays a pivotal role in the Requirements analysis and Design phase.

- **Module Leader/Team Leader**

A project is divided into several manageable modules, and the development responsibility for each module is assigned to module leaders.

- **Programmers**

Programmers are the masons of the software industry. They convert design into

programs by coding using programming language. They are also referred to as coders or developers.

- **Database Administrator (DBA.**

The data in a database environment has to be maintained by a specialist in database administration so as to support the application program. The database administrator handles multiple projects; he ensures the integrity and security of information stored in the database.

- **Quality Assurance Team**

This team checks compliance with the standards set by the organization, by project teams on a periodic basis.

- **Testers**

Testers are junior level quality assurance personnel attached to a project. They test programs and subprograms as per the plan given by the module / project leaders and prepare test reports.

- **Domain Specialist**

Whenever a project team has to develop an application in a field that's new to them, they take the help of a domain specialist.

- **Technology Specialist**

IT is developing so rapidly that even IT professionals find it difficult to keep track of all developments, let alone develop expertise. This has resulted in experts in specific technology areas, such as Microsoft technology, Web-enablement and the like. If the application development team needs guidance in any of these areas, then it could associate with experts on a part- time basis.

- **Documentation Specialist**

These professionals are responsible for the creation of user manuals and other documentation.

- **IS Auditor**

Should the IS auditor be part of the development process? As a member of the team, he can ensure that the application development also focuses on the control perspective. He should be involved at the Design Phase and the final Testing Phase to ensure the existence and the operations of the Controls in the new software.

SDLC Controls

SDLC process begins with, how new systems or modifications to the current systems are requested, prioritized, actually designed, development of the source code to perform the functions desired, testing of the system developed and/or modified,

training of employees to use the new/modified system, and the final system implementation.

IS Auditor should consider the following influencing SDLC:

vi. In-house design and development of the system (using internal resources)
vii. Design and development of the system by using fully or partly the outsourced resources located onsite or offsite
viii. Off the shelf available packages implemented as-is without any customization
ix. Off the shelf available packages implemented as-is with some customization

At times, large complex applications may involve a combination of the above.

IS Auditor should be aware of implications of the following risks, while auditing SDLC:
x. Adoption of inappropriate SDLC for the application system
xi. Inadequate controls in the SDLC process
xii. User requirements and objectives not being met by the application system
xiii. Lack of involvement of all the stakeholders
xiv. Lack of management support
xv. Inadequate project management
xvi. Inappropriate technology and architecture
xvii. Change in scope
xviii. Time over-runs
xix. Budget over-runs
xx. Insufficient attention to security and controls
xxi. Performance criteria not being met
xxii. Inappropriate resourcing / staffing model
xxiii. Incomplete documentation
xxiv. Inadequate contractual protection
xxv. Inadequate adherence to the development methodologies
xxvi. Insufficient attention to interdependencies on other applications and processes
xxvii. Inadequate configuration management
xxviii. Insufficient planning for data conversion/migration and cutover

SDLC Audit Scope

IS auditor should consider the following scenarios while finalizing the SDLC Audit Scope and review relevant SDLC stages:

- **Pre-Implementation Review**: IS auditor should study the proposed SDLC model and the related aspects to assess the appropriateness and the potential risks and provide the necessary risk mitigation recommendations to the management.
- **Parallel / Concurrent Review:** IS auditor should review the relevant SDLC

stages, as they are happening, to highlight risks/issues and provide necessary risk mitigation recommendations to the appropriate management.

- **Post-Implementation Review**: IS auditor should review the relevant SDLC stages after their completion to highlight issues faced and provide recommendations for downstream corrections (if possible) and to serve as a learning tool for the future.

Auditing SDLC

IS auditor should consider following aspects while auditing and evaluating SDLC phases:

xxix. Project charter
xxx. Roles and responsibilities of different groups / committees (e.g. Project steering committee)
xxxi. Adopted project management methodology
xxxii. Application Development methodology / model
xxxiii. Contractual terms with the vendors for purchased applications (E.g. Service Level Agreements –SLAs)
xxxiv. Contractual terms with the vendors for outsourced services (E.g. Service Level Agreements –SLAs)
xxxv. Approvals and sign-offs by the Project steering committee for each SDLC stages
xxxvi. Deliverables of each SDLC stages
xxxvii. Minutes of relevant meetings
xxxviii. Project tracking and reporting documentation
xxxix. Resource management
xl. Ongoing risk management
xli. Quality Control / Assurance
xlii. Change management
xliii. Data conversion/migration
xliv. Application testing documentation
xlv. Relevant legal, regulatory and policy aspects to be complied with, if any

Master Checklist

Checklist for Auditing Entity- Level Controls

S. No.	Checkpoints	Status
1.	Whether a review is done of the overall IT organization structure to ensure that it provides for clear assignment of authority and responsibility over IT operations and that it provides for adequate segregation of duties?	

2.	Whether a review is done for the IT strategic planning process to ensure that it aligns with business strategies. Evaluate the IT organization's processes for monitoring progress against the strategic plan?	
3.	Is it determined whether technology and application strategies and roadmaps exit, and evaluate processes for long-range technical planning?	
4.	Whether a review is done for the performance indicators and measurements for IT. Ensure that processed and metrics are in place (and approved by key stakeholders) for measuring performance of day-to-day activities and for tracking performance against SLAs, budgets, and other operational requirements?	
5.	Whether a review is done for the IT organization's process for approving and prioritizing new projects? Determine whether this process is adequate for ensuring that system acquisition and development projects cannot commence without approval. Ensure that management and key stakeholders review project status, schedule, and budget periodically throughout the life of significant projects.	
6.	Is there any evaluation of standards for governing the execution of IT projects and for ensuring the quality of products developed or acquired by the IT organization? Determine how these standards are communicated and enforced.	
7.	Is there any assurance that IT security policies exist and provide adequate requirements for the security of the environment? Determine how those policies are communicated and how compliance is monitored and enforced.	
8.	Is there any review and evaluation of risk-assessment process in place for the IT organization?	
9.	Is there any review and evaluation of the process for ensuring that IT employees at the company have the skills and knowledge necessary for performing their jobs?	
10.	Is there any review and evaluation of the policies and processes for assigning ownership of company data, classifying the data, protecting the data in accordance with their classification and defining the data's life cycle?	

11.	Whether it is ensured that effective process exist for complying with applicable laws and regulations that affect IT (e.g. HIPAA, Sarbanes- Oxley) and for maintaining awareness of changes in the regulatory environment.	
12.	Is there any review and evaluation of the process for ensuring that end users of the IT environment have the ability to report problems, have appropriate involvement in IT decisions, and are satisfied with the services provided by IT.?	
13.	Is there any review and evaluation of the processes for managing third-party services, ensuring that their roles and responsibilities are clearly defined and monitoring their performance?	
14.	Is there any review and evaluation of the processes for controlling non employee logical access?	
15.	Is there any review and evaluation of the processes for ensuring that the company is in compliance with applicable software licenses?	
16.	Is there any review and evaluation of the controls over remote access into the company's network?	
17.	Is there any assurance that hiring and termination procedures are clear and comprehensive?	
18.	Is there any review and evaluation of the policies and procedures for controlling the procurement and movement of hardware?	
19.	Is there any assurance that system configurations are controlled with change management to avoid unnecessary system outages?	
20.	Is there any assurance that media transportation, storage, reuse, and disposal are addressed adequately by company-wide policies and procedures?	
21.	Is there any verification that capacity monitoring and planning are addressed adequately by company policies and procedures?	
22.	Based on the structure of your company's IT organization and processes, is there any identification and audit for other entity-level IT processes?	

Summary

In this chapter we have learned the following major aspects:

- Meaning of a System,
- The characteristics of a system and how each system has a life cycle after which it needs to be replaced,
- A project should be initiated through a Project Initiation report whenever some major business change in respect of IT is undertaken,
- The need for adopting Structured Methodology for IT Systems development projects due to people oriented nature of these projects. Structured Methodology involves planning and organizational control over the entire project life cycle.
- Risks associated with adopting structured methodology and importance from IS audit perspective of adopting it,
- Distinct activities undertaken in the structured methodology which is nothing but SDLC,
- All the SDLC phases and activities undertaken in each phase,
- Description of 2 phases viz. Feasibility Study and Requirements Analysis. Feasibility Study describes various feasibility aspects of the project such as economic, technical, legal, time etc.,
- Requirements analysis is carried out to know the requirements of various users at all levels. Business systems can be looked upon as collection of processes and sub-processes and how in requirements analysis developers collect data about these processes,
- We have also seen the methods of putting down the collected requirements in descriptive as well as graphical manner for ease of understanding through several diagrams such as context diagrams, data flow diagrams and so on,
- We digressed from the 2nd phase viz. Requirements Analysis to Software Acquisition activities if it has been decided during Feasibility Study to buy a readymade solution. Software Acquisition is not a standard phase of SDLC but important aspects as majority of companies are now buying readymade solutions. This also cites an important point that even for readymade solution, Requirements Analysis must be carried out so that an appropriate readymade solution will be selected,
- In the end, we have given some common roles that are found in today's industry (some roles are typical in IT industry e.g. programmers).

Sources:

- Ron Weber: Information Systems Control and Audit, Pearson Education, India, Third Impression, 2009.

- Valacich George Hoffer: Essentials of Systems Analysis & Design, PHI Pvt. Ltd., N. Delhi, India, 2004.
- Muneesh Kumar: Business Information Systems, Vikas Publishing House Pvt. Ltd., N. Delhi, India, 2001.
- Charles Parker, Thomas Case: Management Information Systems, Mitchell McGraw Hill, India, 1993.
- M. M. Pant: System Analysis, Data Processing and Quantitative Techniques, Pitambar Publishing Co. Pvt. Ltd., N. Delhi, India, 1999.
- Gordon B. Davis, Margrethe H. Olson, Management Information Systems, McGraw-Hill International Editions, 1984.
- Sujata Garg: Professional Approach to Management Information & Control Systems, Bharat Law House Pvt. Ltd., N. Delhi, India, 2005.
- Pankaj Jalote: An Integrated Approach to Software Engineering, Narosa Publishing House, Third Edition, 2005.
- Roger S. Pressman: Software Engineering- A Practitioner's Approach, McGraw-Hill, Sixth Edition, 2005.

Multiple Choice Questions

1. A collection of inter-related components or sub-systems is termed as:
   a. System
   b. SDLC
   c. Framework
   d. None of these

2. …….. is a deliverable of 'Requirements Analysis' phase of SDLC.
   a. Systems Design Report
   b. Requirements Analysis Report
   c. Framework
   d. None of these

3. …….. is a deliverable of 'Systems Design' phase of SDLC.
   a. Systems Design Report
   b. Requirements Analysis Report
   c. Framework
   d. None of these

4. Feasibility study may cover the following aspects of a project:
   a. Economic
   b. Technical
   c. Legal
   d. All of the above

5. A cost/benefit analysis should be done in the following study:
   a. Economic
   b. Technical
   c. Legal
   d. All of the above
6. UML stands for:
   a. Unique Modeling Language
   b. Unique Modeling Limit
   c. Unified Modeling Language
   d. Unique Mathematical Limit
7. DFD stands for:
   a. Data Flow Diagram
   b. Duplicate Functional Diagram
   c. Duplicate Flow Diagram
   d. None of these
8. DBA stands for:
   a. Data Base Administrator
   b. Data Business Administrator
   c. Duplicate Business Administrator
   d. None of these
9. Example of ERP Solution/s may include:
   a. ERP
   b. BAAN
   c. PeopleSoft
   d. All of the above
10. BRS stands for:
    a. Business Requirements Specification
    b. Basic Requirements Specification
    c. Business Requirements System
    d. Basic Requirements System
11. OCR stands for:
    a. Original Character Recognition
    b. Optical Character Recognition
    c. Optical Character Record
    d. Original Character Record

12. …….. is the phase where a detailed elicitation of users requirements to the maximum possible level is done.
    a. Requirements Analysis
    b. Systems design
    c. Testing
    d. None of these

13. The organized process or set of steps that needs to be followed to develop an information system is known as the:
    a. Analytical Cycle
    b. Design Cycle
    c. Program Specification
    d. System Development Life Cycle

14. Actual programming of software code is done during the …….. step in the SDLC.
    a. Maintenance and Evaluation
    b. Design Cycle
    c. Analysis
    d. Programming/ Construction

15. Documentation specialists generally provide the ……. for the new system.
    a. programs
    b. network
    c. analysis
    d. documentation

16. _____ spend most of their time in the beginning stages of the SDLC, talking with end-users, gathering information, documenting systems, and proposing solutions.
    a. Systems Analysts
    b. Project Managers
    c. Network Engineers
    d. Database Administrators

17. _____ is the process of translating a task into a series of commands that a computer will use to perform that task.
    a. Project Design
    b. Installation
    c. Systems Analysis
    d. Programming

18. The _____ determines whether the project should go forward.
    a. Feasibility Assessment

b. Opportunity Identification
c. System Evaluation
d. Program Specification

19. _____ design and implement database structures.
    a. Programmers
    b. Project Managers
    c. Technical Writers
    d. Database Administrators

20. _____ manage the system development, assign staff, manage the budget and reporting, and ensure that deadlines are met.
    a. Project Managers
    b. Network Engineers
    c. Graphic Designers
    d. Systems Analysts

Answers:

1. (a).	2. (b).	3. (a).	4. (d).	5. (a).	6. (c).
7. (a).	8. (a).	9. (d).	10. (a).	11. (b).	12. (a).
13. (d).	14. (d).	15. (d).	16. (a).	17. (d).	18. (a).
19. (d).	20. (a).				

2 Phases in Software Development

Learning Goals

- A clear understanding of all the phases of SDLC except the phase involving Feasibility Study and System Requirement Analysis, which we have already discussed in Chapter 1, and
- A brief discussion about the phases of Programming, Testing, Implementation and Post implementation.

Phase III- Systems Design

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Based on the requirements analysis done by development team, a system will be designed. In the last chapter, we have seen how Requirements Analysis is carried out by using Structured Analysis technique. The same technique is used for describing the design of the system. In this chapter we will focus on some other aspects of Systems Design along with topics described in the last chapter.

When we talk about designing of any object, 'what we really are talking is about the different parts of the objects', 'how they are constructed individually and combined to form the object'. Generally, the design is explained through the description and drawings of these parts and a combination drawing of the whole object. For example, design of a car gives design of it's engine, design of it's body, wheels, suspension and so on. Similar principles are applicable for giving design of a software system with some modifications of the technique. Thus, the system design will:

- describe the parts of the system and their interaction,
- set out how the system shall be implemented using the chosen hardware, software and network facilities (components of the system),
- specify the program and the database specifications and the security plan,
- specify the change control mechanism to prevent uncontrolled entry of new requirements.

User's involvement in this phase will be minimum, as the project has now moved from business aspects (Feasibility, and Requirements) to technical aspects (hardware, programming etc). However, approval of Systems Design Document by the

organization may be still necessary, which may be given by other third-party consultants or in-house technical competent people. Key design phase activities include:

- Developing system flowcharts to illustrate how the information shall flow through the system. We have already seen DFDs.
- Defining the applications through a series of data or process flow diagrams, showing various relationships from the top level down to the detail. We have seen E-R diagrams, data dictionaries etc.
- Describing inputs and outputs, such as screen design and reports. We shall describe this aspect later in this chapter.
- Determining the processing steps and computation rules for the new solution. We have seen Decision Tables / trees and Structured English through which logic and computational rules can be described.
- Determining data file or database system file design. E-R diagram and data dictionaries will lead to design of the table, as they describe entities, their structure and relationships among them. We need not go into the details of designing tables as it is beyond scope.
- Preparing the program specifications for the various types of requirements or information criteria defined. This topic is also beyond our current scope.
- Developing test plans for:
    - Unit (Program) Testing
    - Subsystem (Module) Testing
    - Integration (System) testing
    - Interface with other systems
    - Loading and initializing files
    - Security, backup and recovery
- Developing data conversion plans to convert data and manual procedures from the old system to the new system.

Requirements Analysis gives us various business processes and sub-processes. While eliciting these, the natural grouping of these processes is also described in the Requirements Analysis Document. Some regrouping may be necessary so as to suit design of the software. For example, in a banking application natural grouping would be all deposits, loans, clearing and so on. Some developers may keep this grouping and design his modules on this line only. Some other developers may design modules in a different grouping so as to suit the development and program organization such as Masters (for all deposits, loans etc), transactions (again for all deposits, loans etc), signature scanning and authorization and so on.

Steps involved in System Design

Thus, this phase deals with the way the proposed system can be transformed into a working model. The steps involved in this phase are:

1. Architectural Design
2. Designing of Data / Information Flow
3. Designing of Database
4. Designing of User Interface
5. Physical Design
6. Selection of Appropriate Hardware and Software

Architectural Design

Architectural design deals with the organization of applications in terms of hierarchy of modules and sub-modules, as described above. At this stage, we identify:

- Major modules e.g. Masters, Transactions, Reports etc.
- Function and scope of each module e.g. Master module will deal with all types of master data entry, modifications, viewing, printing etc.
- Interface features of each module e.g. to provide a feature of adding a missing master while user is entering a transaction
- Modules that each module can call directly or indirectly
- Data received from / sent to / modified in other modules.

The architectural design is made with the help of a technique called as functional decomposition wherein top level functions are decomposed (i.e. broken into) and inner-level functions are discovered. This process is continued till our context is met with.

Designing of Data / Information Flow

We have already discussed this issue in the previous chapter.

Designing of Database

We have seen what are entities and E-R diagrams in the last chapter. In designing database, entities are described in detail, with their structure. For example, for an Employee entity, obvious structure elements (also called as attributes, fields, columns) would be Employee ID, Name, Address, Date of Birth etc. Only those attributes which are of current interest with respect to the current system (or system module) are only considered. For example, in a Project allocation system, an employee's spouse's name may not be relevant but in HR system it may be relevant and hence included in entity structure. When design of all entities is over, they can be put in a repository to form a Data Dictionary so that, common entities across the

system can be used by other development team members. The design of database consists of 4 major activities

1. Conceptual Modeling – E-R Diagrams
2. Data Modeling
3. Storage Structure Design
4. Physical Layout Design

1. Conceptual Modeling

The entity structure and relationship leads to design of a database table. Let us consider a simple example.

Suppose that a Student (recollect this is a People type entity) is able to take one Course (a concept entity). For simplicity, let us assume that, one student can take only one course. (One course may have many subjects). Thus there are 2 entities : Student and Course and they are related in one-is-to-one manner. This is shown in the Fig. 2.1. Student's attributes of importance are Student ID, Name and Address and that of Course are Course ID and Course Name. For simplicity, only small numbers of attributes are considered here. In practice each entity may have hundreds of attributes and therefore hundreds of table columns. Entities are shown as rectangles, Attributes as ovals and relationship as diamond.

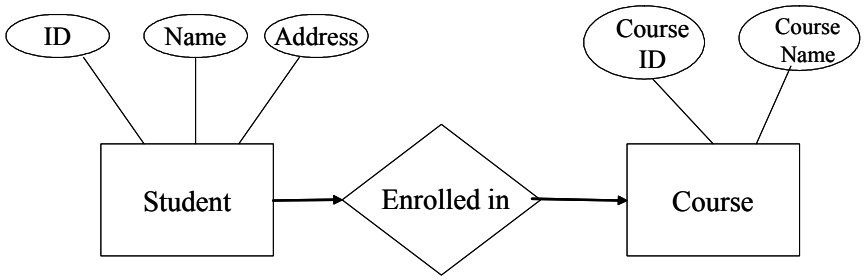


Fig. 2.1: Conceptual Modeling

Based on above E-R diagram, one can go for Data Modeling.

2. Data Modeling

Conceptual models have to be translated into data models, so that they can be manipulated using programming languages. Currently, the most popular data model is relational model, in which data is represented in tables. Every entity identified in the conceptual model is converted into a table, and the attributes are converted into fields (columns in the table). For above E-R diagram, we get following tables:

Student Master Table

Attribute Name (Column Name)	Attribute Type	Attribute Length
Student ID	Numeric	4
Name	Text	30
Address	Text	100

Course Master Table

Attribute Name (Column Name)	Attribute Type	Attribute Length
Course ID	Numeric	2
Course Name	Text	30

Student-Course Table

Attribute Name (Column Name)	Attribute Type	Attribute Length
Student ID	Numeric	4
Course ID	Numeric	2

Note:

1. Student ID is numeric with 4 digits length in this example. If number of students are in lacs then longer ID would be needed. Same is the case for Course ID, Student Name and Course Name attributes. Address is assumed to be 100 character long and not divided for simplicity, into first line of address, second line of address etc
2. The first 2 tables are master tables whereas the third table viz Student-Course Table is indicating the relationship between these two entities. A developer may add Course ID in Student Master table and discard the third table.

A record designing technique called as normalization is used to avoid update anomalies. Here it is being called as a record designing technique as a table contains set of records and we are designing these records' layout rather than table layout. However, table design and record design have the same meaning.

Database normalization is a technique for designing relational database tables to minimize duplication of information and to eliminate data anomalies. A table that is sufficiently normalized is less vulnerable to problems of duplication and data

anomalies. It is ensured that for multiple instances of the same information a single instance is designed to hold data in database tables.

Higher degrees of normalization typically involve more tables and create the need for a larger number of joins, which can reduce performance. Accordingly, more highly normalized tables are typically used in database applications involving many isolated transactions, while less normalized tables tend to be used in database applications that do not need to map complex relationships between data entities and data attributes (e.g. a reporting application, data warehousing application).

Database theory describes a table's degree of normalization in terms of normal forms of successively higher degrees of strictness. A table in third normal form (**3NF**) is also in second normal form (**2NF**); but the reverse is not always the case.

The following are the most common three normal forms:

In the F**irst Normal Form**, the repeating groups in a record are removed and made into a separate record. For example, in the purchase order record-item no, item description, unit of measure, quantity, rate and amount will repeat as many times as the number of items in the purchase order. This will obviously create inefficiency. Not only will space be wasted because of the duplication of repeating items in each record, but it will take a lot of work to maintain data consistency. For example, if there are 20 items on the purchase order, and one of the repeating fields, say the shipping address, changes, the change will have to be made in each of the 20 records. If the repeating fields (of which the shipping address is one) are made into a one table, and the order details kept in another, there will be only one record in the first table for the order, and space will be used only once for storing this information. Also, the change in the shipping address will only have to be made once, in this record, since the shipping address will no longer be in the second table.

In the **Second Normal Form**, a check is done on all non-key elements to ensure that they are dependent on the key. Continuing with the same example of purchase order record, in second normal form all supplier-related details will be moved to a separate supplier table. This is because supplier details are dependent on supplier code and not on the key of purchase order record, viz. purchase order no.

In the **Third Normal Form**, a check is done to ensure that no non-key element is transitively dependent on another non-key element. For example, discount may be a data element in purchase order. If the discount the supplier offers depends on the amount of purchase Order, it is better to keep discount in a separate table called supplier-purchase order amount.

3. Storage Structure Design

During this activity, a decision is made on how to store the data structure on some device. For example, in database there may be hundreds and thousands of tables. Similarly, in a system there may be multiple databases. A decision is taken on 'how these databases will be spread across a storage medium, say hard disk'. One can store all databases in one partition of a hard disk or divide them across different partitions and even different hard disks. Storing databases can become a fairly complex activity and operational performance of the system will depend on the structure of the data storage. Moreover, depending on performance, DBA may decide to change the storage structure.

4. Physical Layout Design of Data

Logical design is linked to previous systems development phases e.g. analysis. Similarly, E-R diagram i.e. conceptual data modeling is linked to the logical design. After the logical design, the next step is specifying the physical design. During this phase, the logical design team will be expanded to include variety of experts such as database designers, security experts, network experts etc. The purpose of physical design is to specify all the technological characteristics of the system so that building of the system on technology platform is possible. Physical design includes three steps:

1. Designing Physical Files and Databases: describes how data will be stored and accessed in secondary computer storage and how quality of data will be ensured.
2. Designing System and Program Structure: describes various programs and program modules that correspond to the data flow and other aspects developed in earlier life cycle stages.
3. Designing Distributed Processing Strategies: describes how system will make data and processing available to users on computer networks within the existing capacity of network setup.

A brief description of computer files and file structure is presented below:

Types of Files:

The following are the main types of files:

Master File: This type of file stores master data or fixed data about an entity; e.g. a customer master file will store data about customers such as customer number, name, address, PAN number and so on. The data in master files changes less frequently or may not change at all during the life span of the entity.

Parameter File: This file contains data which is used during processing of transaction data. In early days this type of data was classified as Master data only. Some times this type of data is also referred as lookup file data. E.g. rate of interest for different banking products, rates of depreciation, unit of measures of materials in an inventory system. The data in this file can change more frequently than the master data.

Transaction file : This file contains data captured about real life events happening in day-to-day life. E.g. cash withdrawn from a bank, delivery of material received, bill submitted to a client. All of these events have data associated with them which changes with every transaction. This data is captured and kept in transaction file to be processed. Depending upon a system, this file can hold data for a very short period or for longer duration. E.g. there may be a transaction file holding only a day's data.

Work File or Temporary File: This file contains data which have been retrieved from transaction and other files (such as master or parameter) for further processing. The program holds data temporarily in this type of files and after the processing is over it is written to final output files or printed. Good programs generally remove data from the work files when desired processing is over.

Protection File or Audit Trails: This type of file contains before and / or after images of the data before or after the processing cycle is over. The purpose of this file is to provide data through different stages of data processing and also to store the chronological events of data processing for future reference or analysis. IS auditors need to ensure that every system should have this type of file and is storing data in correct details.

History File: This type of file contains archival data which is useful to perform trend analysis or for data warehousing kind of applications. Organizations may opt to keep types history files such as yearly, location wise, customer wise and so on.

Report File: This file can be of two types. One, which contains exact print image (i.e. along with cosmetic formatting for the report to be printed) of the final results of data processing and the other which can be used further to be uploaded in some other applications like spreadsheet.

File Organization Methods:

All the above described files can be stored on variety of secondary storage media such as hard disk, floppies, CDs, tapes and so on. The following brief describes 'how these files organize the data inside them'.

Sequential File Organization: This is also known as text file or flat file organization. The data records in this file are stored in sequential order as it is being written to the

file. In order to read say, 1000th record in the sequential file, it is necessary to read or skip first 999 records of the file.

Indexed File Organization: In this type of file, data records are either stored sequentially or non-sequentially and an index is created (in another file) which allows application software to locate a data record from the index and read the record directly from the file without the need to go thru sequentially. Depending on the need one data file can have multiple index files so that data can be searched from the data file quickly. E.g. a customer master file storing customer data can have index files based on customer number, PAN number, date of birth etc. The index file itself can be read sequentially or with different other techniques in order to quickly locate an index entry and then to read the corresponding data record.

Hashed File Organization: In this type of file, address of each data record is determined based on a hashing algorithm which converts a primary key value to a record address.

It is necessary to note that, a relational database application (such as Oracle or Microsoft SQL), stores all the data files (user data files, query program files, input screen files, report file, schema files, audit trail files etc) in a single file at operating system level. Only when you load the particular database application, these files are made available to users for viewing or to programs for processing. Also, a database file can be partitioned across multiple storage media. All the partitions together form the complete database. Ina replicated database, on the other hand, the same database is duplicated on different hard disk which can serve as alternative database. The database management system optimizes the partitioning of database so that the performance is satisfactory.

Designing of User Interface

Users need to enter the data of their transactions and store this data into database tables. For this purpose, developer designs screens and dialogue boxes. Generally, screens are used for entering transaction data whereas dialogue boxes are used for entering other useful data such as user IDs, passwords, confirmations for user actions, selection of folders etc.

Organizations develop standards for user interface based on organization's experience, user needs and aesthetic look requirements. Some of the important aspects looked into while designing menus, screens and dialogue boxes are given below..

1. Menu navigation should be easy and promote the users to use the software
2. Screens with soothing foreground and background colors should be designed

3. Place for company logos, dates etc should be uniform throughout the screens
4. For multipage screen layout, it is better to have tabs with page numbers indicating on which page the user is
5. Mandatory fields should be indicated explicitly e.g. with red color or blinking
6. If system response time after a user action is more, it should be clearly displayed intermittently on screen
7. Developers should design screen by keeping in mind computer awareness level of users. E.g. help desk staff may require to work on more than one software (and therefore more screens) simultaneously, so several page tabs on one screen would be helpful rather than going thru several screens.

Generally, developers design prototype (dummy screens) and take approval of users prior to finalizing the interface.

In some good readymade software packages, users are allowed to modify some features (e.g. foreground and background colors) of user interfaces by changing the configuration settings and storing them as “Favorites” e.g. SAP.

Physical Design and selection of appropriate hardware and software

So far, we have covered the logical aspects of system design. This logical design needs to be ultimately mapped or implemented on a Physical Design. For instance, we have seen that the table design which we have done, will be stored in some folder, sub-folder on hard disk and then it will be mapped to a physical track-sectors on the hard disk.

In this aspect of Physical Design, we have to select and finalize hardware, operating system, database management system and any other software needed. Generally, following types of components need to be selected and finalized. Even though, much standardization is observed in these areas, their sizing and configuration setting etc is very important.

1. Hardware – e.g. hardware for servers, desktops etc. e.g. IBM, HP, Dell etc international brands are selected for servers and desktop machines. Hardware sizing i.e. selecting CPU type and numbers, selecting hard disks size and number (for mirroring or backups), selecting tape drives etc is very critical, especially, in ERP systems
2. Power Systems – such as UPS, generators, line conditioners etc. In case of UPS, it is necessary to know the electrical power consumption of various hardware equipments (servers, desktops, scanners, printers etc) and to select appropriate UPS.
3. Networking and telecommunication equipment – such as hubs, switches, routers, repeaters, networking cables and other terminating equipment.

4. Operating system – such as different types of Windows (XP, Windows 2003 etc), Unix or Linux. Developers may design application software that runs on these three operating systems, however, this may not be always the case.
5. RDBMS – such as Oracle or Microsoft SQL Server or MySQL etc. Here again, some developers develop the application software to run on any RDBMS but may not always be the case.
6. Web server software – for web based systems server will have this software which will interact with database and application software which are loaded on servers (called as database and application servers). E.g. Internet Information Server (IIS), Apache etc.
7. Transactions processing software and message queuing software – These are classified as Middleware layer of software. Front-end layer sends transaction messages to middle-ware which in turn stores and forwards these to back-end layer. Their main function is to process a transaction and/or queue up transactions for further processing.
8. Client software – This software will reside on desktop or client machine. Depending upon type of system, a client software may have to be separately installed (either by downloading from web site or through a CD). The client software will be connected to Application software when user invokes it.

In all the above components, it is necessary to select correct sizes, versions and configurations so that application software can be run satisfactorily. After this phase, the project moves into the Programming Phase.

Phase IV- System Development: Programming Methods, Techniques and Languages

The Development Phase takes the detailed design developed in the Design Phase and begins with coding by using a programming language. The responsibility of this phase is primarily that of the Programmers. The following are the key activities performed during this phase.

1. Coding and developing program and system level documents,
2. Testing and debugging continuously for improvements in program developed,
3. Developing programs for conversion of the data in the legacy system to new system,
4. Formulating the procedures for the transition to the new software from earlier software,
5. Training the selected users on the new system,
6. In case of vendor supplied software, documenting the modifications carried out to ensure that future updated versions of the vendor's code can be applied.

A good program should have the following characteristics:

- **Accuracy:** Accuracy does not only mean that the program should do what it is supposed to do; it should also not do what it is not supposed to do. In fact, the second part becomes more challenging for quality control personnel and auditors
- **Reliability:** The program continues to deliver the functionality, so long as there are no changes required in functionality. Since programs do not wear and tear, reliability is taken for granted. However, poor setting of parameters and hard coding some data, which is of temporary value could result in the failure of a program after some time,
- **Robustness:** In normal circumstances, most programs work. It is the extraordinary circumstances that really differentiate the good programs from the bad ones. This means that the programmer has anticipated even the least likely situations and provided safeguards for them, so that mishaps can be avoided, This process of taking into consideration all possible inputs and outcomes of a program is also known as error / exception handling process,
- **Efficiency:** Performance should not be unduly affected with the increase in input volumes.
- **Usability:** A user-friendly interface and easy-to-understand user document is necessary for any program.
- **Readability:** The program should be easy to maintain even in the absence of the programmer who developed it. The inclusion of comment lines is extremely useful.
- **Maintainability:** If a program has to be maintained by a programmer other than the one who developed it, it should be developed the modular way.

Programming Methods & Techniques

The following are the main programming methods and techniques:

1. Adoption of the Program Coding Standards

To improve the quality of the software developed and its capability for future maintenance, program coding standards should be used.

- Coding standards are essential for reading and understanding the program code in simple and clear manner.
- Coding standards may include source code level documentation, methods for data declaration, and technique for input / output.
- Coding standards serves a method of communication between teams, amongst the members of the teams and the users, thereby working as a good control.

- Coding standards minimize the system development setbacks due to programmer turnover.
- Coding standards help in efficient program modification and maintenance.

2. Online Programming Facilities

To facilitate the effective use of the structured programming techniques, an online programming facility should be allowed. The following are the advantages of this technique:

- This technique allows programmers more flexibility in coding and compiling the programs with a remote computer.
- By the use of this facility, the programmers can enter, modify and delete program codes as well as compile and store the programs on the computer
- This facility, in general, allows a faster development of the programs
- This approach can lower the development costs, maintain rapid response time and effectively expand the programming resources.

In addition, following are the potential weakness of this facility:

- Control of multiple versions of the programs
- Control to unauthorized access and updating of programs
- Control on proper updating of changes

In today's industry, mostly on-line programming facilities are available. On-line programming is also called as Integrated Development Environment. This facility allows a programmer to write a program (program consisting of screen, dialogue boxes, processing logic, reports etc), compile it, test it, debug and modify it. This facility is also called as Programmers Work Bench as a programmer will use it throughout his day of work carry out his work of programming.

Programming Languages

A programming language is a machine-readable artificial language designed to express computations that can be performed by a machine, particularly a computer. Programming languages can be used to create programs that specify the behavior of a machine, to express algorithms precisely, or as a mode of human communication.

As many software experts point out, *the complexity of software is an essential property, not an accidental one.* This inherent complexity is derived from the following four elements:

- The complexity of the problem domain
- The difficulty of managing the development process
- The flexibility possible through software

- The problems of characterizing the behavior of discrete systems

The sweeping trend in the evolution of high- level programming languages and the shift of focus from-programming-in-the-small to programming-in-the-large has simplified the task of the software development team. It also enables them to engineer the illusion of simplicity. The shift in programming paradigm is categorized into the following:

- Monolithic Programming
- Procedural Programming
- Structured Programming
- Object Oriented Programming

Like the computer hardware, programming languages have been passing through evolutionary phases or generations. It is generally observed that most programmers work in work in one language and use only one programming style. They program in a paradigm enforced by the language they use. Frequently they may not have been exposed to alternate ways of solving the problem and hence, they will have difficulties in exploiting the advantages of choosing a style more appropriate to the problem at hand. Programming style is defined as a way of organizing the ideas on the basis of some conceptual model of programming and using an appropriate language to write efficient programs. Five main kinds of programming styles are listed in Table 2.1 with the different types of abstraction they employ.

Programming Style	**Abstraction Employed**
Procedure-oriented	Algorithms
Object-oriented	Classes and Objects
Logic-oriented	Goals, often expressed in predicate calculus
Rule-oriented	If-then-else rules
Constraint-oriented	Invariant relationship

Table 2.1: Types of Programming Paradigms

There is not a single programming style that is best suited for all kinds of applications. For example, procedure-oriented programming would be best suited for the design of knowledge base, and logic-oriented programming would be best suited for a hypotheses derivation .The object-oriented style is best suited for a wide range of applications; indeed, this programming paradigm often serves as the architectural framework in which other paradigms are employed. Each one of these styles of programming requires a different mindset and a different way of thinking about the problem, based on their own conceptual framework.

Monolithic Programming

The programs, written in these languages exhibit relatively flat physical structure as shown in Figure 2.2. They consist of only global data and sequential code. Program flow control is achieved through the use of jumps and the program code is duplicated each time it is to be used, since there is no support of the subroutine concept and hence, it is suitable for developing small and simple applications. Practically, there is no support for data abstraction and it is difficult to maintain or enhance the program code.

Examples: Assembly language and BASIC

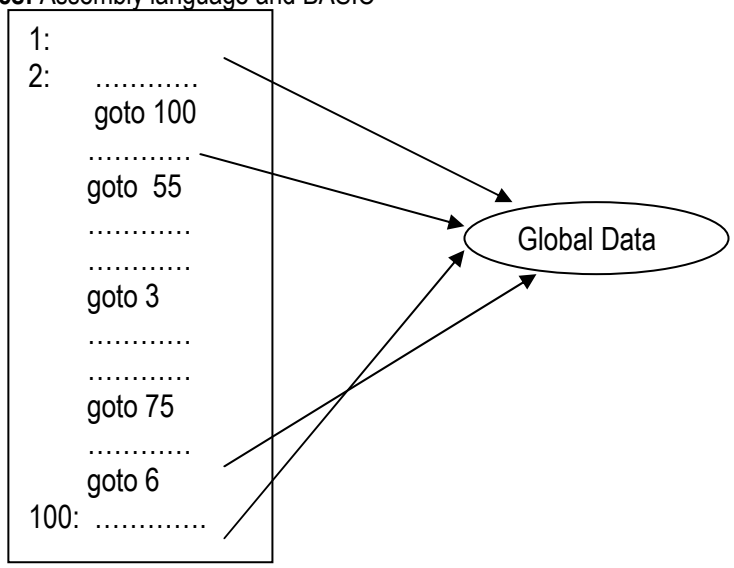


Fig. 2.2: Monolithic Programming

Procedural Programming

Programs were considered as important intermediate points between and the computer in the mid-1960s. Initially, software abstraction achieved through procedural abstraction grew directly out of this pragmatic view of software. Subprograms were originally seen as labor –saving devices but very quickly appreciated as a way to abstract program function s as shown in Fig. 2.3.

The following are the important features of procedural programming:

- Programs are organized in the form of subroutines and all data items are global
- Program controls are through jumps (gobos) and calls to subroutines
- Subroutines are abstracted to avoid repetitions

- Suitable for medium sized software applications
- Difficult to maintain and enhance the program code

Examples: FORTRAN and COBOL

Subprograms

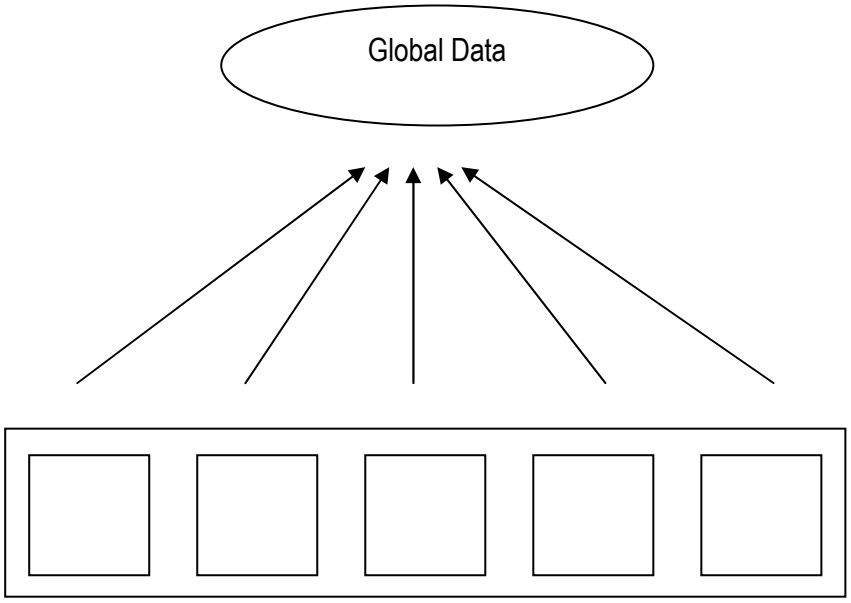


Fig. 2.3: Procedural Programming

Structured Programming

Structured programming has evolved as a mechanism to address the growing issues of programming in-the-large. Larger programming projects consist of large development teams, developing different parts of the same project independently. The usage of separately compiled modules (algorithmic decomposition) was the answer for managing large development teams (see Fig. 2.4). Programs consist of multiple modules and in turn, each module has a set of functions of related types.

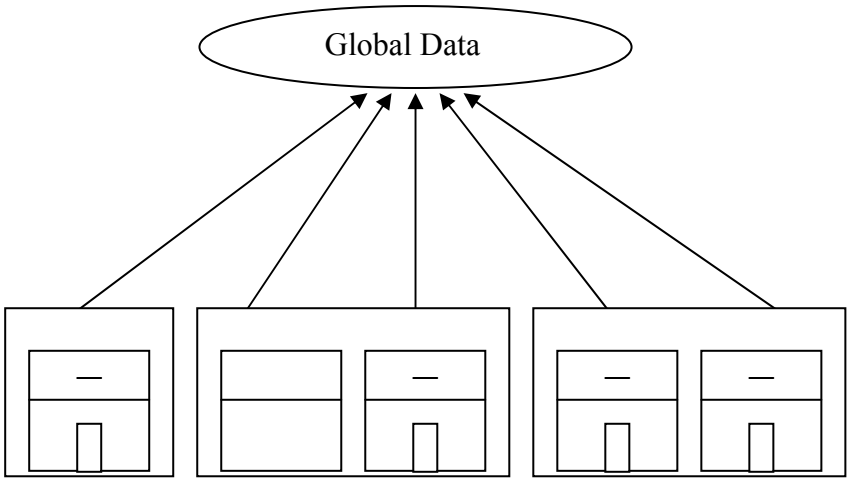


Fig. 2.4: Structured Programming

The following are the important features of structured programming:

- Emphasis on algorithm rather than data
- Programs are divided into individual procedures that perform discrete tasks
- Procedures are independent of each other as far as possible
- Procedures have their own local data and processing logic
- Parameter passing facility between the procedures for information communication
- Controlled scope of data
- Introduction of the concepts of user defined data types
- Support for modular programming
- Projects can be broken up in to modules and programmed I dependently
- Scope of data items is further controlled across modules
- A rich set of control structure are available to further abstract the procedures
- Co-ordination among multiple programmers is required for handling the changes made to mutually shared data items
- Maintenance of a large software system is tedious and costly

Examples: Pascal and C

Object Oriented Programming

The easy way to master the management of complexity in the development of a software system is through the use of data abstraction. Procedure abstraction is suitable for the description of abstract operations, but it is not suitable for the description of abstract objects. This is a serious drawback in many applications since, the complexity of the data objects to be manipulated contribute substantially to the overall complexity of the data objects. This is serious drawback in many applications since, the complexity of the data objects to be manipulated contribute substantially to the overall complexity of the problem. The emergence of data-driven methods provides a disciplined approach to the problems of data abstractions in algorithmic oriented languages. It was resulted in the development of object-based language supporting only data abstraction. Object-based languages do not support features such as inheritance and polymorphism, which will be discussed later. depending on the object features supported, the languages are classified into two categories:

1. Object-Based Programming Languages
2. Object-Oriented Programming Languages

Object-based programming languages support encapsulation object identity without supporting important features of OOP languages such as polymorphism, inheritance, and message based communication. Ada is one of the typical object-based programming languages:

Object-based languages = Encapsulation + Object Identity

Object- oriented languages incorporate all the features of object-based programming languages along with inheritance and polymorphism. Therefore, an object-oriented programming language is defined by the following statement:

Object-oriented language= object based features + Inheritance + Polymorphism

The topology of object-oriented programming languages is shown in Figure 1.6 for small, moderate, and large projects. The *modules* represent then physical building blocks of these languages; a module is a logical collection of classes and object, instead of subprograms as I the earlier languages. Thus making classes and objects as the fundamental building blocks of OOPs.

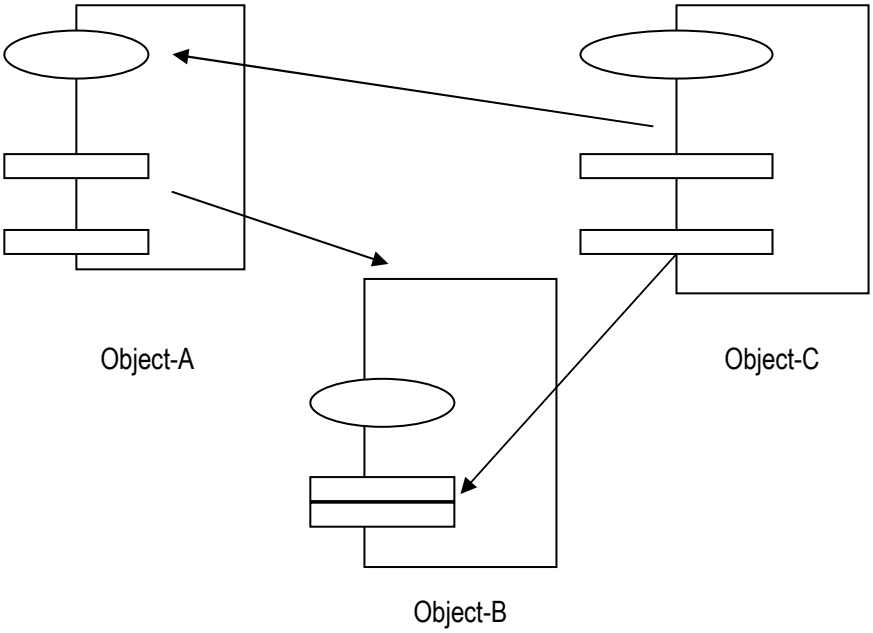


Fig. 2.5: Object Oriented Programming

Object-oriented programming is a methodology that allows the association of data structures with operations similar to the way it is way it is perceived in the human mind. They associate a specific set of actions with a given type of object and actions are based on these associations.

The following are the important features of object-oriented programming:

- Improvement over the structured programming paradigm
- Emphasis on data rather than algorithm
- Data abstraction is introduced in addition to procedural abstraction
- Data and associated operations are unified into a single unit, thus the object are grouped with common attributes, operations and semantics
- Programs are designed around the data being operated, rather than operations themselves (data decomposition rather than algorithmic decomposition)
- Relationships can be created between similar, yet distinct data types

Examples: C++, Smalltalk, Eiffel, Java, etc.

Choice of Programming Language

With several programming languages being available today, the choice of language is the first issue that needs to be addressed. The following are among the most important criteria

- Application area
- Algorithmic complexity
- Environment in which software has to be executed
- Performance consideration
- Data structure complexity
- Knowledge of software development staff
- Capability of in-house staff for maintenance

In the current information technology context, the following types of technologies are being implemented in Indian industry. A detailed discussion is beyond the scope of this module. Kindly refer other chapters for more elaboration.

Client – Server and distributed data

Distributing data over networks and client/server system was very common during 90s. Even today several systems are using this type of technology. “Client” in this context is the machine (usually a PC) where user is doing his/her day-to-day work. “Server” is the machine where the data used by several users is stored. The server also stores all the computer programs. Users run the computer programs by connecting to server and calling required program, generally, through a menu driven system. In this type of system, data can also be stored on client machines together with server machines. Novell Netware and Unix systems were very popular client-server technologies. The data and software instructions travel from client to server and back. The data on server is usually stored in database systems. User communicates with server through several forms or dialogue boxes to supply and retrieve data from server. In order to speed up the traveling time between server and client, designer may opt to store some data on user machines. Thus what you get is a distributed data and distributed processing environment. (data processing is taking place on server as well as client)

Integrating software programs for all business functions through Enterprise wide Resource Planning led to ERP system which is based on client-server technology.

The choice of programming languages and database in client-server technology depended upon how to efficiently use the connectivity between client and server for instruction and data flowing between server and client.

In this technology, a user (a user program) connects to the server database for "pulling" or "pushing" the data to the database. The connection between client and server will be needed only to the extent to complete this data flow. Once user program pulls or pushes the data, the connection can be broken logically(the physical connection will obviously continue to remain connected). This technology is also known as 2-tier architecture, tier one being the client system and tier two being the server system. Remember that, software system may be an integrated system (such as ERP) or collection of different systems such as payroll, accounting, inventory and so on under client server technology.

Web Technology

With the arrival of Internet, the things changed dramatically and they continue to change. In early client-sever systems, client and server both were on LAN i.e. directly connected through a cable running between client and server. With Internet and WAN technologies, clients and servers can be located anywhere on the globe. This global aspect puts a serious hurdle in the speed at which client gets connected to server and pushes and pulls the data from server due to slow connection speeds. This forced the designers to design systems differently. This gave a way to web technologies. Again, a detailed discussion about web technology is not done in this chapter, but an over view is presented in the context of this chapter. In web-based technology, additional tiers were needed through which client will communicate to server for pulling and pushing data. Usually, 3 tiers are deployed however n-tier architectures are also becoming common.

In a multi-tiered approach, a client gets connected to a web-server which manages connections between client system and server system efficiently by a very complex algorithm. Web server after receiving client's request (for connection to server for pulling or pushing data), connects client to the application server which is nothing but software programs implemented on a server machine. Now, Application server will connect to a database server if at all data pull or data push is needed. Web server is responsible for maintaining the connectivity between clients (spread all over across globe) and various servers. Thus, we have client tier, web server tier, application server tier and database server tier as 4-tierd system. Note that, all or some of these tiers can be implemented on physically same or different hardware.

Making a choice of programming language and database will depend on which technology is planned to be selected for the business. If a system is based only on LAN, a non-web-based client-server technology can be selected. This does not mean that for a WAN, web-based system is unavoidable. SAP, for example, is a WAN based client server system which is a non-web-based system. However, connectivity between client and server should be efficient in such case. Current industry trend is

to have a hybrid of web-based multi-tier client-server system supported by a back-end non-web-based client server architecture.

Coding Style

Programming can be improved by following the guidelines given below:

- Consistent and meaningful names for programs, variables and the like to improve understandability and maintainability. A trial balance program can be called trialbal rather than Fin32. Similarly, employee number can be referred to as emp_no and supplier number can be referred to as supp_no. This ensures consistency in naming conventions.
- Modular programming makes maintainability easy
- Languages like C permit more than one instruction to be written in a line. But to improve readability 'one instruction per line' standard must be maintained.
- Comment lines improve the understandability of code. These indicate the function of each logical block. Adequate comment lines should be included in the program
- Indentation, parenthesis and the like improve readability of the code if the logic is complicated, clear formatting will make it easy to understand.

Program Debugging

Debugging is the most primitive form of testing activity. Programmers usually debug their programs while developing their source codes by activating the compiler and searching for implementation defects at the source code level. Bugs (Defects) are always the result of faulty code structure, misunderstanding of the programming language features, incoherent naming or wrongly spelt programming statements.

The need for extensive debugging is often an indication of poor workmanship. Debugging activities should be kept at a minimum and the programmer should not entirely be dependent on the compilers but should also take help of the debugging software. Debugging software tools assist the programmer in fine tuning, fixing and debugging the program under development. These tools fall in the following three categories.

- **Logic Path Monitors:** provides clues to the programmer on logic errors by reporting on the sequence of events achieved by the program
- **Trace:** This lists the changes in selected variables at different stages of the program.
- **Memory Dumps:** provides a picture of the internal memory content at the point where the program has abruptly ended, providing the clues to the programmer on the inconsistencies in data and parameter values.

- **Output Analyzer:** checks the accuracy of the output which is the result of processing the input through that program by comparing the actual results with the expected results.

Baselines

Before starting the discussion of next phase in SDLC, it is worthwhile to discuss about baselining. In the context of software development, baselining every work products at every stage of the SDLC phase is very crucial.

A baseline is a software configuration management concept that helps control change without seriously impeding justifiable change. The IEEE standard defines basline as "A specification or product that has been formally reviewed and agreed upon, that thereafter serves as the basis for further development, and that can be changed only through formal change control procedures."

It is important to note from the above definition that,

- A software product (also called as work product) is a deliverable to be accepted by customer e.g. a feasibility study document is a work product. In this context any process generally will result into a product which can be distinctly identified as a result of the tasks executed in that process. E.g. if an analyst discusses about a bank deposit process with a user and documents the same adequately, then this document explaining bank deposit process is a product which is to be evaluated by the user.
- Also note that till the product is not formally reviewed and accepted, informal changes may be done to the product. But once a formal acceptance is given to the product then it is said that a baseline has been established. After baslining, no informal change will be accepted and done to the product. This is to control the configuration of the product at the baselining stage. Note that, this product may not be the final deliverable to the user. E.g. user may not be interested in a feasibility document but rather in the resulting system.
- Change control procedure will have to be defined in order to formally change the baselined products.

Baselining a work product is also known as freezing the work product. The term indicates that the product is "frozen" and it's frozen status cannot be changed without formal change management procedure. Thus the concept of baselining is a generic and can be applied in other areas also.

Baselining can be done throughout the SDLC life cycle stages. If it is used for program development phase, then programs (source and executable) will be the work products. Once baselined, these programs will be frozen in a software library

(either manually controlled library but generally auto controlled through version control software). If any change is to be made to these baselined programs, it should only be done through a formal procedure. Since a set of programs become a software module, baselining can be applied to completed modules as well. Work product in this case will be the module which will be set of programs which are also controlled. Version numbering will be done to a completed system which is released to the user.

Some of the Software Configuration items (not in any particular order) which are baselined is given below:

1. System specifications document
2. Software project plan
3. Software requirements specifications
4. User manual
5. Design specification
   5.1. Data design specification
   5.2. Architectural design specification
   5.3. Interface design specification
6. Source code
7. Test specification

Notice in the above discussion that controlling baselined products could be a manually controlled activity. If it is a manually controlled activity, then a Scope Creep can happen in a baselined product. A **Scope creep** (requirement creep or feature creep) refers to uncontrolled changes happening in products. This can occur when scope of a project or a product is not properly defined, documented, or controlled (meaning baselined). This is an undesirable phenomenon and needs to be controlled through formal change management procedures. Scope creep can be a result of:

- poor change control procedure
- lack of proper initial identification of what is required in a product
- weak management
- poor communication between parties

Scope creep is a risk in most projects and needs to be identified and managed. It will often result in cost overrun and/or time overrun.

The next phase in SDLC is Software Testing.

Phase V- System Testing

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the

context in which it is intended to operate. Software Testing also provides an objective, independent view of the software to allow the business to appreciate and understand the risks at implementation of the software. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs. It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics. Software Testing, depending on the testing method employed, can be implemented at any time in the development process, however the most test effort is employed after the requirements have been defined and coding process has been completed.

Objectives of Testing

The following are the main objectives of testing:

- To uncover as many as errors (or bugs) as possible in a given timeline including errors in:
  - requirements from requirement analysis,
  - design documented in design specifications,
  - coding (implementation),
  - system resources and system environment,
  - hardware problems and their interfaces to software.
- To demonstrate a given software product matching its requirement specifications,
- To validate the quality of a software testing using the minimum cost and efforts,
- To generate high quality test cases which should have high probability of finding an error, perform effective tests, and issue correct and helpful problem reports.

Levels of Testing

Every software normally goes through the following levels of tests:

- Unit Testing
- System Testing

Unit Testing

In computer programming, Unit Testing is a software verification and validation method where the programmer gains confidence that individual units (i.e. individual programs or functions or objects) of source code are fit for use. A unit is the smallest testable part of an application. In procedural programming a unit may be an individual program, function, procedure, etc., while in object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class or

derived/child class. Unit testing can be done by something as simple as stepping through code in a debugger; modern applications include the use of a test framework such as xUnit.These tests ensure that the internal operation of the program performs as per the specification. There are five categories of tests that a programmer typically performs on a program unit:

- **Functional Tests:** These tests check 'whether programs do what they are supposed to do or not'. The test plan specifies operating conditions, input values, and expected results, and as per this plan programmer checks by inputting the values to see whether the actual result and expected result match.
- **Performance Tests:** These should be designed to verify the response time, the execution time, the throughput, primary and secondary memory utilization and the traffic rates on data channels and communication links.
- **Stress Tests:** Stress testing is a form of testing that is used to determine the stability of a given system or entity. It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results. Stress testing may have a more specific meaning in certain industries.These tests are designed to overload a program in various ways. The purpose of a stress test is to determine the limitations of the program. For example, during a sort operation, the available memory can be reduced to find out whether the program is able to handle the situation.
- **Structural Tests:** These are concerned with examining the internal processing logic of a software system. For example, if a function is responsible for tax calculation, the verification of the logic is a structural test.
- **Parallel Tests:** By using the same test data in the new and old system, the output results are compared.

Types of Unit Tests

The following are the main types of unit testing:

Static Analysis Tests

Some important static analysis tests are:

- **Desk Check**: This is done by the programmer himself. He checks for logical syntax errors, and deviation from coding standards.
- **Structured walk-through**: The application developer leads other programmers through the text of the program and explanation
- **Code inspection**: The program is reviewed by a formal committee. Review is done with formal checklists. The procedure is more formal than a walk-through.

Dynamic Analysis Tests

Black Box Testing

Black box testing takes an external perspective of the test object to derive test cases. These tests can be functional or non-functional, though usually functional. The test designer selects valid and invalid inputs and determines the correct output. There is no knowledge of the test object's internal structure. This method of test design is applicable to all levels of software testing: unit, integration, functional testing, system and acceptance. The higher the level, and hence the bigger and more complex the box, the more one is forced to use black box testing to simplify. While this method can uncover unimplemented parts of the specification, one cannot be sure that all existent paths are tested. If a module performs a function, which is not supposed to, the black box test does not identify it.

White Box Testing

White box testing uses an internal perspective of the system to design test cases based on internal structure. It requires programming skills to identify all paths through the software. The tester chooses test case inputs to exercise paths through the code and determines the appropriate outputs. In electrical hardware testing, every node in a circuit may be probed and measured; an example is in-circuit testing (ICT).

Since the tests are based on the actual implementation, if the implementation changes, the tests probably will need to change, too. For example ICT needs updates if component values change, and needs modified/new fixture if the circuit changes. This adds financial resistance to the change process, thus buggy products may stay buggy. Automated optical inspection (AOI) offers similar component level correctness checking without the cost of ICT fixtures; however changes still require test updates.

While white box testing is applicable at the unit, integration and system levels of the software testing process, it is typically applied to the unit. While it normally tests paths within a unit, it can also test paths between units. After obtaining a clear picture of the internal workings of a product, tests can be conducted to ensure that the internal operation of the product conforms to specifications and all the internal components are adequately exercised. In other words, the focus is on structural correctness. This is called white box testing. White box test will not verify if the module performs all the functions that it is supposed to perform.

Gray Box Testing

In recent years, the term grey box testing has come into common usage. Gray box testing is a software testing technique that uses a combination of black box testing and white box testing. Gray box testing is not black box testing, because the tester does know some of the internal workings of the software under test. In gray box testing, the tester applies a limited number of test cases to the internal workings of the software under test. In the remaining part of the gray box testing, one takes a black box approach in applying inputs to the software under test and observing the outputs.

Gray box testing is a powerful idea. The concept is simple; if one knows something about how the product works on the inside, one can test it better, even from the outside. Gray box testing is not to be confused with white box testing; i.e. a testing approach that attempts to cover the internals of the product in detail. Gray box testing is a test strategy based partly on internals. The testing approach is known as gray box testing, when one does have some knowledge, but not the full knowledge of the internals of the product one is testing.

In gray box testing, just as in black box testing, we test from the outside of a product, just as you do with black box, but you make better-informed testing choices because we are better informed; because we know how the underlying software components operate and interact.

Integration / Interface Testing

The objective is to evaluate the connection of two or more components that pass information from one area to another. This is carried out in the following manner:

- **Bottom-up Integration:** Bottom-up integration is the traditional strategy used to integrate the components of a software system into a functioning whole. It consists of unit testing, followed by sub-system testing, and then testing of the entire system. Unit testing has to uncover errors in the individual modules of the system. Sub-system testing checks the operation of the interfaces between modules in the subsystem. System testing is concerned with control flow, recovery procedures, throughput, capacity and performance characteristics of the entire system. Bottom-up testing is easy to implement. This is because at the time of module testing, tested subordinate modules are available. The disadvantage, however is that testing of major decision / control points is deferred to a later period.
- **Top-down Integration:** Top-down integration starts with the main routine, and stubs are substituted, for the modules directly subordinate to the main module.

An incomplete portion of a program code that is put under a function in order to allow the function and the program to be compiled and tested, is referred to as a stub. A stub does not go in to the details of implementing details of the function or the program being executed.

Once the main module testing is complete, stubs are substituted with real modules one by one, and these modules are tested with stubs. This process continues till the atomic modules are reached. Since decision- making processes are likely to occur in the higher levels of program hierarchy, the top-down strategy emphasizes on major control decision points encountered in the earlier stages of a process and detects any error in these processes. The difficulty arises in the top-down method, because the high-level modules are tested, not with real outputs from subordinate modules, but from stubs.

- **Regression Tests:** Each time a new module is added as part of integration testing, the software changes. New data flow paths are established, new I/O may occur and new control logic is invoked. These changes may cause problems with functions that previously worked flawlessly. In the context of the integration testing, the regression tests ensure that changes or corrections have not introduced new errors. The data used for the regression tests should be the same as the data used in the original test.

Types of System Testing

System testing is a process in which software and other system elements are tested as a whole. System testing begins either when the software as a whole is operational or when the well defined subsets of the software's functionality have been implemented. The test target in this case is the whole implementation model for the system. The purpose of system testing is to ensure that the new or modified system functions properly. These test procedures are often performed in a non-production test environment. The following types of testing might be carried out:

Recovery Testing: This is the activity of testing 'how well the application is able to recover from crashes, hardware failures and other similar problems'. Recovery testing is the forced failure of the software in a variety of ways to verify that recovery is properly performed. Recovery testing should not be confused with reliability testing, which is tries to discover the point at which failure occurs. Some examples of recovery testing are as follows:

- While the application running, suddenly restart the computer and after that check the validness of application's data integrity;

- While application receives data from the network, unplug and then in some time plug-in the cable, and analyze the application ability to continue receiving of data from that point, when network connection disappeared;
- To restart the system while the browser will have definite number of sessions and after rebooting check, that it is able to recover all of them.
- Checking the ability of recovery of the system after the failure of hardware or software.

Security Testing: This is the Process to determine that an IS (Information System) protects data and maintains functionality as intended or not. The six basic security concepts that need to be covered by security testing are: confidentiality, integrity, authentication, authorization, availability and non-repudiation. This testing technique also eensures the existence and proper execution of access controls in the new system.

Stress or Volume Testing: Stress testing is a form of testing that is used to determine the stability of a given system or entity. It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results. Stress testing may have a more specific meaning in certain industries. This is performed by testing the application with large quantity of data during peak hours to test its performance.

Performance Testing: In the computer industry, software performance testing is used to determine the speed or effectiveness of a computer, network, software program or device. This process can involve quantitative tests done in a lab, such as measuring the response time or the number of MIPS (millions of instructions per second) at which a system functions. Qualitative attributes such as reliability, scalability and interoperability may also be evaluated. Performance testing is often done in conjunction with stress testing. This testing technique ccompares the new system's performance with that of similar systems using well defined benchmarks.

Final Acceptance Testing

Final Acceptance testing is conducted when the system is just ready for implementation. During this testing, it is ensured that the new system satisfies the quality standards adopted by the business and the system satisfies the users. Thus the final acceptance testing has two major parts:

i. **Quality Assurance Testing:** It ensures that the new system satisfies the prescribed quality standards and the development process is as per the organization’s quality assurance methodology.

ii. **User Acceptance Testing:** It ensures that the functional aspects expected by the users have been well addressed in the new system. There are two types of the user acceptance testing:

- **Alpha Testing**: This is the first stage, often performed by the users within the organization.
- **Beta Testing**: This is the second stage, generally performed by the external users. This is the last stage of testing, and normally involves sending the product outside the development environment for real world exposure.

Pilot Testing

This kind of testing involves the users just before actual release to ensure that users become familiar with the release contents and ultimately accept it. This is often suggested before "Go Live" in a typical ERP system going to production. It may involve many users and is generally conducted over a short period of time and is tightly controlled.

Sociability Testing

In this type of testing, an application is tested in its normal environment, along with other standard applications, to make sure they all get along together; that is, they do not corrupt each other's files, they don't crash, they don't consume system resources, they don't lock up the system, they can share the printer and other shared resources. This is especially useful for web-based applications.

Auditor's role in Testing

Auditor can play a very crucial role in the testing phase of SDLC. Typically, technical people fail to test boundary or exception conditions in their testing activities. Auditor must carefully review various test plans and test cases in addition with corresponding test data and results. Programmers tend to prepare irrelevant fictitious data which does not reflect real life data values. e.g. Names can be put as XYZ.

Throughout the testing phase, auditors need to identify whether written test plans and scripts are available and adequate. E.g. test plans should include the test's set-up and scope, testing procedures and data, expected results, and sign-off from the appropriate staff. Also it is necessary to check whether testing is performed in an environment that is separate from the production environment. Logging of test results should be ensured for post testing analysis or review. If test results have unexpected results, it is necessary to check and review adequacy and appropriateness of reasons. Auditors may have to ensure that nothing is installed in the production

environment until it is successfully examined in a test environment and formally approved by the business user.

Auditor can conduct audit of testing phase to verify compliance of the testing carried out by various teams. The prime reason may be to judge if the process complies with a standard. An auditor may compare the actual testing process with the documented process; e.g. ISO Standards require us to define our Software testing process. The audit will try to verify if we actually conducted the testing as documented.
Sometimes audit objective could be to improve process or to address some problems in the testing phase. Since auditor can do a value addition as a third-party independent reviewer, the business may wish to seek auditor's opinion on process improvement aspects. Auditor's involvement may also be necessary to find out a root cause to a problem. Auditing Test Process helps the management to understand if the process is being followed as specified. Typically Testing audit may be done for one or more of the following factors:

- To ensure continued reliability and integrity of the process
- To verify compliance of standards (ISO, CMM, etc)
- To solve process related problems
- To find the root cause of a specific problem
- To improve the Testing process

Auditor may have to conduct testing by preparing test strategies, designing test plans and preparing test cases. Auditor will need a testing environment which is similar to the production environment. Auditor will carry out testing by running various software processes module by module and reviewing the results.

After the testing of software is over, SDLC project goes in the next phase viz. implementation of the software.

Phase VI- System Implementation

Planning of the implementation should be commenced much before actual date of the implementation and the implementation plan as developed in the Design Phase should be used with the modifications if required. While implementing the new system, if test data was used for carrying out testing, care should be taken to delete the same before making the new system live.

Accreditation of system implementation should be sought from competent authorities. In order to enable them to accredit the system (or parts thereof), the team may have to print control reports. E.g. if data is migrated from earlier system, count of records, totals of numeric fields matching with earlier system's numerical totals etc may be printed.

When migration is from earlier computerized system, it is necessary to provide for fall back arrangement if the working on new system fails due to some reason. In order to achieve this fall back arrangement, earlier system should be kept ready with experienced users on earlier system ready to take over at short notice. Backup should start from day one on the new system with roll back features in place. E.g. "before image" copy and "after image" of data before and after processing should be made and used if roll back is to be done.

There are four types of implementation strategies:

- **Direct implementation / Abrupt change-over:** In this strategy, the old system is suspended on a specific day and the new system is tried out. This is not recommended, because in case the new system crashes, the old system is also not available. In case of small applications, or when migrating from a manual to computer system, this may be used.
- **Parallel implementation:** Both the old and new systems are run in parallel to verify if their output is the same. Then the old system is suspended.
- **Phased implementation:** This strategy consists of implementing the new system in parts. This makes implementation more manageable.
- **Pilot implementation:** The new systems is first implemented in a small, non-critical unit and then moved to larger unit.

Except direct implementation, others strategies are not mutually exclusive. A judicious combination of the strategies can be adopted, depending on the type of application.

Activities during Implementation Stage

Some of the main activities during implementation are:

- **Installation of new hardware / software:** If the new system interfaces with the other systems or is distributed across multiple software platforms, some final commissioning tests of the production environment may be desirable to prove end to end connectivity.
- **Data conversion:** Following steps are necessary for this activity:
    - Determining what data can be converted through software and what data manually.
    - Performing data cleansing before data conversion
    - Identifying the methods to access the accuracy of conversion like record counts and control totals
    - Designing exception reports showing the data which could not be converted through software.

  - Establishing responsibility for verifying and signing off and accepting overall conversion by the system owner
  - Actual conversion
- **User Final Acceptance testing:** Ideally, the user acceptance test should be performed in a secured testing environment where both source and executable codes are protected. This helps to ensure that unauthorized or last minute change to the system does not take place without going through the standard system maintenance process.
- **User training:** The following types of training may be given to the users:
  - Manager's training on overview and MIS
  - Operational user training on how to use the software, enter the data, generate the output
  - EDP training on the technical aspects.

The next phase is Post Implementation Review and Maintenance.

Phase VII- Post-Implementation Review and Maintenance

After the system stabilizes, a check should be done to ensure that the system has fulfilled the objectives. Otherwise, move back to the appropriate stage of the development cycle. The post implementation review should be performed jointly by the project development team and the appropriate end users or alternatively, an independent group not associated with the development process, either internal or external, should carry out the audit, to meet the following objectives:

1. Assess the adequacy of the system:
   a. Whether the system met management's objectives and user requirements
   b. Whether the access controls have been adequately implemented and actually working
2. Evaluation and comparison of the actual Cost Benefit or ROI as against the same projected in the feasibility study phase.
3. Recommend on the system's inadequacies and deficiencies
4. Develop a plan for implementing the accepted recommendations
5. Evaluate the system development project process

Maintenance is also part of the post implementation review. The definition of software Maintenance by IEEE is as follows:

The modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment.

Maintenance can only happen efficiently if the earlier phases are done properly. There are four major problems that can slow down the maintenance process:

unstructured code, maintenance programmers having insufficient knowledge of the system, documentation being absent, out of date, or at best insufficient, and software maintenance having a bad image. The success of the maintenance phase relies on these problems being fixed earlier in the life cycle.

Maintenance consists of four parts. **Corrective maintenance** deals with fixing bugs in the code. **Adaptive maintenance** deals with adapting the software to new environments. **Perfective maintenance** deals with updating the software according to changes in user requirements. Finally, **preventive maintenance** deals with updating documentation and making the software more maintainable. All changes to the system can be characterized by these four types of maintenance. Corrective maintenance is 'traditional maintenance' while the other types are considered as 'software evolution.'

As products age it becomes more difficult to keep them updated with new user requirements. Maintenance costs developers time, effort, and money. This requires that the maintenance phase be as efficient as possible. There are several steps in the software maintenance phase. The first is to try to understand the design that already exists. The next step of maintenance is reverse engineering in which the design of the product is reexamined and restructured. The final step is to test and debug the product to make the new changes work properly.

Types of Software Maintenance

There are four types of maintenance according to Lientz and Swanson, which are given as follows:

- **Corrective Maintenance:** Corrective maintenance deals with the repair of faults or defects found. A defect can result from design errors, logic errors and coding errors. Design errors occur when, for example, changes made to the software are incorrect, incomplete, wrongly communicated or the change request is misunderstood. Logic errors result from invalid tests and conclusions, incorrect implementation of design specifications, faulty logic flow or incomplete test of data. Coding errors are caused by incorrect implementation of detailed logic design and incorrect use of the source code logic. Defects are also caused by data processing errors and system performance errors. All these errors, sometimes called 'residual errors' or 'bugs', prevent the software from conforming to its agreed specification. The need for corrective maintenance is usually initiated by bug reports drawn up by the end users. Examples of corrective maintenance include correcting a failure to test for all possible conditions or a failure to process the last record in a file.

- **Adaptive Maintenance:** Adaptive maintenance consists of adapting software to changes in the environment, such as the hardware or the operating system. The term environment in this context refers to the totality of all conditions and influences which act from outside upon the system, for example, business rule, government policies, work patterns, software and hardware operating platforms. The need for adaptive maintenance can only be recognized by monitoring the environment. An example of a government policy that can have an effect on a software system is the proposal to have a 'single European currency', the ECU. An acceptance of this change will require that banks in the various member states, for example, make significant changes to their software systems to accommodate this currency. Other examples are an implementation of a database management system for an existing application system and an adjustment of two programs to make hem use the same record structures. A case study on the adaptive maintenance of an Internet application 'B4Ucall' is another example. B4Ucall is an Internet application that helps compare mobile phone packages offered by different service providers. In their study on B4Ucall, Bergin and Keating discuss that adding or removing a complete new service provider to the Internet application requires adaptive maintenance on the system.
- **Perfective Maintenance:** Perfective maintenance mainly deals with accommodating to new or changed user requirements. Perfective maintenance concerns functional enhancements to the system and activities to increase the system's performance or to enhance its user interface. A successful piece of software tends to be subjected to a succession of changes, resulting in an increase in the number of requirements. This is based on the premise that as the software becomes useful, the users tend to experiment with new cases beyond the scope for which it was initially developed. Examples of perfective maintenance include modifying the payroll program to incorporate a new union settlement, adding a new report in the sales analysis system, improving a terminal dialogue to make it more user-friendly, and adding an online HELP command.
- **Preventive Maintenance:** Preventive maintenance concerns activities aimed at increasing the system's maintainability, such as updating documentation, adding comments, and improving the modular structure of the system. The long-term effect of corrective, adaptive and perfective changes increases the system's complexity. As a large program is continuously changed, its complexity, which reflects deteriorating structure, increases unless work is done to maintain or reduce it. This work is known as preventive change. The change is usually initiated from within the maintenance organization with the intention of making programs easier to understand and hence facilitating future maintenance work.

Examples of preventive change include restructuring and optimizing code and updating documentation.

Umbrella Activities

In addition to the activities associated with each phase, there are may others which are undertaken throughout the life cycle; these are referred to as umbrella activities. Some of these activities are:

- Project tracking and control
  - Budgeting
  - Reporting
  - Review
  - Signoffs
- Quality assurance
- Configuration management
- Documentation
- Change management
- Reusability management
- Measurement
- Risk management

Auditors' Role in SDLC

The audit of systems under development can have three main objectives: first, to provide an opinion on the efficiency, effectiveness, and economy of project management; second, to assess the extent to which the system being developed provides for adequate audit trails and controls to ensure the integrity of data processed and stored; and third to assess the controls being provided for the management of the system's operation.

For the first objective to achieve, an auditor will have to attend project and steering committee meetings and examine project control documentation and conducting interviews. This is to ensure what project control standards are to be complied with, (such as a formal systems development process) and determining the extent to which compliance is being achieved. For addressing second objective, the auditor is can examine system documentation, such as functional specifications, to arrive at an opinion on controls. The auditor's opinion will be based on the degree to which the system satisfies the general control objectives that any Information Technology system should meet. A list of such objectives should be provided to the auditee. The same is true for the third objective, viz. system's operational controls. The auditor should provide the a list of the standard controls, over such operational concerns as

response time, CPU usage, and random access space availability, that the auditor has used as assessment criteria.

An Auditor may adopt a rating system such as on scale of 1 to 10 in order to give rating to the various phases of SDLC. E.g. in rating a Feasibility Study, auditor can review Feasibility Study Report and different work products of this study phase. An interview with personnel who have conducted this feasibility study can be conducted. Depending on the content and quality of the Feasibility Study report and interviews, an auditor can arrive at a rating between 1 to 10 (10 being best). After deriving such a rating for all the phases, the auditor can form his/her overall opinion about the SDLC phases.

In order to audit technical work products (such as database design or physical design), auditor may opt to include a technical expert to seek his/her opinion on the technical aspects of SDLC. However, auditor will have to give control objectives, directives and in general validate the opinion expressed by technical experts. Some of the control considerations for an auditor are:

1. Documented policy and procedures
2. Established Project team with all infrastructure and facilities
3. Developers/ IT managers are trained on the procedures
4. Appropriate approvals are being taken at identified mile-stones
5. Development is carried over as per standards, functional specifications
6. Separate test environment for development/ test/ production / test plans
7. Design norms and naming conventions are as per standards and are adhered to
8. Business owners testing and approval before system going live
9. Version control on programs
10. Source Code is properly secured
11. Adequate audit trails are provided in system
12. Appropriateness of methodologies selected.

Further, Post-Implementation Review is performed to determine whether the system adequately meets earlier identified business requirements and needs (in feasibility studies or Requirements Specifications). Auditors should be able to determine if the expected benefits of the new system were realized and whether users are satisfied with the new system. In post implementation review, auditors need to review which of the SDLC phases have not met desired objectives and whether any corrective actions were taken. If there are differences between expectations and actual results, auditors need to determine the reasons for the same. E.g. it could be due to incomplete user requirements. Such reasons can help auditors to evaluate the current situation and offer guidelines for future projects.

Master Checklist

The process objectives are:

- To ensure an appropriate acquisition and / or development of information systems including software,
- To maintain the information systems in an appropriate manner.

The following checklist may be used by the IS Auditors for this purpose:

S. No.	Checkpoints	Status
1.	Whether information system acquisition and / or development policy and procedure documented?	
2.	Whether system acquisition and / or development policy and procedure approved by the management?	
3.	Whether the policy and procedure cover the following: • Problems faced in the existing system and need for replacement • Functionality of new IS • Security needs • Regulatory compliance • Acceptance Criteria • Proposed roles and responsibilities • Transition/ Migration to new IS • Interfaces with legacy systems • Post implementation review • Maintenance arrangements.	
4.	Whether policy and procedure documents are communicated / available to the respective users?	
5.	Whether policy and procedure documents are reviewed and updated at regular intervals?	
6.	Whether the organization has evaluated requirement and functionalities of proposed IS? *(Verify the requirement analysis conducted at three levels viz. process level, application level and organization level. Verify the site visit reports and other customer references obtained with respect to functionalities of proposed IS).*	

7.	Whether the organization carried out feasibility study in respect of the following • Financial feasibility • Operational feasibility • Technical feasibility	
8.	Whether the selection of vendor and acquisition terms considers the following: • Evaluation of alternative vendors • Specification on service levels and deliverables • Penalty for delays • Escrow mechanism for Source codes • Customization • Upgrades • Regulatory Compliance • Support and maintenance.	
9.	Whether the organization has identified and assigned roles in development activities to appropriate stakeholders? *(Verify the assigned roles should be on "need to know" and "need to basis". and duties of developers and operators are segregated).*	
10.	Whether the organization has a separate development, test and production environments?	
11.	Whether the IS developed plan is prepared and approved by the management? *(Verify that IS development plan to include:* • *Input data elements,* • *Validations controls viz. Field/ Transactions/ File with appropriate error reporting* • *Process workflow* • *data classifications with security are in place, viz. Read only for users, Read/ Write for authorized persons* • *Output).*	
12.	Whether the testing of IS includes: • Confirms the compliance to functional requirements	

	• Confirms the compatibility with IS infrastructure • Identifies bugs and errors and addresses them by analyzing root causes Escalating functionality issues at appropriate levels.	
13.	Whether the adequate documentation for: • Preserving test results for future reference • Preparation of manuals like systems manual, installation manual, user manual • Obtaining user sign off / acceptance	
14.	Whether the implementation covers the following? • User Departments' involvement and their role • User Training • Acceptance Testing • Role of Vendor and period of Support • Required IS Infrastructure plan • Risk involved and actions required to mitigate the risks • Migration plan	
15.	If the development activities are outsourced, are the outsourcing activities evaluated based on the following practices: • What is the objective behind Outsourcing? • What are the in-house capabilities in performing the job? • What is the economic viability? • What are the in-house infrastructure deficiencies and the time factor involved? • What are the Risks and security concerns? • What are the outsourcing arrangement and fall back method? • What are arrangements for obtaining the source code for the software? • Reviewing the capability and quality of software development activities by visit to vendor's premises? • Review of progress of IS development at periodic intervals.	

16.	Whether the organization carried out a post implementation review of new IS?	
17.	Whether a process exists for measuring vendors' performance against the agreed service levels?	
18.	Whether the post implementation review results are documented?	

Summary

In this chapter, we have learned the following major aspects of SDLC:

- The design phase is undertaken only if software is being developed. It details how the proposed system will translate into a working model.
- The steps involved in this phase are: Architectural design; Design of data / Information Flow; Design of database; Design of user interface; Physical design; and Selection of appropriate hardware and software.
- The next phase is programming or construction. As indicated earlier, for object oriented programming sometimes it is referred as Construction. This phase converts the design into a working program, known as Source Code or Source Program. Source code when compiled generates Executable Program which is run by user and loads inside computer's memory.
- Programmers have to ensure under this phase that the programs being developed are accurate, reliable, robust, efficient, readable, user friendly, and maintainable. The coding follows debugging exercise.
- The next phase is Software Testing. This is a process of testing software in a controlled manner to ensure it meets the specifications.
- Software testing comprises of
    - Unit testing
    - Integration / Interface testing
    - System testing
    - Final acceptance testing
- After the information system is through with the testing process, it moves on to the implementation phase. This involves installation of the new hardware and software, data conversion, user acceptance of the new system, user training and test runs.
- The final stage of SDLC is the post implementation review phase, which involves assessment, evaluation, deficiencies, recommendations and corrective, adaptive, perfective and preventive maintenance.

- In addition to the activities associated with each phase, there are others, which are undertaken throughout the life cycle, which are referred to as umbrella activities. Some of them are:
    - Project tracking and control
    - Quality assurance
    - Configuration management
    - Documentation
    - Change management
    - Reusability management Measurement
    - Risk management

Sources:

- Ron Weber: Information Systems Control and Audit, Pearson Education, India, Third Impression, 2009.
- Valacich George Hoffer: Essentials of Systems Analysis & Design, PHI Pvt. Ltd., N. Delhi, India, 2004.
- Muneesh Kumar: Business Information Systems, Vikas Publishing House Pvt. Ltd., N. Delhi, India, 2001.
- Charles Parker, Thomas Case: Management Information Systems, Mitchell McGraw Hill, India, 1993.
- M. M. Pant: System Analysis, Data Processing and Quantitative Techniques, Pitambar Publishing Co. Pvt. Ltd., N. Delhi, India, 1999.
- Gordon B. Davis, Margrethe H. Olson, Management Information Systems, McGraw-Hill International Editions, 1984.
- Sujata Garg: Professional Approach to Management Information & Control Systems, Bharat Law House Pvt. Ltd., N. Delhi, India, 2005.
- Pankaj Jalote: An Integrated Approach to Software Engineering, Narosa Publishing House, N. Delhi, India, Third Edition, 2005.
- Roger S. Pressman: Software Engineering- A Practitioner's Approach, McGraw-Hill, Sixth Edition, 2005.

Multiple Choice Questions

1. Debugging is:
    a. creating program code
    b. finding and correcting errors in the program code
    c. identifying the task to be computerized
    d. creating the algorithm

2. The problem statement includes the _____, which lists specific input numbers, a program would typically expect the user to enter and precise output values that a perfect program would return for those input values.
   a. testing plan
   b. error handler
   c. IPO cycle
   d. input-output specification

3. The design of a database consists of following major activities:
   a. Conceptual Modeling
   b. Data Modeling
   c. Storage Structure Design
   d. All of the above

4. …… is a technique for designing relational database tables to minimize duplication of information and to eliminate data anomalies.
   a. Database Normalization
   b. Data Modeling
   c. Storage Structure Design
   d. None of these

5. Physical design includes the following step/s:
   a. Designing physical files and databases
   b. Designing system and program structure
   c. Designing distributed processing strategies
   d. All of the above

6. …….. stores data or fixed data about an entity.
   a. Master file
   b. Parameter file
   c. Transaction file
   d. None of these

7. The data in ……. Files changes less frequently or may not change at all during the life span of the entity.
   a. Master file
   b. Parameter file
   c. Transaction file
   d. None of these

8. C++ is a/an:
   a. Procedural Language
   b. Object Oriented Language
   c. Assembly Language
   d. Machine Language

9. …….. is a technique for designing relational database tables to minimize duplication of information and to eliminate data anomalies.
   a. Database Normalization
   b. Data Fragmentation
   c. Data Modeling
   d. None of these

10. ……. contains data captured about real life events happening in day-to-day life.
    a. Master file
    b. Parameter file
    c. Transaction file
    d. None of these

11. Sequential file organization is also known as:
    a. Text File Organization
    b. Flat File Organization
    c. Indexed File Organization
    d. Both a. and b.

12. In ……., address of each data record is determined based on a hashing algorithm which converts a primary key value to a record address.
    a. Text File Organization
    b. Flat File Organization
    c. Indexed File Organization
    d. None of these

13. A good program should have the following characteristic/s:
    a. Accuracy
    b. Reliability
    c. Robustness
    d. All of the above

14. …….. is the process of testing individual units (i.e. individual programs or functions or objects) of software in isolation.
    a. Unit Testing
    b. System Testing
    c. Penetration Testing
    d. All of the above

15. In …….., testing is done by using the same test data in the new and old system, and the output results are compared.
    a. Unit Testing
    b. Parallel Testing
    c. Penetration Testing
    d. All of the above

16. In Black Box Testing, the focus is on …….
    a. Functional Correctness
    b. Structural Correctness
    c. Both a. and b.
    d. None of these

17. In White Box Testing, the focus is on …….
    a. Functional Correctness
    b. Structural Correctness
    c. Both a) and b.
    d. None of these

18. ……. is conducted when the system is just ready for implementation.
    a. Unit Testing
    b. Parallel Testing
    c. Penetration Testing
    d. Final Acceptance Testing

19. ……. is the first stage, often performed by the users within the organization.
    a. Alpha Testing
    b. Beta Testing
    c. Both a. and b.
    d. None of these

20. In ..... the new system is first implemented in a small, non-critical unit and then moved to a larger unit.
    a. Pilot Implementation
    b. Phased Implementation
    c. Parallel Implementation
    d. None of these

Answers:

1. b.	2. a.	3. d.	4. a.	5. d.	6. a.
7. a.	8. b.	9. a.	10. c.	11. d.	12. d.
13. d.	14. a.	15. b.	16. a.	17. b.	18. d.
19. a.	20. a.				

3 Alternative Methodologies of Software Development

Learning Objectives

To provide an understanding of:

- Different approaches to system development: advantages, problems, and selection criteria.
- Different aspects involved in the maintenance of information systems.

In system development, an analyst looks for the processes that are reusable and predictable. The main aim is to improve productivity and quality of the system and also focusing on delivering the system on time by adhering to a systematic schedule and keeping a check on the budget. An analyst of the system can achieve it by formalizing the process and applying project management techniques. These techniques help meet the desired expectations in terms of functionality, cost, delivery schedule, productivity and quality of the system. A life cycle called SDLC (System Development Life Cycle) is followed to achieve a system with the above mentioned features. SDLC is implemented in the form of various methodologies and their respective models.

Methodology: Methodology is a comprehensive guideline that is followed to successfully complete every SDLC phase. It is a collection of models, tools and techniques. Methodologies are used to enhance performance in the system. They are comprehensive and multi-step approaches to systems development.

Model: Model describes the system in both formal and informal ways. For example, one model can show the various components of the system and how they are related to each other. Another model can show the order of information flow among the components of the system. Some of the models used in system development are:

- Flowchart
- DFD (Data Flow Diagram)
- ERD (Entity Relationship Diagram)
- Structure Chart
- Use Case Diagram

- Class Diagram
- Seq. Diagram

Moreover, some models that manage the development process are -

- PERT charts
- Gantt chart
- Hierarchy chart
- Financial analysis models like NPV, ROI, etc.

Tools: A tool is a software support that helps create models or other components required in the project. Some examples of the tools used in system development are smart editors, C-S help, debugging tools, CASE (Computer-Aided System Engineering), etc. Tools help the analyst to create the important system models.

Techniques: A technique is a collection of guidelines that help an analyst complete a system development activity. Some of the techniques used in it are OO (Object Oriented) analysis, data modeling, relational database design, structured analysis and design, and software-testing.

Researchers all over the world have done a lot of experimentation and have adopted various methodologies and models for conducting the SDLC phases. Software Development Methodology is a framework which is adapted to structure, plan, and control the process of developing an information system. The framework of a software development methodology consists of the following:

- A software development philosophy, with the approach or approaches of the software development process.
- Multiple tools, models and methods, to assist in the software development process.

Over the past few years a wide variety of frameworks have evolved. Each framework has its own accepted strengths and weaknesses. Choice of system development methodology is dependent on the nature of the project under development. The system development methodology is based on various technical, organizational, project and team considerations. The methodology is often documented formally.There are nearly 40 different models. This chapter focuses on some of the models that are generally used as a tool of system development methodology for system development.

The present trend of using OOP and web-based systems demands that instead of using traditional methods, it is better to adopt alternative development methodologies. This chapter focuses on the models available for software development. But in order

to understand and appreciate these alternative models, a brief discussion of SDLC is necessary.

SDLC (System Development Life Cycle)

SDLC, also known as Software Development Life Cycle, is a logical process used by a systems analyst to develop an information system, including requirements, validation, training, and user ownership. It results in a high quality system that meets customer expectations, reaches completion within time and cost estimates, works effectively and efficiently in the current and planned Information Technology infrastructure, and is inexpensive to maintain and cost-effective to enhance. SDLC consists of the following phases: -

1. Feasibility Study
2. Requirements Analysis
3. Systems Design
4. Programming
5. Testing
6. Implementation
7. Post-Implementation Support

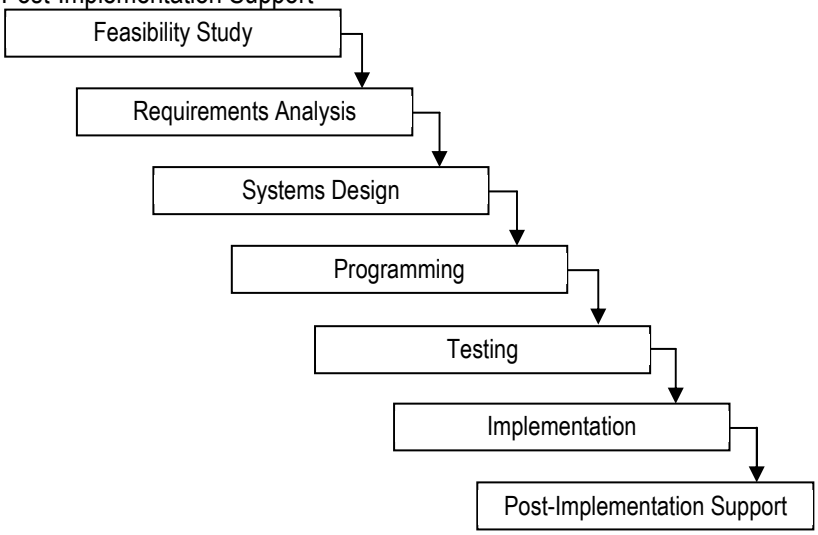


Fig 3.1 System Development Life Cycle

[ASG1]Waterfall Model: Waterfall model is a traditional SDLC model, which is also called a 'Classical Life Cycle'. It is the oldest paradigm in system development. This model assumes a sequential and systematic approach, in which all the SDLC phases run serially (sequentially) from top to bottom. In other words, it is a model with a

sequential software development process, in which progress is seen as flowing steadily downwards like a waterfall as shown in fig 3.2.

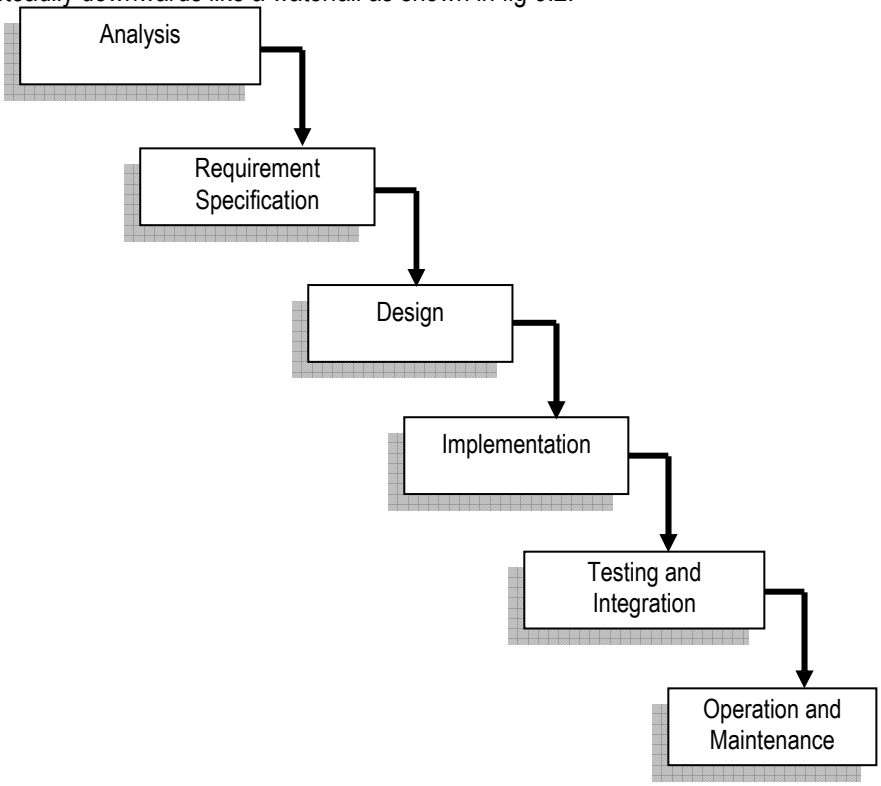


Fig 3.2 Waterfall Model

For example, when the requirement phase is fully completed and perfected, the design phase is taken care of. Thus movement to the next phase is possible only when the preceding phase is completed and perfected.

This model was most suited for traditional programming languages like COBOL (Common Business Oriented Language). It is still in use especially in those systems which are stabilized over a period of time, such as payroll or financial accounting system.

On of the limitation of waterfall model is that, phases do not run in a parallel manner. Secondly, although, the model gathers all the possible requirements during the requirement gathering and analysis phase, still, requirements from clients go on getting added to the list even after the end of "Requirement Gathering and Analysis" phase. This affects the system development process and its success negatively. Thirdly, the problems with a particular phase do not get solved completely during the

time it is on. In fact, many problems arise only after the phase is signed off. The result is that the system thus designed and created has a poor structure. Moreover, the project is not partitioned in phases in a flexible manner. As the requirements of the client go on getting added to the list, not all the requirements are fulfilled completely. The result is that the system becomes almost unusable. These requirements are then met only in a newer version of the system, which increases the cost of system development.

Prototyping: Prototyping aims at putting together quickly a working model (a prototype) to test the various aspects of a design.

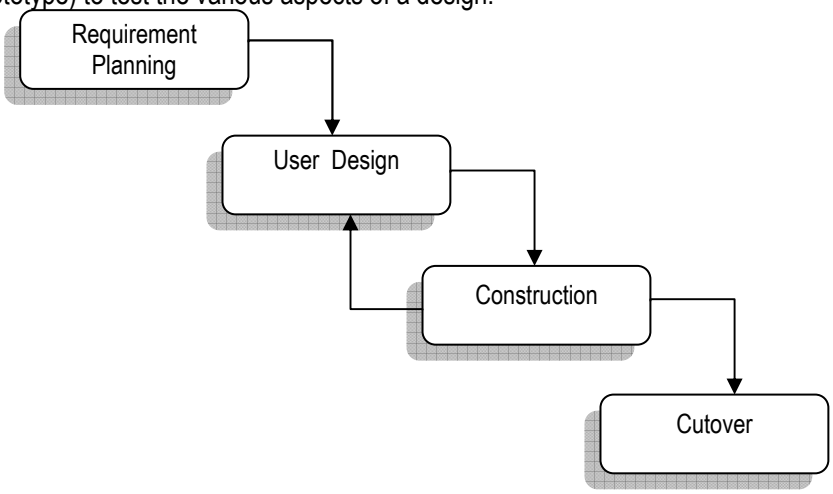


Fig 3.3 Simple Prototype Model

It also illustrates ideas or features and gathers early user feedback. One or many prototypes are made in a process of iterative and incremental development where each prototype is influenced by the performance of previous designs. When the prototype is sufficiently refined and meets the requirements of functionality and other design goals, the product is ready for production.

Prototyping model is implemented when one of the following conditions is present:

- When a customer defines a general objective for software, but does not identify detailed input processing or output requirements.
- When the developer is ensure of one or all of the following:
  - The efficiency of an algorithm
  - The adaptability of an operating system
  - The form that human-machine interaction should take

The prototyping software development process begins with requirements collection, followed by prototyping and user evaluation. Often the end users may not be able to provide a complete set of application objectives, detailed input, processing, or output requirements in the initial stage. After the user evaluation, another prototype will be built based on feedback from users, and again the cycle will return to customer evaluation.

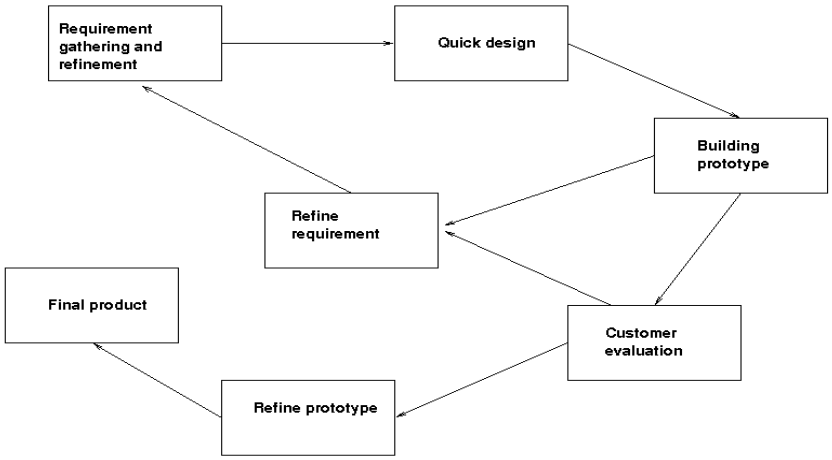


Fig 3.4 Detailed Working of a Prototype Model

The following are the steps in the prototyping approach:

- Requirements gathering: The developer gets initial requirements from the users.
- Quick design: The emphasis is on visible aspects such as input screens and output reports.
- Construction of the prototype: by the developer on the basis of inputs from the users.
- Users' evaluation of the prototype: The users accept the screens and options as shown to them.
- Refinement of the prototype: It is refined by fine tuning the user's requirements.
- The last two steps are iterated till the user is fully satisfied with the prototype.

The drawbacks of the prototyping approach are:

- The user sees the 'working' version of the software, without realizing that the processing logic is still not ready. So, the user starts making unreasonable delivery date demands without realizing that prototype has to be expanded to handle transaction volume, client server network connectivity, backup and recovery procedures and control features.
- As the development has to be carried out very fast, the developer uses the 4GL (4 Generation Language) tools. But in larger developments, design strategy has

to be clearly laid down; otherwise the effort will result in poor quality, poor maintainability and low user acceptance, resulting in the failure of the effort.

- The prototype is only for eliciting user requirements. Even the input and output programs may have to be rewritten taking into account the target environment and efficiency considerations. Having worked hard at developing a prototype, the developer tries to work around it to develop the application thereby leading to sub optimality.
- The capability of the prototype to accommodate changes could also lead to problems. The user may at times add changes at the cost of strategic objectives of the application.
- In prototyping, the software that is being developed is modified and coded, as and when the user feels the need for change. It becomes difficult to keep track of such changes in the controls.
- Changes in design and development keep happening so quickly that formal change control procedures may be violated.

Though the IS auditor is aware about the risks associated with prototyping, the IS auditor also knows that this method of system development can provide the organization with substantial saving in time and cost. Similarly, since users are giving approval to data entry screens and report layouts early in SDLC life cycle, chances of meeting user requirements are very high in this model.

RAD: RAD, also known as Rapid Application Development, is an incremental model which has a short development cycle in which the requirements have to be clearly understood and the scope well defined. It is a high speed adaptation of the waterfall model, in which rapid development is achieved by using a component based construction approach.

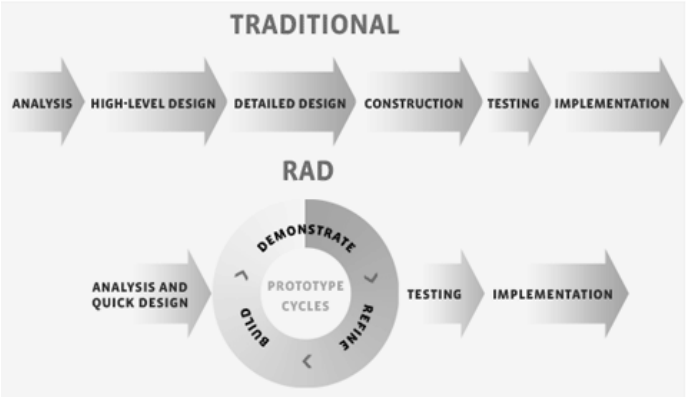


Fig 3.5 Comparison of Traditional Model with RAD (Rapid Application Development) Model

RAD leverages the following techniques to keep the development cycle short:

- Multiple small teams
- Modular applications
- Evolutionary prototype
- Automated tools
- Design workshops
- Component- based development
- Fourth generation languages
- Rigid time frames

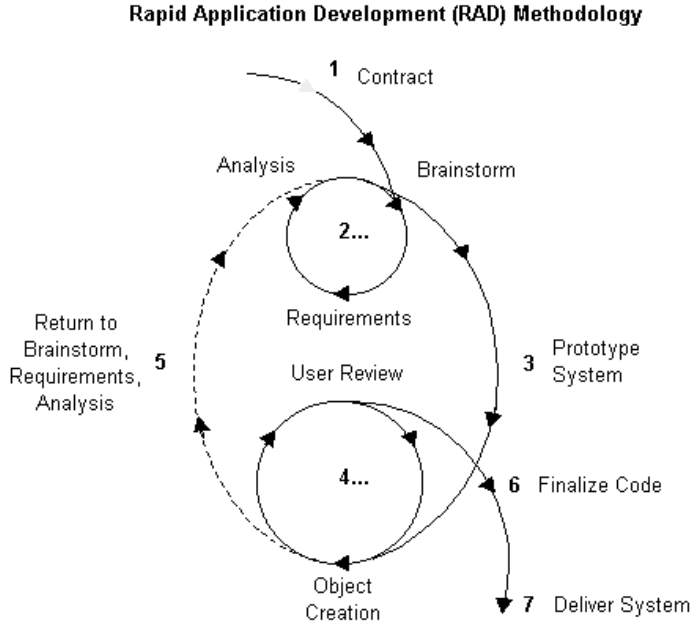


Fig 3.6 RAD Methodology

This approach is adopted only for individual strategically important systems and not for ERP (Enterprise Resource Planning) kind of systems. It is undertaken only if the following four pillars of an organization are strong.

- Management – capable of quick decisions on development and user teams
- People – in user team and development team
- Methodology – to use proven methodology and not the recently invented one
- Tools – proven integrated tools, such as VB / Delphi c should be used.

The four stages in this approach are:

- Definition of scope
- Creation of a functional design
- Construction of application
- Deployment

Methodology: The RAD methodology includes the following:

- Combining the best available technique and specifying the sequence of tasks that will make these techniques most effective.
- Using evolutionary prototypes that are eventually transformed into the final product.
- Using workshops, instead of interviews, to gather requirement for review design.
- Selecting a set of case tools to support modeling, prototyping and code reusability, as well as automating many combination techniques.
- Implementing time boxed development that allows the development team to quickly build the core of the system and implement refinements in subsequent releases.
- Providing guidelines for success and describing pitfalls to be avoided.

The structure of the RAD life cycle is thus designed to ensure that developers build systems that users really need. This life cycle, which goes through four stages, includes all the activities and tasks required to define business requirements and design, develop, and implement the application system that supports these requirements.

Requirements Planning: It is also known as the concept definition stage. It defines the business functions and data subject areas that it will support and thus determine also its scope.

User Design: It is also known as the functional design stage. It uses workshops to model the system's data and processes o build a working prototype of critical system components.

Construction: It is also known as the development stage. It completes the construction of the physical application system, builds the conversion system, and develops user aids and implementation work plans.

Implementation: It is also known as the deployment stage. It includes final user testing and training, data conversion, and the implementation of the application system.

People: The success of RAD is contingent upon the involvement of people with right skills and talent. Excellent tools are essential to fast application development, but they do not, by themselves, guarantee success. Fast development relies equally heavily on the people involved. They must therefore be carefully selected, highly trained, and highly motivated. They must also be able to use the tools and work together in close-knit teams. Rapid development usually allows each person involved to play several different roles, so a RAD project mandates a great degree of cooperative effort among a relatively small group of people.

The key players in a RAD project include the following:

Key Player	Characterized By
Sponsor	A high level user executive who funds the system and is dedicated to both the value of the new system and to achieving results quickly.
User Coordinator	A user appointed by the Sponsor to oversee the project from the user's perspective.
Requirements Planning Team	A team of high-level users who participate in the Joint Requirements Planning workshop.
User Design Team	A team of users who participate in the design workshop. It should comprise both high-level users from the Planning Team and lower-level users with a more detailed knowledge of the system.
User Review Board	A team of users who review the system after construction and decide whether modifications are needed before cutover.
Training Manager	The person responsible for training users to work with the new system.
Project Manager	A person who oversees the development effort.
Construction (SWAT) Team	The SWAT (Skilled Workers with Advanced Tools) team is a small team of two to six developers who are highly trained to work together at high speed. To achieve the fastest possible development, the team members must be highly skilled in the RAD methodology and in using the chosen CASE (Computer-Aided Systems Engineering) toolset.
Workshop Leader	The specialist who organizes and conducts workshops for Joint Requirements Planning and Joint Application Design.

Management: It is well known that achieving high-speed development is a very complex process. Management must be totally committed to RAD to manage the change in work culture. They must be prepared to motivate both users and IT staff, selects and manage SWAT (Skilled Workers with Advanced Tools) teams, and demonstrate through the use of performance measurements that RAD does mean speed, quality, and productivity. Good management and dedication to the ideals of RAD are thus essential to faster system building. To successfully introduce rapid development, management must pay careful attention to human motivation. Managers should target 'Early Adapters,' professionals who see the value of a new methodology and lead the way in using it. Similarly, managers must use effective motivational strategies for each individual employee, in the form of money, pride, prestige, excitement, or some combination thereof.

Tools: RAD methodology uses both computerized tools and human techniques to achieve the goals of high-speed and quality. Success of such projects is also dependent upon the tools used, which are mainly CASE (Computer-Aided Systems Engineering) tools.

RAD uses hybrid teams, usually of about 6 people, including both developers and full-time users of the system plus anyone else who has a stake in the requirements. Developers chosen for RAD teams should be multi-talented "renaissance" people who can work as analysts, designers and programmers.

The advantages of RAD are:

- Sometimes it is a better idea to buy software because buying may save money compared to building the system. Is it an advantage?
- In RAD, deliverables, based on condition, are easier to port because they make greater use of high-level abstractions, scripts, intermediate code.
- The development in RAD is conducted at a higher level of abstraction because RAD tools operate at that level.
- Early visibility is present because of prototyping.
- Greater flexibility is provided because developers can redesign almost at will.
- It greatly reduces the manual coding because of wizards, code generators, and code reuse.
- There is an increased user involvement because they are represented on the team at all times.
- Possibly fewer defects are noticed because CASE tools may generate much of the code.
- Possibly reduced cost is there because time is money, also because of reuse.
- A shorter development cycle is present because development tilts toward schedule and away from economy and quality.

- Standardized look and feel is a plus point because APIs (Application Programming Interface) and other reusable components give a consistent appearance.

The drawbacks of RAD are as follows:

- For large projects, it requires a lot of human resources creating management problems.
- It is a joint effort between the client and developer. If the commitment weakens on either side RAD approach will not work out properly.
- It is not suitable for monolithic (very big) applications.
- For mission critical applications, where quality and reliability assume higher importance than the time of development, this approach is not feasible.

RAD model can work in the following situations:

- The application will be run as a standalone system.
- Major use can be made of preexisting class libraries (APIs [Application Programming Interface]).
- Performance is not critical
- Product distribution will be narrow (in-house or vertical market).
- Project scope (macro-schedule) is constrained.
- Reliability is not critical.
- System can be split into several independent modules.
- The product is aimed at a highly specialized IS (information Systems) market.
- The project has strong micro-schedule constraints (time boxes).
- The required technology is more than a year old.

RAD model is not a good choice for system development when one or more of the following situation(s) are present:-

- Application must interoperate with existing programs.
- Few plug-in components are available.
- Optimal performance is required.
- Product development can't take advantage of high-end IS tools (e.g., 4GLs).
- Product distribution will be wide (horizontal or mass market).
- RAD becomes QADAD (Quick And Dirty Application Development).
- RAD methods are used to build operating systems and computer games.
- Technical risks are high due to the use of "bleeding" edge technology.
- The product is mission-critical or life-critical.
- The system cannot be modularized.

Spiral Model: Spiral Model is an evolutionary software process model that combines the iterative nature of prototyping and the controlled and systematic aspects of waterfall model. Thus it combines the advantages of top-down as well as bottom-up approach. The Spiral Model was proposed by Boehm. In this model, the focus is on risk assessment and minimizing risk by breaking a project into smaller segments and providing more ease-of-change during the development process. Each cycle involves a progression through the same sequence of steps, for each portion of the product and for each level of elaboration, from an overall concept-of-operation document down to the coding of each individual program. Each trip around the spiral traverses four basic quadrants:

1. determine objectives, alternatives, and constraints of the iteration;
2. evaluate alternatives; Identify and resolve risks;
3. develop and verify deliverables from the iteration; and
4. plan the next iteration.

The steps in the spiral model can be generalized as follows:

1. The new system requirements are defined in as much detail as possible. This usually involves interviewing a number of users external and internal and also other aspects of the existing system.
2. A preliminary design is created for the new system.
3. First prototype of the new system is constructed from the preliminary design. This is usually a scaled-down system, and represents an approximation of the characteristics of the final product.

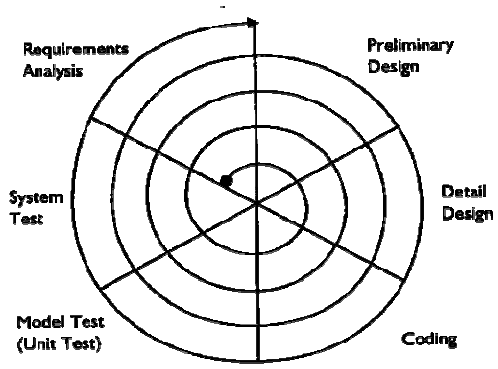


Fig 3.7 Simple Spiral Model

4. A second prototype is the result of a fourfold procedure:

- evaluating the first prototype in terms of its strengths, weaknesses, and risks;
- defining the requirements of the second prototype;

- planning and designing the second prototype;
- constructing and testing the second prototype.

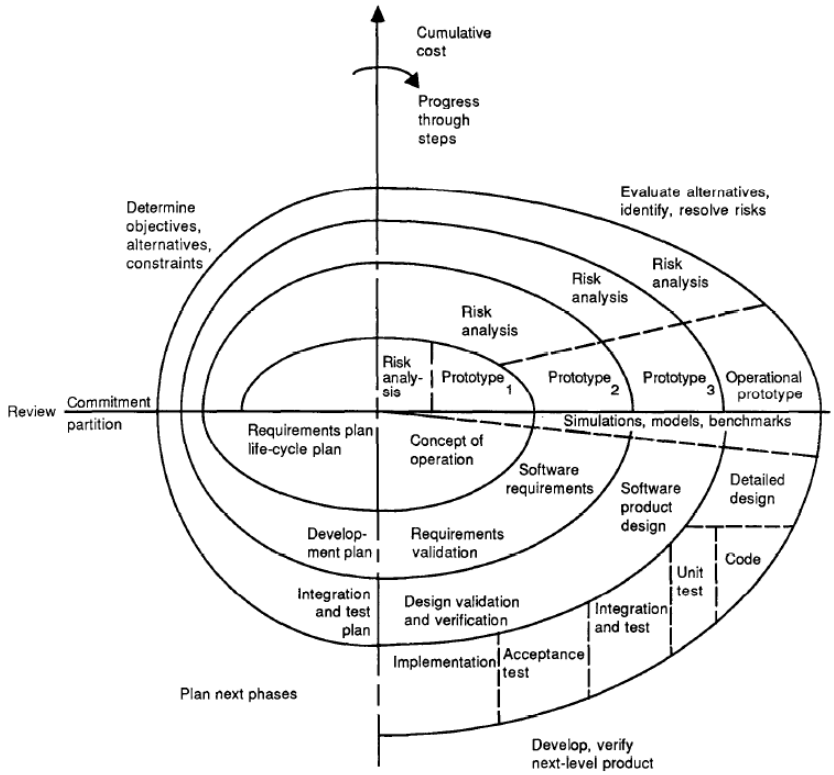


Fig 3.8 Spiral Model in Detail

The spiral model is intended for large, expensive and complicated projects. The progress of the process helps the developer and customer to better understand and react to risks at each level.

Advantages of the Spiral Model

- Its primary advantage is that its range of options accommodates the good features of existing software process models.
- In appropriate situations, the spiral model becomes equivalent to one of the existing process models. In other situations, it provides guidance on the best mix of existing approaches to a given project.
- It focuses early attention on options involving the reuse of existing software.
- It accommodates preparation for lifecycle evolution, growth, and changes of the software product.

- It provides a mechanism for incorporating software quality objectives into software product development.
- It eliminates errors and unattractive alternatives.
- It takes care both of software development and software enhancement or maintenance.
- It provides a viable framework for integrated hardware-software system development.

Areas of Concern

The three primary challenges of the spiral model involve the following:

- Matching contract software: The spiral model works well on internal software developments but needs more work to match it to the world of contract software acquisition.
- Relying on risk-assessment expertise: The model relies heavily on the ability of software developers to identify and manage sources of project risk.
- The need for further elaboration of spiral model steps: The model process steps need further elaboration to ensure that all software development participants operate in a consistent manner. .

Data Oriented Systems Development: Transaction processing is integral to many organizations involving both the data and the processing system. Approaches to system development can be categorized as follows:

- Data Oriented Approach
- Process Oriented Approach

Data Oriented approach to system development is a method for representing software requirements by focusing on data structure and not on data flow. . It portrays the ideal organization of data, independent of where and how it is used. The data model describes its kinds their business relationships. Here the business rules depict how (organization?) captures and processes data. Systems that optimize data usage are called data-oriented systems. This approach considers data independently of the processing that transforms it. . Management Information Systems (MIS) and Data Warehousing applications fall in this category. (It is very repetitive. I have tried to change it. Please check it.)

Process Oriented approach to system development focuses on flow, use and transformation of data. It also specifies how it is moved and / or changed in the system. It involves creating graphical representations such as data flow diagrams and charts. The data are tracked from sources, through intermediate steps to final destinations. Here the natural structure of data is not specified.

The following diagrams illustrate the above mentioned aspect.

Process Oriented Approach

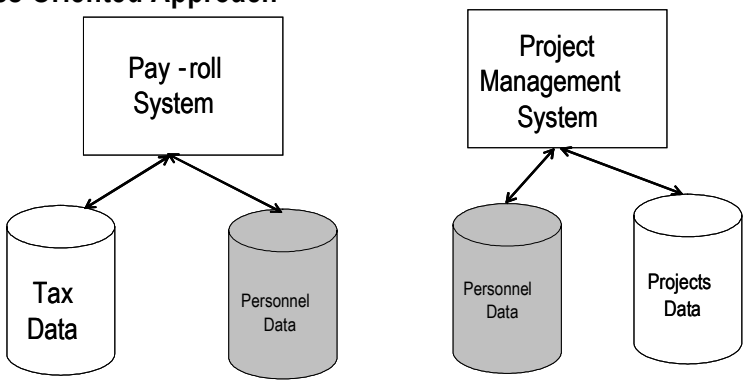


Fig 3.9 Process Oriented Approach

Data Oriented Approach

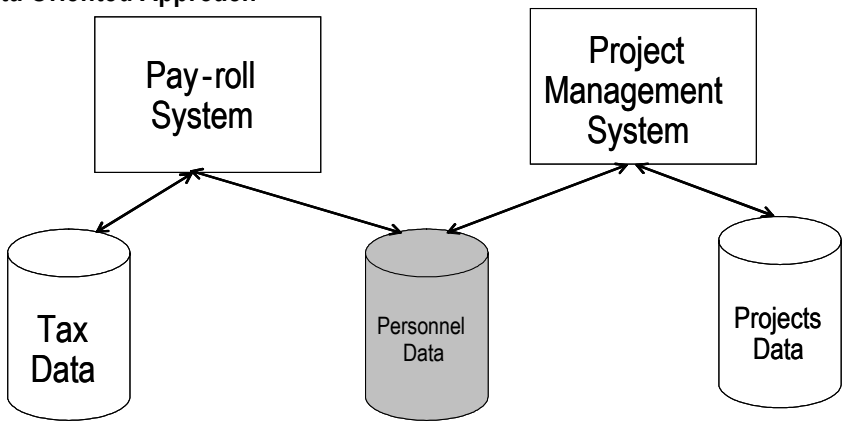


Fig 3.10 Data Oriented Approach

In Data oriented approach, the emphasis is on data being used across the system rather than its flow through a system's processes. This way, its redundancy and inconsistency can be reduced. However, this approach may not be feasible for all systems, because it is difficult to pre-decide which data items will be useful in other systems too.

Alternatively, this approach can be used for systems wherein processing logic is not important because the emphasis is on data management. E.g., in a library of

books or a medicine information system for a chemists shop, developers may design data structures directly without going through the process of SDLC phases. However, if transaction processing is also to be done in these systems, SDLC has to be followed. .

A data oriented approach is generally used for data intensive applications, such as database applications, but may also be used for handling human-computer interactions and comprehensive interfaces to other systems. Object oriented data design may structure the data according to objects and relationships, with attributes and behaviour as subordinate details. Inheritance and information hiding may not be appropriate for data design. Other data oriented approaches may be relational or equational.

A process oriented approach typically starts with function block and process design, and adds channels, interfaces, behaviour and data types. Some approaches may be abstract, like process algebra, while others may be close to implementation, i.e., much like program design. Program design oriented methods may provide good control over the use of (hardware) resources, real time operations and quality of each process.

Because of the centralized design of data, data oriented approach can provide better harmonization of the design of the entire system and its interfaces than the process oriented approaches, which focus on correct functioning of each process. Both may be applied to one and the same application area.

Re-engineering: In general, to reengineer is to rethink, to redefine, to redesign, to radically change the way work gets done as shown in fig x.x. In other words, re-engineering is the systematic transformation of an existing system into a new form to realize quality improvements in operation, system capability, functionality, performance, at a lower cost, or risk to the customer.

In the context of information system, re-engineering refers to the modification of a software system that takes place after it has been reverse engineered to add new functionality or to correct errors. It can also be defined as the examination and alteration of a system to reconstitute it in a new form. Reverse engineering is the initial examination of the system, and reengineering is the subsequent modification.

Many large organizations have a large number of applications that were developed 15-20 years ago, and have undergone many modifications during this time. In many of these applications structured programming could have been violated. It is difficult to migrate these huge mission critical applications to new systems quickly.

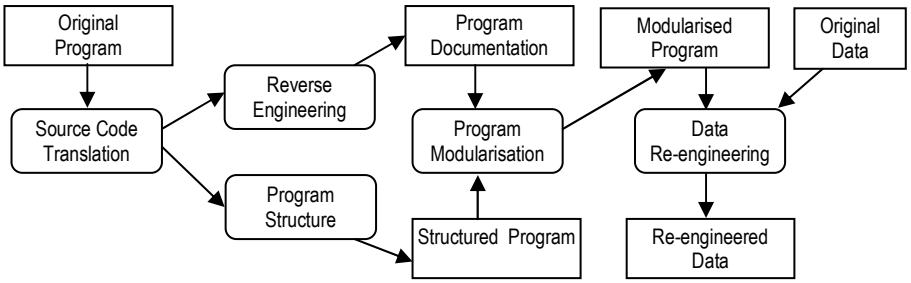


Fig 3.11 Reengineering Activities

In the current business scenario, the companies must keep re-inventing themselves according to the changing business needs and at the same time preserve their IT investments. With the coming up of newer technologies, it is imperative that businesses realize that automation of their operations and processes will lead to better interaction with the customers as well as realize their full potential..

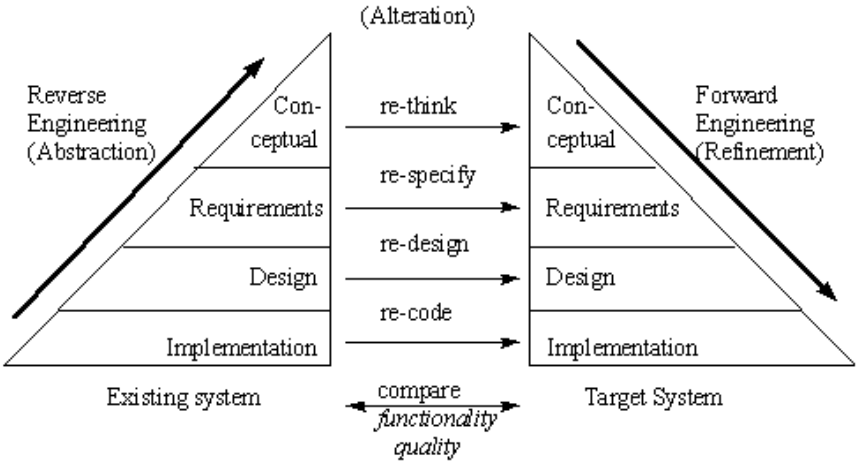


Fig 3.12 Relationship between Reverse Engineering, Reengineering & Forward Engineering

It is important for organizations to update and migrate their business systems to more recent technology platforms so that they do not become obsolete, which can be done by reengineering. It is like remodeling / rebuilding an old house.

In the process of re-engineering, the input to the process is a legacy program and the output is its structured, modularized version. At the same time, the data for the system may also be re-engineered.

The following six activities are carried out under software reengineering:

Inventory analysis: Every organization has an inventory of all applications that it uses. This includes details such as size, age, business criticality. Applications for reengineering can be sorted and selected on the basis of business criticality, age, current maintainability and other locally important criteria. . This analysis should be done on a periodic basis because factors such as current maintainability can change significantly.

Document restructuring: In many legacy applications, documentation is unclear, or non-existent. . In a large application environment, documentation must be carefully planned, taking into account the resources available. Static applications can be left with current level of documentation, but for dynamic applications, the documentation is updated, as and when the program is being modified. For critical applications, the documentation may be reworked completely.

Reverse Engineering: Reverse engineering for software is similar to reverse engineering of hardware, in which the hardware product is disassembled to understand its design and manufacturing. Similarly, in reverse engineering of software the design of the existing software is recovered in order to create a representation of the software at a higher level of abstraction. The reverse engineering tools extract data, architectural and procedural design information from the existing software. This engineering focuses on the challenging task of understanding legacy program code.

Code Restructuring: In code restructuring, only individual modules are restructured. Sometimes, legacy software can have solid program architecture but its individual components could be coded in such a way that it becomes difficult to understand the module(s) test & maintain them. In such cases, code restructuring comes into the picture. Restructuring tool is used to analyze the source code and violation of structured programming constructs, which are further restructured manually or automatically. The restructured code thus produced is reviewed and tested for further anomalies, if any.

Data Restructuring: Data restructuring is a full scale reengineering activity in which the current data architecture is dissected and necessary data models are defined. Data objects and attributes are identified, and existing data structures are reviewed for quality.

Forward Engineering: Forward engineering is a set of engineering activities that consume products and artifacts derived from legacy software and new requirements to produce a new target system. It is a process of moving from

high-level abstractions and logical, implementation-independent designs to the physical implementation of a system. Forward engineering is different from software engineering. Software engineering follows the sequence of events of normal development lifecycle. Forward engineering uses the output of reengineering. For example, the most common forward engineering activity involves the generation of source code from design information which was captured by a previous reverse engineering activity.

Some factors that affect re-engineering costs are:

- The quality of the software to be re-engineered: the lower the quality of the software and its associated documentation (if any), the higher the reengineering cost..
- The tool support available for re-engineering: It is not normally cost effective to re-engineer a software system unless we use CASE tools to automate program changes.
- The extent of data conversion required:. Since re-engineering requires conversion of large volumes of data, it increases the process cost.
- The availability of expert staff: If the staff responsible for maintaining the system cannot be involved in the re-engineering process, this will increase cost. System re-engineers will have to spend a great deal of time understanding the system.

The benefits of re-engineering process are:

Lower costs: Evidence from a number of projects done worldwide suggests that reengineering costs significantly less than new system development. Ulrich, for example, reports on a reengineering project that cost $12 million, compared to the estimated redevelopment cost of $50 million.

Lower risks: Reengineering is based on incremental improvement of systems, rather than on radical system replacement. The risk of losing critical business knowledge embedded in a legacy system is drastically reduced.

Better use of existing staff: Existing staff expertise can be maintained and extended to accommodate new skills during reengineering. The incremental nature of reengineering means that existing staff skills can evolve as the system evolves. The approach is less risky and less expensive because it saves hiring new staff.

Revelation of business rules: As a system is reengineered, business rules that are embedded in the system are rediscovered. This is particularly true of the rules that govern exceptional situations.

Incremental development: Reengineering can be carried out in stages, as resources become available. The operational organization always has a working system, and end users are able to gradually adapt to the reengineered system as it is delivered in increments.

Some of the disadvantages of re-engineering process are:

- There are practical limits to the extent to which a system can be improved by re-engineering. For example, it is not possible to convert a system with a functional approach to an object-oriented system.
- Since major architectural changes in the system of data management cannot be carried out automatically, it involves high additional costs. Major architectural changes of the system of data management have to be done manually.
- Although re-engineering can improve maintainability, the reengineered system will not be as maintainable as a new system developed with modern software engineering methods.

Reverse Engineering: Reverse engineering is the process of analyzing software with the objective of recovering its design and specification. The program itself remains unchanged. In other words, we can say that it is the technique of drawing design specifications from the actual product by studying its source code. The program is first analyzed and then design specifications are worked out.

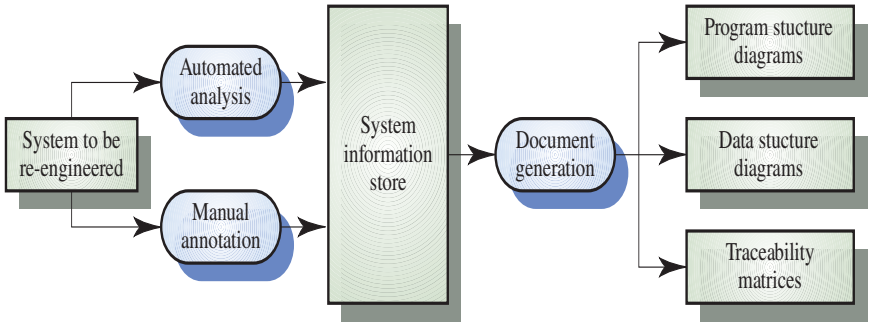


Fig 3.13 Reverse Engineering Process

The main difference between re-engineering and reverse engineering is that the objective of re-engineering is to produce a new, more maintainable system, whereas the objective of reverse engineering is to derive the design or specification of a system from its source code. Reverse engineering is used during the software re-engineering process to recover the program design which engineers use to understand a program before re-organizing its structure.

However, reverse engineering need not always be followed by re-engineering. The design and specification of a system may be reverse engineered so that they can be an input to the requirements of specification process for the system's replacement. The design and specification may be reverse engineered to support program maintenance.

This process can be carried out by:

i Decomposing the object or executable code into source code and using it to analyze the program.
ii. Utilizing the reverse engineering application as a black box test and unveiling its functionality by using test data.

The purpose of reverse engineering involves

- Security auditing
- Removal of copy protection also called cracking
- Customization of embedded systems
- Enabling of additional features on low-cost hardware

The advantages of reverse engineering are faster development of a system and its improvement. The IS auditor should be aware that many software license agreements prohibit reverse engineering to safeguard their trade secret and programming techniques.

Structured Analysis: This has already been discussed in detail in Chapter-2.

Agile Development: Agile development refers to a family of similar development processes that involve a non-traditional way of developing a complex system. One of the first agile processes which emerged in 1990s was Scrum, which aimed at planning and directing tasks from the project manager to the team thus helping the project manager to remove obstacles faced by the team to achieve their objectives. Other agile processes are Extreme Programming (XP), Crystal, Adaptive Software Development, Feature Driven Development and Dynamic System Development Method.

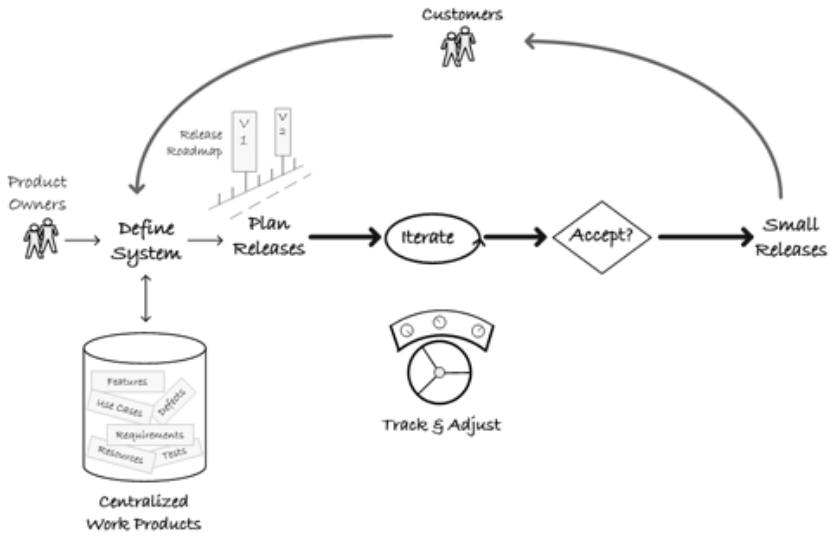


Fig 3.14 Agile Development Process

While building a critical software system, all project requirements cannot be accurately gathered at the beginning of the project. Not all the details may be known to the developers prior to building a software system. It is because the client's business may still be growing and certain aspects of the system might be changing. Moreover, business climates and objectives often change rapidly, especially in the present age of instant information. These days clients want to see and feel the product (software system) before they decide on what they want. The "I'll Know it When I See It" (IKIWISI) law says that software development clients can better describe what they really want after seeing and trying working, functional software. This problem of undefined, changing, and emerging requirements can be addressed by the agile software development process.

These processes are called "agile" because they are designed to handle changes to the system being developed or the project team that is performing the development.

Agile software development methodologies embrace iterations. Small teams work together with stakeholders to define quick prototypes, proof of concepts, or other visual means to describe the problem to be solved. The team defines the requirements for iteration, develops the code, and defines and runs integrated test scripts, and the users verify the results. Verification occurs much earlier in the development process than it would with waterfall, allowing stakeholders to fine-tune requirements while they're still relatively easy to change.

The principles of agile software development are:

- Satisfy customers through early and frequent delivery.
- Welcome changing requirements even at a late stage in the project.
- Keep delivery cycles short, for example, weekly or fortnightly.
- Business people and developers work together daily throughout the project, giving them the environment and support they need, and trusting them for getting the job done.
- Build projects around motivated individuals.
- Place emphasis on face-to-face Communication. The most efficient and effective method of conveying information to and within development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. Sponsors, developers, and users should be able to maintain a constant pace.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity "the art of maximizing the amount of work not done" is essential.
- The best architectures, requirements, and designs emerge from self organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

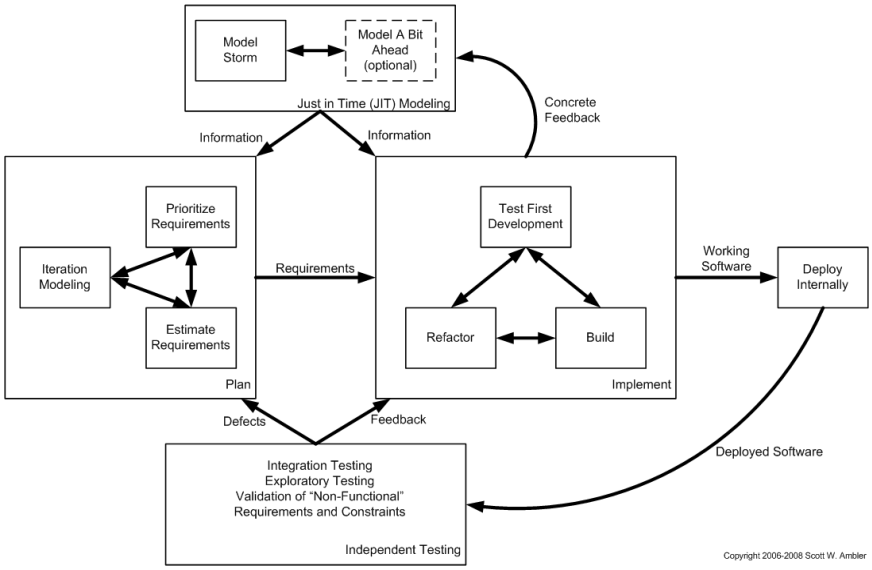


Fig 3.15 Data and Process flow in Agile Development

Agile development process involves:

1. Setting up of small subprojects or iterations, which become the basis of the next iteration. Re-planning the project at the end of each iteration which involves resetting priorities, identification of new priorities, etc.
2. The teams are generally small, cohesive and comprise both business and technical representatives.
3. In some cases of agile development, two programmers code the same part of the system as a means of knowledge sharing and quality improvement.

Unlike a normal project manager who has the role of planning a project, allocating tasks and monitoring its progress, the project manager has the job of a facilitator and advocate.

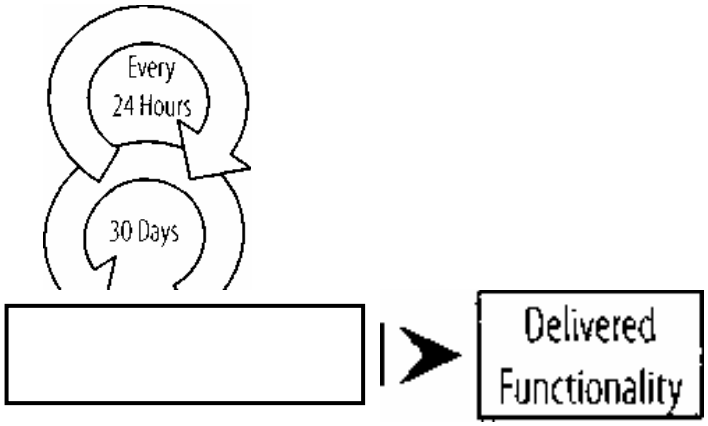


Fig 3.16 Agile Development

Agile development does not ignore the concerns of traditional software development; it approaches them from a different perspective.

1. Agile development plans for the next development in detail, rather than planning subsequent development phases far out of time.
2. Agile development's adaptive approach to requirements does not emphasize managing baseline requirements.
3 Agile development's focus is to quickly prove architecture by building actual functionality against development of formal or more detailed models and descriptions.
4. Agile development assumes limits to detect testing but attempts to validate functions through a frequent build test cycle and corrects problems in the next subproject before incurring too much time and cost.

5. Agile development does not emphasize defined and repeatable process and instead it performs and adapts its development based on frequent inspections.

So far, we have dealt with the phases in SDLC, and how these have been undertaken in traditional and alternative development models.

Latest System Development Approaches

The following sections will deal with the latest development methodologies which are appropriate to the recent technological inventions. Some writers classify these as alternate development methodologies since they are not done in a traditional manner. However, SDLC phases are not eliminated in these methodologies, only fine tuned to suit newer techniques. The three latest development methodologies are:

1. Object Oriented Systems Development
2. Component Based Software Development
3. Web Based Application Development

1. Object Oriented Systems Development

Object oriented systems development is an extension of structured programming. In this method, the system is analyzed in terms of objects and classes and the relationship between objects and their interaction. Objects are entities that have both data structure and behaviour. Conceptually, an object has a fixed or defined set of properties. For example, if an employee record is an object, then the properties are employee name, employee ID etc., and the behaviour of the Employee object would form Methods (i.e. procedures) such as AddEmployee, RemoveEmployee, TransferEmployee and so on.

Object oriented development emphasizes the benefits of modular and reusable computer code and modeling real-world objects, just as structured programming emphasizes the benefits of properly nested structures.

In the object oriented approach, it is easier to establish relationships between any two entities of a program, whether it is a database or an application program, where each entity is considered an object. In this way, two similar objects can be easily associated. It allows inheritance of properties from one object to create another, which simplifies the process of creating newer objects with all the attributes of the older one, and some additional attributes too. For example, the principles of compound interest can be inherited from simple interest.

What object technology promotes is the ability to build applications by selecting and assembling objects from libraries. If a developer must create a missing object to meet the application's requirements, that new object may be placed in a library for reuse in future applications. For many simple systems, the developer may use the available objects to form the entire application instead of a writing code. More complex development efforts require the developer to modify the objects to meet specific requirements.

Object orientation is much more than an entry into a program. It can apply across every system development activity, including requirements analysis, design, testing, and business process reengineering. Developing an Object Oriented application requires giving more thought to the design than developing in the traditional structured programming environment, because the focus on future reuse requires a long-term view during analysis and design. However, a well-stocked library of reusable components reduces the need to perform original analysis and design.

Object-oriented software development life cycle consists of the following:

- Object-oriented analysis
- Object-oriented design
- Object-oriented implementation

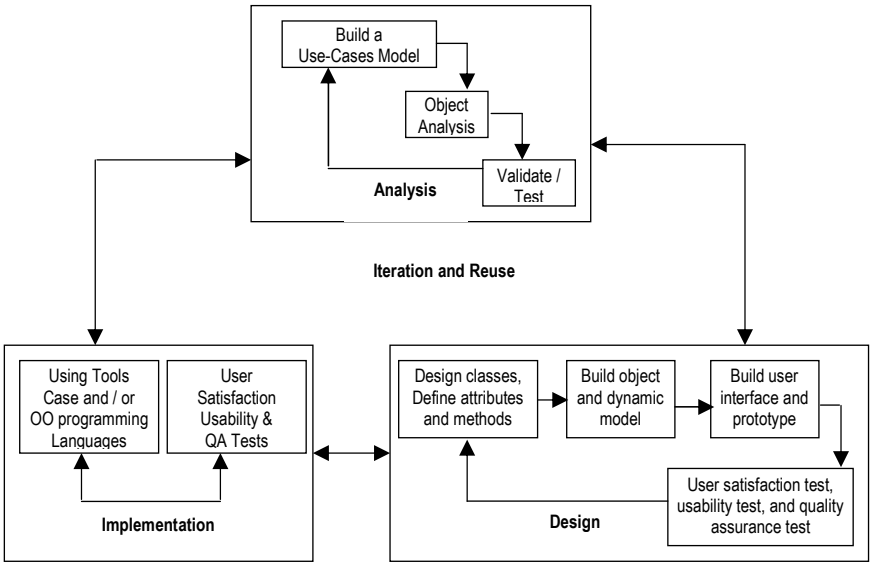


Fig 3.17 Object-Oriented Software Development Life Cycle

Major advantages of this approach are:

- The model allows full exploitation of the power of object-based and object-oriented programming languages.
- Object oriented development model can manage a variety of data types.
- It has the ability to manage complex relationships.
- It has the capacity to meet demands of a changing environment.
- Since object-based models appeal to the workings of human cognition, human input into the development of a software system is likely to be more natural and less prone to error.
- It encourages the re-use of modules and also of entire designs, which leads to reduced development and maintenance costs.
- Object-oriented systems are based on stable forms (i.e. objects and classes) which are resilient to change.
- Delaying decisions about representation of objects and hiding as much information as possible within an object leads to strongly cohesive and weakly coupled software, which is easier to modify.
- Data Security aspect is also taken care of by implementing object oriented technology.

Object oriented technology is widely used in:

- Computer Aided Engineering (CAE)
- Systems software

This approach is gaining popularity in application development as well.

2. Component Based Software Development

Component based software development is similar to engineering component approach. It deals with the assembly of pre-existing software components into larger pieces of software. Software components are written in such a way that they provide functions common to many different systems.

It emphasizes decomposition of the engineered systems into functional or logical components with well-defined interfaces that are used for communication across them. Components are a higher level of abstraction than objects and as such they do not share state and communicate by exchanging messages carrying data.

Just as various components are assembled together to form a usable machine, software components are used to build a usable end-user system. For example, a vehicle is an assembly of various components such as engine, chassis, wheels, etc. Each of these components is also an assembly of components, similar to

systems and sub-systems. In a similar way, software components can be built to form a module of a system and several modules can then lead to a component based end-user system.

A software component is a system element offering a predefined service and able to communicate with other components. The components are designed in such a way that they have a multiple-use and can be combined with other components, encapsulated, i.e., inside parts are not visible.

Software components often take the form of objects or collections of objects (from object-oriented programming), in some binary or textual form, adhering to some Interface Description Language (IDL) so that the component may exist autonomously from other components in a computer.

Reusability is an important characteristic of a high quality software component, which is designed and implemented in such a way that it can be reused in many different programs.

Component-based software development encompasses two processes:

1. Assembling software systems from software components and
2. Developing reusable components.

The activity of developing systems as assemblies of components may be broadly classified in terms of four activities.

1. Component qualification,
2. Component adaptation,
3. Component assembly, and
4. System evolution and maintenance.

Qualification: Qualification is the process for determining the suitability of a component for use within the intended final system. In a marketplace of competing products, selecting the most suitable component becomes a necessity. The selection dependents on the condition that the various methods for comparing and evaluating the fitness of components used should also exist. It is during this activity that the issues of trust and certification arise. The process of certification is two-fold: (1) to establish facts about a component and to determine that its properties are in conformity with its published specification; and (2) to establish trust in the validity of these facts by having them confirmed as a trusted third-party organization. The motivation for component certification is that there is a causal link between a component's certified properties and the properties of the end system. If enough is known about the (certified) components selected for assembly then it may be possible to predict the properties of the final assembled

system. For many components in the marketplace, prediction is difficult because of a lack of information about the capabilities of a component and lack of trust in this information. Conventional software doctrine states that component specifications should be sufficient and complete, static — writable once and frozen, and homogenous. However, full specifications may be impractical: some components may exhibit (non-functional) properties which are infeasible to a document. One method for addressing this issue is to use credentials i.e., knowledge-based specifications that evolve as more are discovered about a component.

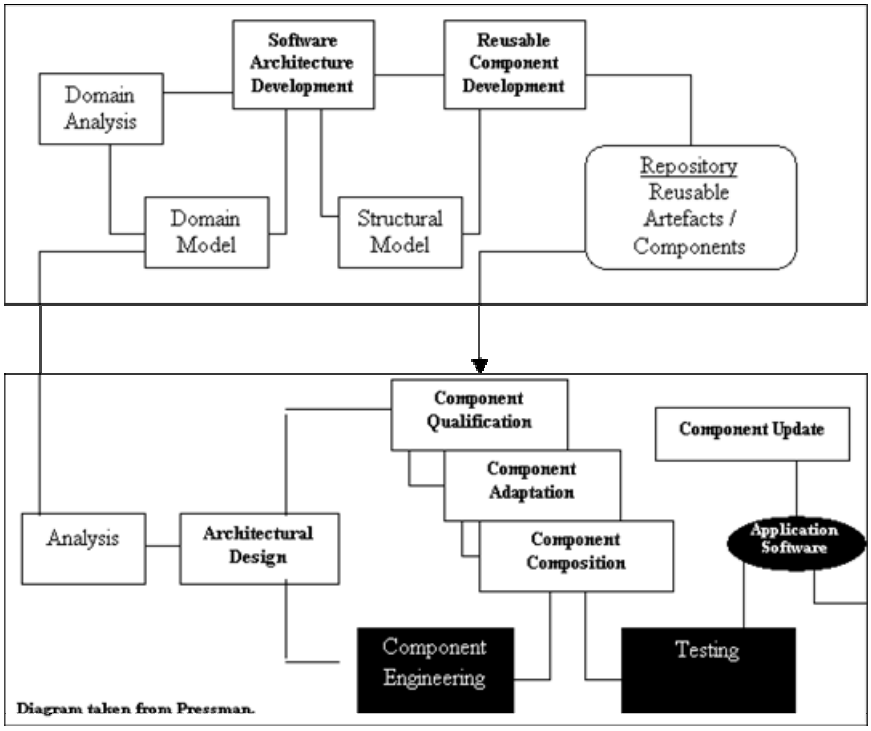


Fig 3.18 Activities of Component-Based Software Development

Adaptation: Individual components are written to meet different requirements, each one making certain assumptions about the context in which it is deployed. The purpose of adaptation is to ensure that conflicts among components are minimized. Different approaches to adaptation depend upon the accessibility of the internal structure of a component. White-box components may be significantly rewritten to operate with other components. Grey-box components provide their own extension language or Application Programming Interface (API). Black-box,

or binary, components have no extension language or API. Ideally, a component is a black box and its services are only accessible through some well-defined interface.

Assembly: Assembly is the integration of components through some well defined infrastructure, which provides the binding that forms a system from disparate components. COTS components, for example, are usually written to some component model defined by say Enterprise JavaBeans, COM, CORBA, or NET.

System Evolution and Maintenance: Because components are the units of change, system evolution is based around replacing outdated components by new ones. The treatment of components as plug-replaceable units is a simplistic view of system evolution. In practice, replacing a component may be a non-trivial task, especially when there is a mismatch between the new component and the old one, triggering another stage of adaptation with the new component.

Component-based software development promises to reduce development costs by enabling rapid development of highly flexible and easily maintainable software systems.

Does it differ from OOP?

The idea in OOP is to create objects which are usable in other application systems as well. For example, if a dialogue box is designed to accept user name and password as an object, then it can also be used in other systems where user name and password are required. A component is not just one object but a collection of objects developed using OOP techniques as well as other constructs which do not need to have been developed by using OOP technique. For example, a calculator is a component which can be used in many applications wherever a calculator is a means for providing calculation capacity within an application system. It may contain objects as well as programs developed with a traditional approach. The focus is on building a component and not objects.

A good deal of effort and awareness is required to write a software component that is effectively reusable. IS auditors need to note the following points in component based development methodology:

- Components should be fully documented
- Should be thoroughly tested
- Must have robust input validity checking
- Appropriate error handling should be ensured
- Should be built by keeping in mind with an awareness that it *may* be put to unforeseen uses;

The phases of SDLC such as feasibility study, requirements analysis may have to be revisited to suit component based development. For example, in a component-based approach, it is necessary to analyze whether these requirements can be fulfilled by available components. This means that the analysts have to be aware of the components that can possibly be used. Since appropriate components may not be available, some other components have to be implemented, which can be risky. IS auditors should be aware of this and carefully study if a compromise has been made for correct functioning of the software. IS auditors should also ensure that unit testing (which now applies to individual component) and integration testing (integration of components to build system modules) have a significant role in the entire testing phase.

For example, in Microsoft's Transaction Server, one component is the component Move Money. It moves money, i.e., a number, from source account to destination account for a given transaction. A programmer who wishes to move a number from one source to another can use this component and build his module.

3. Web-Based Application Development

The World Wide Web and the Internet have added to the value of computing. With the help of web based applications, we can purchase shares, download music, view movies, get medical advice, book hotel rooms, schedule airline flights, do banking, take college courses, etc.

Web-based systems and applications become integrated in business strategies for small and large companies. The following are the attributes of the Web based applications:

1. **Network Intensive:** By its nature, a web based application is network intensive. It resides on a network and serves the needs of diverse community of clients. It may reside on the internet (thereby enabling open worldwide communication) or intranet (implementing communication across the organization) or extranet (making available intranet for external users with proper access controls.)
2. **Content Driven:** In many cases, the primary function of a web based application is to use hypermedia to present text, graphics, audio, and video contents to the end user.
3. **Continuous evolution:** Unlike conventional application software that evolves over a series of planned, chronologically spaced releases, web based applications evolve continuously.

The following application categories are most commonly encountered in Web based applications:

- **Informational:** Read only content is provided with simple navigation and links.
- **Download:** A user downloads information from the appropriate server.
- **Customization:** The user customizes contents to specific needs.
- **Interaction:** Communication among a community of users occurs via chat-room, bulletin boards, or instant messaging.
- **User Input:** Forms based input is the primary mechanism for communicating users' need, like, in the form of query.
- **Transaction oriented:** The user makes a request (e.g. places an order) that is fulfilled by the web based application.
- **Service Oriented:** The application provides a service to the user (e.g. assists the user in calculating the EMI of loan).
- **Portal:** The application channels the user to other web content or services outside the domain of the portal application
- **Database Access:** The user queries a large database and extracts information.
- **Data Warehousing:** The user queries a collection of large databases and extracts information.

The web based application development process begins with the formulation of an activity that identifies the goals and objectives of such development, planning estimates of overall project cost, evaluates risks associated with the development effort, and defines a development schedule. Analysis establishes technical requirements for the application and identifies the content items that are to be incorporated in it. The engineering activity incorporates two parallel tasks: content design and technical design. Page generation is a construction activity that makes heavy use of the automated tools for the web based application and testing ensures web application navigation, attempting to uncover errors in function and content, while ensuring that the web based application operates correctly in different environments. Web engineering makes use of an iterative, incremental process model because the development timetable for web based applications is very short.

In the case of web based Applications, the clients database resides on a back end processor and the software and the data related to the frequently asked information may reside on a front end (unlike client server based applications where both may reside on the same processor) to save the time of the user.

Risks associated with Web Based Applications:

1. As the web based applications are available via network access, it is difficult to limit the possible end users who may access the application. In order to

protect the sensitive content and provide secured modes of data transmission, strong security measures must be implemented throughout the infrastructure that supports a web based application and within the application itself.

2. In the absence of disciplined process for developing web based systems, there is an increasing concern that we may face serious problems in the successful development, deployment and maintenance of these systems. Poorly developed web based applications have too high a probability of failure. As web based systems grow more complex, a failure in one can propagate broad based problems across many. In order to avoid this, there is a pressing need for disciplined web engineering approaches and new methods for development, deployment and evaluation of web based systems and applications.

4. Extreme Programming

Extreme Programming (or XP) is a set of values, principles and practices for rapidly developing high-quality software that provides the highest value for the customer in the fastest way possible. XP is extreme in the sense that it takes 12 well-known software development "best practices" to their logical extremes.

The 12 core practices of XP are:

1. **The Planning Game**: Business and development team cooperate to produce the maximum business value as rapidly as possible.
2. **Small Releases**: XP start with the smallest useful feature set which is released early. Further, during every fresh release, new features are added to the system.
3. **System Metaphor**: Each project has an organizing metaphor, which provides an easy to remember naming convention.
4. **Simple Design**: It is always better to use the simplest possible design for successful completion of job. The requirements may change in future, so it is better to concentrate on today's requirements for development of the system.
5. **Continuous Testing**: Before programmers add a feature, they write a test for it. They test the feature using that test for successful execution.
6. **Refactoring**: Refactor out any duplicate code generated in a coding session.
7. **Pair Programming**: All production codes are written by two programmers sitting at one machine. Essentially, all code is reviewed as it is written.
8. **Collective Code Ownership**: No single person "owns" a module. Any developer is expected to be able to work on any part of the code base at a particular time.

9. **Continuous Integration**: All changes are integrated into the code base daily. The tests have to run 100% both before and after integration.
10. **40-Hour Work Week**: Programmers go home on time. In crunch mode, up to one week of overtime is allowed. But multiple consecutive weeks of overtime are treated as a sign that something is very wrong with the process.
11. **On-site Customer**: Development team has a continuous access to a real live client, that is, someone who will actually be using the system. For commercial software with lots of clients, a client proxy (usually the product manager) is used.
12. **Coding Standards**: Everyone write program codes using the same standards.

5. XML – Extensible Markup Language

XML, also known as Extensible Markup Language, is a general-purpose markup language. It is an extensible language because it allows its users to define their own tags. Its primary purpose is to facilitate the sharing of structured data across different information systems, particularly via the Internet. XML was designed to define and describe data whereas HTML was designed to display data and to show how it looks.

XML documents contain data and tags to process it. For example, see the following text in an XML document:

<note>

<to>General Manager</to>

<from>Avinash</from>

<body>Can we meet on this Friday at 2.00 PM?</body>

</note>

The above XML document will be processed to read the tags and display the data contained in the tags as follows:

To: General Manager

From: Avinash

Can we meet on this Friday at 2.00 PM?

For any XML document XML parsers (processing programs) will derive its meaning and do as per the tags within the document and process the data. E.g., XBRL is eXtensible Business Reporting Language which is based on XML and can be used for reporting business data (accounts, costing, legal etc) uniformly

across all computer systems irrespective of what their hardware and software systems.

Web Services Description Language (WSDL), is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. It defines services as collections of network endpoints or ports. WSDL specification provides an XML format of documents for this purpose.

Information Systems Maintenance Practices

After a system moves into production, it seldom remains static. Change is a reality, and systems undergo changes right through their life cycle. These changes often create problems in the functionality and other characteristics of a system. So it is necessary that a procedure for change is formalized.

Change Control

Any request for change by the user has to be submitted to the EDP department, along with the reasons for this. . (In case the developer himself wants to change the program to overcome a processing problem or to improve performance, then he has to prepare a written change request document).

The user request is then assessed by the relevant application developer. He evaluates the impact of the modifications on other programs. The number of days required for making all the necessary changes, and time for testing and changes in documentation is also estimated. A report is then prepared by the developer on the basis of time and cost of change.

Every organization has a defined CCA (Change Control Authority). CCA is a person or a committee who is the final authority that approves changes. The CCA reviews the report and approves / rejects the change request. An Engineering Change Order (ECO) is then generated for the changes approved.

The ECO describes the change that should be made; constraints within which change should be made; and criteria for review and audit. The program(s) to be changed is / are then copied to the test directory from the production directory with access control for the designated programmer.

The programmer then makes the approved changes, and the programs go through all the tests that they had gone through, when they were initially developed. If a program change warrants a change in the database, then it is first made in the test data base, and all related documents are changed.

The CCA then reviews the changes made to programs, data and documents. After it is approved, the systems administrator moves the changed version into the production directory, and informs all users of the change and the revised version number.

After running the new version of the application the user who requested the change certifies that the change requested by him has been made.

Maintenance records of all programs exist in hard copy or in the machine. The maintenance records include the programmer id, time and date of change, change request number, and before and after images of program lines, which were changed. Library management software is available, which help in controlling the changes on the machine.

Automated change control software is useful in preventing unauthorized program changes. This is particularly necessary if the role of the systems administrator has not been well defined and programmers themselves run the application. The change control software will then be responsible for migrating changed programs to the production directory and not operations personnel. In SME sector, generally, IT department is handled by fewer people. In such cases it is impossible to adhere to segregation of duties and hence in such companies automated change control software can be deployed to transport the changed code to production environment. Audit trails can also be included in automated change control software.

Continuous Update of Systems Documentation[R3]

To maintain the systems effectively, documents should reflect the current version of the software. Often because of time and other resource constraints, as well as tedious procedures, the result in programs is not updated. Documents that have to be kept updated are:

- System manual
  - Input screens
  - Output reports
  - Program specifications
  - Flow charts, Decision tables, Decisions trees
  - Narrative of program logic
  - Data dictionary
  - Entity relationship diagram
  - Object diagrams
  - Data flow diagrams

- User manual
  - Data entry procedures
  - Batch operation procedures

The current version of the documentation should also be available on the back-up site.

In Literate programming, a computer program is written like literature, with human readability as its primary goal. Programmers aim at a "literate" style in their programming just as writers aim at an intelligible and articulate style in their writing. This is in contrast to the view that the programmer's primary or sole objective is to create source code and that documentation is a secondary objective.

In practice, literate programming is achieved by combining human-readable documentation and machine-readable source code into a single source file, to maintain close correspondence between documentation and source code. The order and structure of this source file are specifically designed to aid human comprehension: code and documentation together are organized in logical and / or hierarchical order (typically according to a scheme that accommodates detailed explanations and commentary as necessary). At the same time, the structure and format of the source files accommodate external utilities that generate program documentation and / or extract the machine-readable code from the same source file(s)

UML (Unified Modeling Language)

UML, Unified Modeling Language, is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is very important parts of developing object oriented software and software development process. The UML uses mostly graphic notations to express the design of software projects. Using the UML helps project teams to communicate, explore potential designs, and validate the architectural design of the software.

Goals of UML

The primary goals in the design of the UML are:

- Provide users with a ready-to-use, expressive visual modeling language so that they can develop and exchange meaningful models.
- Provide extensibility and specialization mechanisms to extend the core concepts.

- Be independent of particular programming languages and development processes.
- Provide a formal basis for understanding the modeling language.
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices.

Program Migration Process

During change control, the process of taking out one or more copies of program(s) from the production directory and moving them back to the production directory are important steps that require additional control.

These are also called 'check out' and 'check in'. Only an authorized software engineer can check out the program(s). This is controlled by the access control software. The systems administrator copies the program(s) to the test directory and controls the access to a particular user-id and password, which he discloses only to the authorized software engineer. Similarly, after the changes are made, it is the responsibility of the systems administrator to copy the changed version into the production directory. Then access for the specified user ID and password are disabled.

In the change control process, synchronization control stops parallel changes from being made by two different programmers. Once a program is checked out for change, synchronization control locks it against release for any other change. Only after the program is checked in, is it unlocked.

Testing Program Changes

While assessing the effectiveness of the change control procedure, the auditor should check if:

- All change requests are documented in a standard form containing:
  - Adequate description of the desired change
  - Explanation of reasons for the change request
  - Signature of the requestor
  - Date of request
- Cost and time analysis of change has been conducted by the developer.
- It is compliant with the approval process for change request
- Access rights and the rationale behind them have been clearly defined
- The process of checking and communicating change to the users is implemented.

- The configuration of audit process is done (a periodic review of the changes made to configuration items)

If the auditor so desires, he can take a change request and trace all activities from the documentation to assure himself that all the processes are being followed.

Library Control Software

The purpose of the library control software is to separate production libraries from test libraries.

The functions of this software are:

- It prevents programmers from accessing source and object programs in the production directory.
- It does not permit the program to be updated in bulk.
- It enforces discipline. The programmer, after making the requested change in the source code and testing, hands it over to the official authorized by the organization to update the production directory-control group or systems administrator. The production directory is then updated with the revised version of the source code and object.
- It provides read-only access to the source code. Any modification has to be authorized by the change control procedure detailed earlier.
- It maintains clear distinction between programs in production and test directories.

Executable and Source Code Integrity

At any point of time, the current version of the source code and object code should match. In a manual program migration practice, the changed source code may be moved to the production directory, but compilation is omitted. In such a case, the previous version of the object code continues executing. Thus, when the auditor checks the source code, it does not correspond to the object program. To overcome this difficulty Library Control Software is employed to automate the process of conversion. Once the source code is moved to the production directory, the software takes the responsibility of converting it to the object code.

Some of the controls the auditor uses to check in code integrity are:

- The time stamp on the object code should always be later than that of the corresponding source code.
- Users and application programmers should not have access to the production source code.

In an automated environment, where the users themselves develop applications, controls may be lax. So auditors should focus on evaluating controls in such applications.

Program Code Comparison

Auditors use program code comparisons to ensure that the source code provided to them for review is the same as the one that has been compiled to produce the current version of the object code.

The Program code comparison can be of two types:

- Source code comparison
- Object code comparison

Source Code Comparison

The IS auditor should check whether the current / latest version of the source code and the version that is available in production directory are the same. Software to check this is now available.

While manual checking is feasible in source code comparison, it may not be effective. In languages like COBOL, even column number will matter in the functioning of the program. In languages such as C, a punctuation mark can make a difference. Almost all languages permit comment lines with a prefix of asterisks. So lines of code that are prefixed with asterisks will be physically present but not functional. So if an auditor is attempting a manual check, he should be very thorough, taking care to verify the column number and punctuation.

While using software for code comparison, the auditor should know how to differentiate material and non-material changes between the codes. For example, change of position in a text does not make any difference in many languages. But the software will mechanically report this as a difference between the codes. The auditor should recognize that this change does not materially affect the functioning of the program.

Object Code Comparison

Object code comparison is quite difficult, because the object code is not readable. This is done through the following steps:

- After going through the source code comparison, the auditor compiles a copy of the source code in the production directory and generates an object code.
- The test plan for the program is then applied on the object code and the results must be documented.

- The same test is e applied on the object code available in the production directory.
- The results of both these tests should match.

Emergency Changes

Sometimes emergency changes may have to be made. If the management recognizes that there is not much time available to create a new version because of an extraordinary situation, they can relax the change control procedure. Normally, this is operated by a special log-in id, which is known only to the systems administrator and available in a sealed envelope with the EDP management. Within their approval the programmer may access the production directory using the special log-in ID and make the necessary changes in the program and run the new version to create outputs. These must be diligently logged in the machine. The follow-up procedure will include all necessary documentation and written approvals post-facto.

Configuration Management[R4]

Configuration management involves various procedures throughout the life cycle of the software to identify, define and baseline software items in the system thus providing a basis for problem management, change management and release management.

Configuration management process involves identification of items like programs, documentation and data. Configuration management team takes care of programs, documentation and data for safekeeping. Each is assigned a reference number for a quick retrieval. Once it goes to the team, the item cannot be changed without a formal change control process which is approved by a change control group.

CI, Configuration Identification, is selection, identification and labeling of the configuration structures and configuration items, including their respective 'owner' and the relationships between them. CIs may be hardware, software or documentation. These CIs will be stored in CMDB (Configuration Management Database) and will be used in configuration management and other services, such as incident handling, problem solving and change management. Any changes done to CIs will be updated in CMDB and CMDB will be kept up-to-date.

The goals of Configuration Management are:

- to account for all the IT assets and configurations within the organisation and its services.

- to provide accurate information on configurations and their documentation to support all the other Service Management processes.
- to provide a sound basis for Incident Management, Problem Management, Change Management and Release Management.
- to verify configuration records against infrastructure and correct any exceptions.

Implementing configuration management to the software systems is quite a tedious and difficult task. It is always cumbersome to identify and finalize the CIs. For example, whether to store programs as whole CIs or also to store individual functions or objects or even screens handled by the program, is a question that needs to be tackled. Generally, in ERP systems like SAP, CMDB of CIs is maintained by the SAP system itself. So changes to CIs will be automatically monitored by the native system.

The process of moving an item to the controlled environment is called checking in. When a change is required, the item will be checked out by the configuration manager. Once the change is made, it is checked in by a different version number.

Management's support is vital for the success of configuration management. The job profile of the maintainer involves the following task steps:

1. Develop the configuration management plan
2. Baseline the code and associated documents
3. Analyze and report the results of configuration control
4. Develop the reports that provide configuration status information
5. Develop release procedures
6. Perform configuration control activities, such as identification and recording of the request
7. Update the configuration status accounting database

Summary

A few of the alternate approaches to system development are: Data-oriented approach, object oriented design, prototyping, RAD (Rapid Application Development), Reengineering and Structure Analysis. In the object-oriented analysis, the system is analyzed in terms of objects and classes and the relationship between objects and their interaction. Object Oriented Technology is widely used in Computer Aided Engineering (CAE) and systems software. In prototyping, a set of general objectives for the software is defined, instead of listing detail input/output and processing requirements. RAD is an incremental model that supports a short development cycle. In this approach, the requirements must be clear and the scope must be well-defined. Reverse engineering or reengineering involves separating the components of a

system and observe its working with the intention of creating its replica or improving upon the original product. Structured analysis is a framework for the physical components (data and process) of an application. It is done by using data flow diagrams. Web based applications development deals with application systems that work on the network to make the information available to the users. Agile development involves development processes that are similar to the traditional way of developing a complex system.

Once a system moves into production, it seldom remains static. Any changes in it often create problems in its functionality. So it is essential that a systematic approach for maintaining the information system is formulated and implemented. These processes are collectively called information systems maintenance practices. They include change control, continuous updating of systems documentation, program migration process and testing.

Multiple Choice Questions:-

1. In _____ model, all the SDLC phases run serially and you are not allowed to go back and forth in these phases, and it most suited for traditional programming languages such as COBOL.
   a. Spiral Model
   b. Iterative Enhancement Model
   c. RAD Model
   d. Waterfall Model
2. _____ is a method for representing software requirements by focusing on data structure and not data flow while processing.
   a. Information Oriented System Development
   b. Data Oriented System Development
   c. Process Oriented System Development
   d. Method Oriented System Development
3. In _____ the developer gets initial requirements from the users.
   a. Quick Design
   b. Requirement Gathering
   c. Construction of Prototype
   d. Refinement of Prototype
4. RAD stands for _____.
   a. Reverse Application Development
   b. Requirement Application Development
   c. Rapid Application Development
   d. Reengineering Application Development

5. In _____ method, the system is analyzed in terms of objects and classes and the relationship between objects and their interaction.
   a. Object Oriented Systems Development
   b. Component Based Software Development
   c. Web Based Application Development
   d. Process Based System Development

6. A _____ is a system element offering a predefined service and is able to communicate with other components. (Check?)
   a. System Component
   b. Software component
   c. Object Component
   d. Web Component

7. By its nature, a _____ is network intensive and it resides on a network and must serve the needs of diverse community of clients.
   a. Object Oriented Application
   b. Component Based Application
   c. Web Based Application
   d. Process Oriented Application

8. _____is a set of values, principles and practices for rapidly developing high-quality software that provides the highest value to r the customer in the fastest way possible.
   a. Extreme Programming
   b. Extensive Programming
   c. Express Programming
   d. External Programming

9. XML stands for _____.
   a. Express Markup Language
   b. Extensible Makeup Language
   c. Extensible Markup Level
   d. Extensible Markup Language

10. UML stands for _____.
    a. Unified Modeling Language
    b. Unified Method Language
    c. Unified Medium Language
    d. Unified Model Language

11. Auditors use _____ to ensure that the source code provided to them for review is the same as the one that has been compiled to produce the current version of object code.
    a. Program Code Comparison
    b. Program Code Compare
    c. Program Code Evaluation
    d. Program Code Matching

12. During change control, the process of taking out one or more copies of program(s) from the production directory and moving them back to the production directory are important steps that require _____.
    a. Individual Controls
    b. Additional Controls
    c. Added Controls
    d. Standard Controls

13. CAE stands for _____.
    a. Component Aided Engineering
    b. Code Aided Engineering
    c. Computer Aided Engineering
    d. Control Aided Engineering

14. In the _____, it is easier to establish relationships between any two entities of a program, whether it is a database or an application program, where each entity is considered an object.
    a. Object Related Approach
    b. Object Oriented Approach
    c. Object Referenced Approach
    d. Class Oriented Approach

15. IDL stands for _____.
    a. Interface Definition Language
    b. Integrated Definition Language
    c. Interactive Definition Language
    d. Identity Definition Language

16. _____ Model aims at putting together a working model to test various aspects of a design, illustrate ideas or features and gather early user feedback.
    a. Prototype
    b. Spiral
    c. Prototype
    d. RAD

17. The attributes of a Web Based Application are _____.
    a. Network Intensive
    b. Content Driven
    c. Continuous Evolution
    d. All of the above

18. In the case of Web Based Applications, the client's database resides on _____ processor and the software and the data related to the frequently asked information may reside on a _____.
    e. Front end, Back End
    f. Back End, Front End
    g. Back End, Server
    h. Server, Back End

19. WSDL stands for _____.
    a. Web Server Description Language
    b. Web Standard Description Language
    c. Web Software Description Language
    d. Web Service Description Language

20. _____ was designed to describe data and to focus on what data is whereas _____ was designed to display data and to focus on how data looks.
    a. XML, XHTML
    b. XHTML, XML
    c. XHTML, HTML
    d. XML, HTML

21. CCA stands for _____.
    a. Change Control Authority
    b. Change Conduct Authority
    c. Change Control Application
    d. Change Conduct Application

Answers:-

1. (d)	2. (b)	3. (b)	4. (c)	5. (a)	6. (a)
7. (c)	8. (a)	9. (D)	10. (a)	11. (a)	12. (b)
13. (c)	14. (b)	15. (a)	16. (c)	17. (d)	18. (b)
19. (d)	20. (d)	21. (a)			

4 Project Management Tools and Techniques

Learning Objectives

To provide a clear understanding of:

- What is meant by Project Management in the context of IT Projects?
- Software Size Estimation Techniques: The significance of budgets and schedules in system development
- PERT (Program Evaluation Review Technique) as a project management tool
- Various kinds of tools and techniques for Project Management, such as Critical Path Method (CPM), Time Box Management, etc
- Computer Aided Software Engineering: CASE

Introduction

By now, we know that software is developed with a Structured Methodology and consists of various phases. We also know that these phases can be undertaken by using various models. Moreover, software is designed, programmed, used and managed with the help of various kinds of hardware and software. Software development is a complex process of managing resources like people, machines, etc. Researchers think that engineering principles and practices can be applied to software development process. Therefore, software development is treated as a Software Development Project and all Project Management tools and techniques are applicable to it.

These tools and techniques are:.

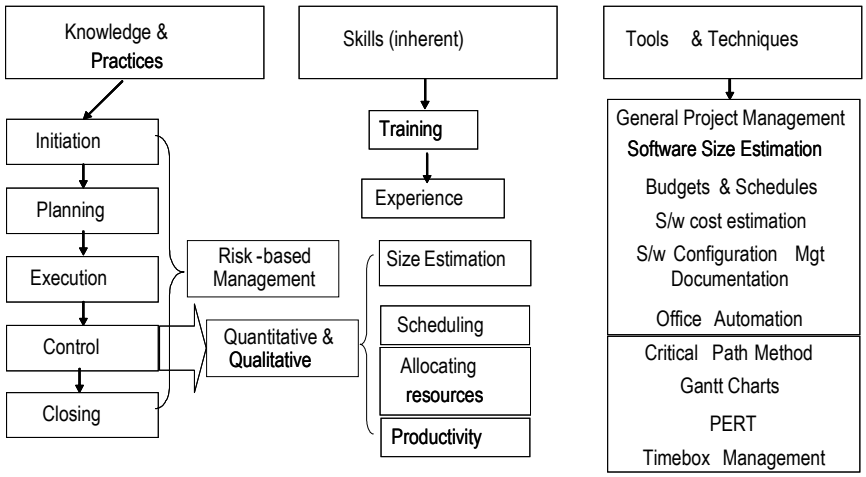


Fig 4.1 Various Tools & Techniques of Project Management

The diagram indicates that project management is the application of knowledge and practices followed in industry, skills of people, and various tools and techniques. The skills of people may be inherent but can be enhanced through proper training. Project Management involves the following:-

- Project Initiation
- Project Planning & Design
- Project Execution
- Project Monitoring & Control
- Project Closing

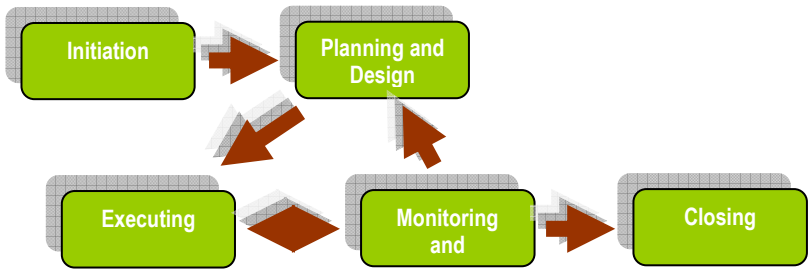


Fig 4.2 Development Phases of Project Management

Project Initiation: The initiation stage determines the nature and scope of the development project. Business needs are studied and analyzed as a measurable goal, and current operations are reviewed. The conceptual design of the operation

of the final product is created. Equipment needed during the project is identified. Financial analysis of costs and benefits and the required budget is carried out. Analysis of stakeholders including users, and support personnel for the project is done. Project charter, which includes, costs, tasks, deliverables, and schedule are prepared. If the initiation is not done correctly, the chances of success of the project are very low.

Project Planning & Design: Project planning relates to the use of schedules to plan and report progress within the project environment. Firstly, the project scope is defined and the appropriate methods for completing it are determined. Further, the duration for the required tasks necessary are listed and grouped in a work breakdown structure. The logical dependencies between tasks are defined by using an activity network diagram that identifies the critical path. Float or slack time (slack time for an activity is the time between an earliest and latest start time or between an earliest and latest finish time) in the schedule is calculated by using project management software. Then the necessary resources are estimated and cost of each activity allocated to each resource gives the total project cost. At this stage, the project plan may be optimized to achieve the appropriate balance between resource usage and project duration to comply with the project objectives. In the project design, the system as a whole is designed by creating a small prototype of the final product. This is tested by a combination of testers. The product is also designed in such a way that it satisfies the project sponsor, end user, and business requirements. Further, it should function as it was intended and can be produced within quality standards, time and budget constraints.

Project Execution: Project execution process involves coordinating people and resources as well as integrating and performing the activities of the project in accordance with the project management plan. The deliverables are produced as outputs from the processes performed as defined in the project management plan.

Project Monitoring and Control: Monitoring and Controlling consists of the processes performed to observe project execution so that potential problems are identified in a timely manner and corrective action taken, when necessary, to control the execution of the project. Monitoring and Controlling includes measuring the ongoing project activities, monitoring the project variables such as cost, effort, etc. against the project management plan and the project performance baseline , identifying corrective actions to properly address issues and risks and influencing the factors that circumvent integrated change control, so that only approved changes are implemented.

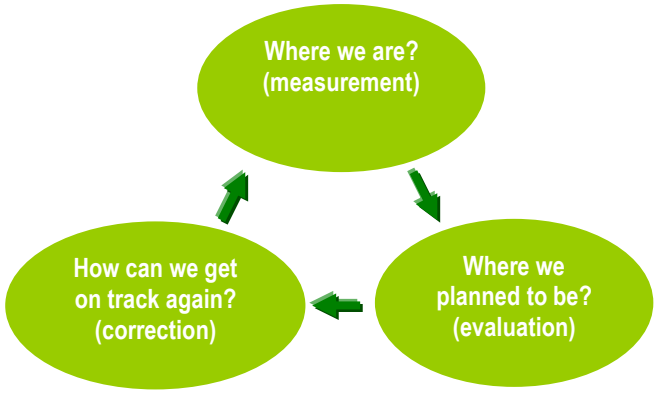


Fig 4.3 Monitoring and Controlling Cycle

Project Closure: Closing includes the formal acceptance of the project and its ending Administrative activities include the archiving of files and documenting lessons learned.

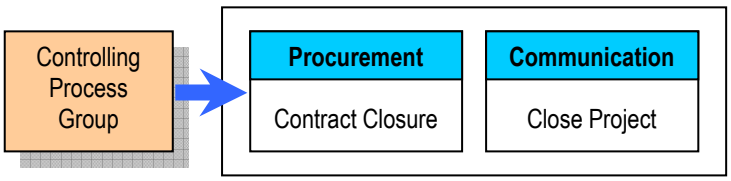


Fig 4.4 The Closing Process

Closing phase consists of two parts:

- Close project: to finalize all activities across all of the process groups to formally close the project or a project phase
- Contract closure: necessary for completing and settling each contract, including the resolution of any open items, and closing each contract applicable to the project or a project phase.

In order to control a project, the work product of each activity of a project is to be measured. For this, the qualitative and quantitative tools used are size estimation (of software), scheduling of resources, allocating resources to various activities under the project and measuring productivity of resources. Other kinds of software used are PERT (Preview Evaluation Review Techniques), CPM (Critical Path Method), and TBM (Time Box Management).

Software projects, as we know by now, also involve risks. Therefore for project management of software projects, a risk-based approach is adopted. The project

manager monitors a project through various activities and keeps on taking corrective actions wherever and whenever necessary. Thus controlling a project is a risk-based approach iterative in nature.

Let us see these aspects in more detail.

Any software development (or even software acquisition) is treated as a project. In a company, many such projects may be undertaken simultaneously or one after the other. So, a company will have a portfolio of Projects, which requires their management.

As soon as a project is initiated, the next activity is planning for the project. Now, similar to systems, here each project can be considered as a collection of sub-projects and within each project there will be activities (processes) to be undertaken. For example, if a company decides to purchase Sales & Distribution system, a project will be initiated and under this project feasibility study, requirements gathering, preparing RFP, etc will be the project activities. At the same time, in the same company, personnel department may be interested in buying Payroll & Personnel Management system. So, this project will also be initiated and will go through the phases of SDLC.

We have already learnt about feasibility study, requirements analysis etc. So now, we will concentrate on other aspects of project management, such as budgets and scheduling, CPM, PERT, etc.

Budgets and Schedules

Two critical problems in software development are time and cost overruns, and they need to be addressed by the project manager. [ASG2]These problems arise because of poor estimation of effort required and cost involved in developing an application. Budgeting involves estimating human and machine / software efforts in each task. Machine efforts refer to any piece of hardware required to develop a system. For example, in a project where a software requires a special card to be inserted in a machine or attached to a hardware port (this is called hardware lock without which the software will not start), the cost of this special hardware lock has also to be taken into account.

Now this gross person-month effort has to consider details, such as

- What are the activities in the project? E.g., requirements analysis, programming, data entry of masters, etc.
- In which sequence will these activities be performed? Activities can be performed serially one after the other or simultaneously (in parallel). E.g., a

program module once completed can be taken up for testing while other programming is going on.

- How will the total person-month effort be distributed over these activities, i.e., how much time will each activity take and how many persons of which type of skills will be required for each of these activities? Some of the persons required for a software development project could be Systems Analysts, Programmers, DBAs, Testers, Documentation people, etc.
- When will each activity start and finish?
- What additional resources are required to complete the activity? Does the activity require any special software or equipment?
- What will assess the completion of an activity? E.g., completion of hardware installation will be indicated by the client's sign-off on installation report given by hardware vendor at various locations.
- On what points will the management review the project? (In project management terminology, these are called milestones.) For example, unless hardware installation and operating system installation are over, application software cannot be installed. So, hardware and OS installation will be treated as a milestone.

Software Size Estimation

In order to arrive at the cost of software, it is necessary to determine its size. In early days, when procedural programming was used (mostly COBOL [Common Business Oriented Language]), the count of number of lines of source code (SLOC – Source Lines of Code) was the size of the software. However, this method did not work well with complex programs and newer techniques of programming. Therefore, function point analysis was developed by researchers. A function point represents the size and complexity of the application and this is computed on the basis of number of inputs, outputs, files, queries and interfaces that the application is likely to have. This estimate is arrived at in terms of person-months required to develop an application. Function point is then calculated based on reliability, criticality, complexity and reusability expected from the system. Thus Productivity = FP / Person-Month, Quality = Defects / FP, Cost = Rupees / FP.

Apart from software size estimation, some other components of cost to be taken into consideration for other phases of the project are:

1. Main storage constraints
2. Data storage constraints
3. Execution time constraints
4. Staff experience
5. Computer access

6. Security environment
7. Source code language
8. Target machine used for development

Gantt Charts

Gantt Charts are prepared to schedule the tasks involved in the software development. They show when tasks begin and end, what tasks are undertaken concurrently, and what tasks have to be done serially. They help to identify the consequences of early and late completion of tasks. The following is a sample Gantt Chart:

ID	Task Name	Duration	Start	Finish
1	Project Inititation	7 days	Mon 7/23/07	Tue 7/31/07
2	Finalisation of Project	2 days	Wed 8/1/07	Thu 8/2/07
3	Feasibility Study	15 days	Fri 8/3/07	Thu 8/23/07
4	Acceptance of FS	7 days	Fri 8/24/07	Mon 9/3/07
5	SRS	30 days	Tue 9/4/07	Mon 10/15/07
6	Acceptance of SRS	7 days	Tue 10/16/07	Wed 10/24/07
7	Programming	60 days	Thu 10/25/07	Wed 1/16/08
8	Testing	70 days	Mon 11/5/07	Fri 2/8/08
9	Implementation	7 days	Mon 2/4/08	Tue 2/12/08
10	Go Live	2 days	Wed 2/13/08	Thu 2/14/08

Fig 4.5 Schedule

Activity	Months 2007-2008							
	July	Aug	Sept	Oct	Nov	Dec	Jan	Feb
Project Inititation								
Finalisation of Project								
Feasibility Study								
Acceptance of FS								
SRS								
Acceptance of SRS								
Programming								
Testing								
Implementation								
Go Live								

Fig 4.6 Gantt Chart for the above schedule

In the above diagram, activities like project initiation, finalization of project, feasibility study are serial activities. Activities like programming and testing are parallel activities.

PERT : Program Evaluation Review Techniques

PERT represents activities in a project as a network. It indicates the sequential and parallel relationship between activities.

PERT Terminology

Activity

An activity is a portion of the project that requires resources and time to complete. The activity is represented by an arrow. Fig 4.7 shows activity A to activity H.

Event

An event is the starting or end point of an activity. It does not consume resources or time. It is represented by the starting circle in fig 4.7.

Predecessor activity

Activities that must be completed before another activity can begin are called predecessor activities for that activity. In fig 4.7, Activity A is the Predecessor Activity for Activity E.

Successor activity

Activities that are carried out after an activity is completed are known as successor activities. In fig 4.7, Activity E is the successor activity of Activity A while Activity G is the successor activity for Activities F, C and D

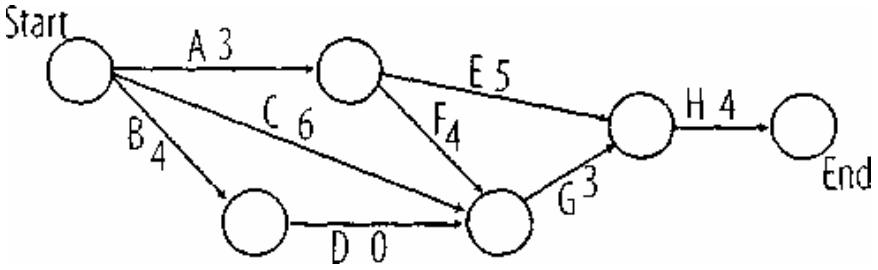


Fig 4.6 Showing Various Activities

Slack

Slack is the difference between earliest and latest completion time of an activity.

Maximum Total duration of this project = 14 days

While the earliest, B, can start on the 0th day, the project duration will not be affected even if it is started after one day. So the latest start day for D is 1. Hence Slack = 1-0 = 1. Similarly, for D, earliest start day = 4, latest start day = 5 and slack is 1.

Dummy

Dummy activity is an activity which requires no resources. In fig 4.7, activity D is a dummy activity. It does not have any real life significance. These activities are required in PERT, because as per its rules, not more than one activity can have the same preceding and succeeding activity. .

Time Estimate

PERT recognizes that estimates may not be precise; therefore, it allows a weighted average of different estimates such as pessimistic, optimistic and most likely. A heavier weightage is given to the most likely estimate and the calculation is as follows:

to - optimistic estimate

tp - pessimistic estimate

tm - most likely estimate

Expected time = (to + 4tm + tp) / 6

CPM (Critical Path Method)

In a network, critical path represents the path with the highest duration of time. For example, in the PERT network shown in fig 4.7, the following are the various paths from Start to End with their durations.

Path I : A – E - H = 12days

Path 2 : A – F – G – H = 14 days

Path 3 : C - G - H = 13 days

Path 4 : B – D – G – H = 11 days

Critical Path is Path 2 which has highest duration of 14 days. **The critical path assumes importance in project management**, because it is the shortest time in

which the project can be completed. Maximum control is required on the completion of any activity on critical path because if any activity on this path gets delayed, the whole project will be delayed.

Activities in the critical path have a zero slack.

The critical path is found by working forward through the network, computing the earliest possible completion time for each activity, and thus earliest possible completion time for the project. Taking this as the completion time of the project, working backwards the latest completion time of each activity is found. The path on which activities have the same earliest and latest completion time is the critical path, where the slack is zero.

For effective monitoring of the project, activities in the critical path have to be closely monitored. When there is contention for a resource between activities, critical path activities should be given preference. If the duration of the project has to be reduced, then it should be seen how activities in the critical path can be crashed, that is, how their duration can be reduced.

Time Box Management

A time box is a limited time period within which a well-defined deliverable must be produced with given resources. The deadline to produce the deliverable is fixed and cannot be changed. The difference between a time box and classical progress control is that when a time box is used, the scope of the deliverable is one of the variables of project management. The quality, however, is never a variable.

The project manager must continuously weigh the trade-off between the scope and quality of the deliverable and the time limit for accomplishing the work. If the scope of the deliverable cannot be further reduced and/or its quality is not high enough, the time box cannot be met.

Time Box Management Requirements

The following requirements must be met to conduct a project within a time box:

- A well-defined project plan that includes deliverable, resources, scope of deliverable and flexibility of that scope.
- A project manager who has the authority to prevent an increase in requirements, system specification or ongoing, unproductive discussion during the project.
- Procedures to limit the amount of time taken for decision-making, such as a problems and issues list for items that cannot be solved in a predefined period.

Use of Time Box Management

Time box management is useful for RAD (Rapid Application Development) type of SDLC (System Development Life Cycle) model. In such projects, a time box can be used to limit the time available for producing a working system.

A large system may be broken into subsystems, each of which can be built within a time box, either in a parallel manner or in a sequence. The ability to subdivide the development depends upon the system structure and whether each subsystem can be implemented independently or all subsystems have to be implemented together. The development of subsystems must be coordinated with the overall model and stored in the repository.

Project Management Methodologies

There are many formal project management methodologies that combine a framework or approach with a set of project tools and guidelines. Some are 'proprietary' approaches developed by consulting firms and software houses whilst others are in the public domain. They vary in scale and complexity, but all are based around a small core of common sense principles. Some of them are listed below:

1. PRINCE2 (Projects IN Controlled Environments) is a UK based model which focuses on Product Breakdown Structure and Work Breakdown Structure.
2. PRIDE is a project management system designed specifically to help those involved in projects that are part of the European Union's Fifth Framework programme. While providing project management resources, it focuses on good communication and team working, providing a tool for project monitoring and guidance on goal-oriented product planning.
3. The scalable methodology of the US Project Management Institute introduces the key principles of project management and provides guidance on how to fit the various tools and techniques available to a particular project.
4. Logical Framework (LogFRAME) methodology was originally developed by the United States Department of Defense and is an analytical tool to plan, monitor and evaluate projects.
5. MITP - Managing the Implementation for the Total Project - is an IBM project management methodology
6. Some researchers have also classified RAD, agile software development, waterfall, and SDLC methodologies as project management methodologies.

Product and Work Breakdown Structure

The idea of breaking down a project into products and work structures is similar to the concept of system and sub-systems. In product breakdown structure, the main concept is similar to what is applied in the engineering industry. A similar concept works in component based software development.

In product breakdown, the entire project is treated as a product and further sub-divided into sub products, and is called the PBS (Product Breakdown Structure). It depicts a hierarchical tree-like structure of components that make up an item, arranged in a whole-part relationship.

A PBS can help clarify what is to be delivered by a project and build a WBS (Work Breakdown Structure).

The PRINCE2, is product based planning, part of which is developing a product breakdown structure. E.g., in a banking system, products can be deposits and loans. Deposit, as a product, can further be divided into time and demand deposits and so on. Finally, we can reach a product breakdown structure which is sufficient to the context of the project. For example, PC -> Main Unit -> Motherboard -> CPU.

Work Breakdown Structure is the tree structure of work that is carried out to arrive at the product. PBS and WBS can be mapped one-to-one and can be represented as a network or tree structure.

Project Risk Management

Project Risk Management involves conducting risk management planning, engaging in risk identification, completing risk analysis, creating a risk response action plan, and monitoring and controlling risk on a project. It is a continuous process which is on during the course of the project. A key point to remember is that risk is not always bad. There are opportunities and there are threats. The opportunities are good risks; and the threats are bad risks. The purpose of project risk management is to increase the likelihood and impact of positive events and to decrease the probability and impact of negative events. The six risk management processes are:

1. **Risk Management Planning:** It is the process where decisions are made on how to approach, plan, and execute risk management activities. This is completed as a part of the planning process group.
2. **Risk Identification:** It determines the risk which can affect the project's objectives and identifies the characteristics of those risks. Risk Identification is commonly taken up by the planning process group.

3. **Qualitative Risk Analysis:** It prioritizes risk for future analysis by analyzing the probability of occurrence and impact. Qualitative Risk Analysis is commonly taken up within the planning process group.
4. **Quantitative Risk Analysis:** It assigns a number to risks as a part of determining the impact on overall project objectives. Quantitative Risk Analysis is commonly taken up by the planning process group.
5. **Risk Response Planning:** It ascertains the options and action plans to enhance opportunities and mitigate threats. Risk Response planning is normally first started in the Risk Response Planning Group.
6. **Risk Monitoring and Control:** It is an ongoing process and involves overseeing the effectiveness of risk responses, monitoring residual risks, identifying & documenting new risks, and assuring that risk management processes are followed. This is done throughout by the monitoring and controlling process group.

Each risk management process results in a specific deliverable which is used as a foundation for the subsequent process. The risk management processes provide the best practice pattern for managing risks involved in a project.

System Development Tools and Productivity Aids

Code Generators

Code generators generate program codes on the basis of parameters defined by a system analyst with data flow diagrams. These help in improving programmer efficiency. Such tools, which help in automation of software life cycle activities, are included in CASE (Computer Aided Software Engineering) tools.

Computer Aided Software Engineering (CASE)

CASE automates all activities associated with the software development life cycle.

Classification of CASE tools

CASE tools are generally divided into three categories, depending on the stage of the life cycle at which they can be used.

- **Upper CASE:** These tools are useful in the early stages of the life cycle. Tools that help in defining application requirements fall in this category.
- **Middle CASE:** These tools address the needs like those of design in the middle levels of SDLC. Tools that help in designing screen and report layouts and data and process design fall in this category.
- **Lower CASE:** The later parts of the life cycle make use of these tools. These use design information to generate program codes.

Integrated CASE Environments

While it is possible to use separate CASE tools for individual activities, an I CASE (Integrated CASE) tool can be used for better efficiency.

CASE Database (Repository)

- A CASE database or repository, as it is called, in an integrated CASE environment contains the following data:
- Enterprise information, such as organizational structure, business area analysis, etc.
- Application design information, such as data structures, menu trees, processing logic, etc.
- Construction / Programms information such as source code, object code, etc.
- Testing information such as test plan, test results, etc.
- Project management details such as project plan, work breakdown structure, estimates, schedules, etc.
- Documentation details such as systems requirements specifications, design document, and user manuals.

Advantages and limitations of CASE

Benefits of using CASE

- Since CASE strictly follows SDLC, its use enforces discipline in the steps of SDLC.
- The standardization / uniformity of processes can be achieved.
- Since CASE tools generate inputs of each stage from the outputs of previous stage, consistency of application quality can be ensured.
- Tasks such as diagramming, which are monotonous, need not be done by the programmer, and can be left to the CASE tool.
- This results in helping the programmer to do more productive tasks; thus development time can be shortened and cost economy can be achieved.
- Stage outputs and related documentation are created by the tool.

Disadvantages of CASE

- CASE tools are costly, particularly the ones that address the early stages of the life cycle.
- Use of CASE tools requires extensive training.

4th Generation Languages

Fourth generation languages are generally "application specific" languages. They refers to the non-procedural high level languages built around database systems. A lot of experience was gathered in certain areas, and it was leveraged to generalize by adding limited programming languages to them. This led to the creation of report-generator languages. A description of the data format and report has to be fed into the system. It has to be turned into a COBOL (or other language) program which actually contained the commands to read and process the data and place the results on the page.

Some other successful 4th-generation languages are: database query languages, e.g. SQL; Focus, Metafont, PostScript, RPG-II etc. xBase such as dBase and Foxbase / Foxpro have the capacity included in 4th generation languages. Microsoft Access is a PC platform which is similar to RDBMS having SQL languages and other features built into the platform.

Auditor's Role in Project Management

An IS auditor can participate in a project to monitor the project from control perspective, if the auditor's role is so defined in the project. An IS auditor may be asked to conduct a post-audit of a project after it is over. But the auditor's findings will hardly have any influence on the project outcome.

In general, an auditor has the following objectives for the project management activity:

- Projects have clearly defined objectives, and responsibilities and ownership of project products are clearly and sufficiently defined.
- Costs and benefits are clearly determined and properly monitored.
- Projects are completed successfully in line with the plan, and on time and within the budgetary limits.

The auditor will evaluate the project and carry out audit activities to get answers on the following questions:

1. Are project risks identified in the project and are they appropriate? Is there a risk mitigating plan? Are Project risks documented in Project Charter or Project initiation document?
2. Does the project provide for a sufficient budget, and does it have a project sponsor?
3. Does the project have a plan that is divided into PBS/WBS? Are the roles and responsibilities adequately defined and allocated / communicated to project personnel?

4. Does the project have a quality management activity as an important milestone?
5. Are the stakeholders and customers (internal and external) identified and is the "customer voice" planned in the project?
6. Does the project have a project office, organizational structure, and tools to monitor and manage it? Has the project manager been appointed?
7. Does the project require interaction with other projects running in a company? How has this interaction / intercommunication within projects been planned?
8. Does the project have a change management in place for project WIP and deliverables?
9. Does the project have defined templates / project directory structure on computers and other necessary paraphernalia?
10. How is the project being monitored for time and cost adherence? What corrective actions are being taken for lapses?

Based on such questions, the IS auditor will have to develop and conduct tests, interviews, and other verification mechanisms to form his opinion about the project under consideration. If the auditor has an independent role, he will submit his opinion formally to senior management and / or to project manager. He may opt to rank the project based on its performance. E.g., the ranking could be excellent, satisfactory, good or poor. The auditor may choose to use a scale of say 1 to 10 to rate the project. Evidence collection in the audit of a project management activity could be a tedious task as many times the evidence would be of a subjective kind.

Summary

The two important issues to be addressed by a project manager in the process of software development are time and cost overruns. As estimates are not very accurate, more time and money is required to complete a project.

A continuous effort to refine the effort estimation procedures has resulted in the development of many system development and project management tools. These enable the developer to represent the size and complexity of the application and evaluate the project cost / time estimate on the basis of number of inputs, outputs, files, queries and interfaces that the application is likely to have.

Some widely used approaches are Gantt Chart, PERT (Program Evaluation Review Technique) and CPM (Critical Path Method).

Gantt Chart shows when the tasks begin and end, what tasks can be undertaken concurrently, and what tasks must proceed serially.

PERT represents the sequential and parallel relationship between activities. It realizes that estimates cannot be spot on, and hence allows a weighted average of different estimates such as pessimistic, optimistic and most likely. A heavier weightage is given to the most likely estimate.

Another widely used approach for estimating time in the process of system development is CPM (Critical Path Method). In a network, the critical path represents the path which has the highest duration of time among different paths. This assumes importance in project management, because it is the shortest time in which the project can be completed.

In an attempt to automate all activities associated with software development, Computer

Aided Software Engineering (CASE) came into being. Some of the tools employed in CASE are project planning tools, prototyping tools, requirement tracing tools, web application development tools, testing tools, documentation tools and metric tools.

Questions:-

1. Project Management cycle involves:-
   a. Project Initiation -> Project Planning -> Project Execution -> Project Control -> Project Closing
   b. Project Initiation -> Project Execution -> Project Planning -> Project Control -> Project Closing
   c. Project Planning -> Project Initiation -> Project Execution -> Project Control -> Project Closing
   d. Project Initiation -> Project Planning -> Project Control -> Project Execution -> Project Closing

2. _____ are prepared to schedule the tasks involved in software development. They show when tasks should begin and end, what tasks can be undertaken concurrently, and what tasks can be done serially.
   a. PERT
   b. CPM
   c. TIME BOX MANAGEMENT
   d. GANTT CHART

3. PERT stands for _____.
   a. Preview Evaluation Review Technique
   b. Post Evaluation Review Technique
   c. Program Evaluation Review Technique
   d. Project Evaluation Review Technique

4. CPM stands for _____.
   a. Common Path Method
   b. Common Path Measure
   c. Critical Path Method
   d. Critical Path Measure

5. The deadline to produce the deliverable is _____ and _____ be changed.
   a. Variable, Can
   b. Variable, Cann't
   c. Fixed, Can
   d. Fixed, Cann't

6. Time box Management is useful for _____ type of SDLC model where a time box can be used to limit the time available for producing a working system.
   a. Prototype
   b. Spiral
   c. RAD
   d. Waterfall

7. _____ is the process where decisions are made on how to approach, plan, and execute risk management activities.
   a. Risk Identification
   b. Risk Management Planning
   c. Risk Response Planning
   d. Risk Monitoring and Control

8. _____ prioritizes risk for future analysis by analyzing the probability of occurrence and impact. Qualitative Risk Analysis is commonly first engaged within the planning process group.
   a. Qualitative Risk Analysis
   b. Quantitative Risk Analysis
   c. Cumulative Risk Analysis
   d. Calculative Risk Analysis

9. Code generators generate program codes on the basis of parameters defined by _____ or data flow diagrams which aid in improving programmer efficiency.
   a. System Programmer
   b. System Developer

c. System Manager
d. System Analyst

10. CASE stands for _____
    a. Computer Aided Software Engineering
    b. Computer Aided System Engineering
    c. Computer Aided Software Evaluation
    d. Computer Assisted Software Engineering

11. _____ tools address the needs of SDLC during design and help in designing screen and report layouts and data and process design.
    a. Upper CASE
    b. Middle CASE
    c. Lower CASE
    d. None of the above

12. Disadvantage of using CASE tools are _____.
    a. It is costly
    b. It requires extensive training
    c. Both (a) and (b)
    d. None of the above

13. Fourth generation languages are generally "_____" languages
    a. Data Specific
    b. Application Specific
    c. Object Specific
    d. System Specific

14. An IS auditor can participate in a project to monitor the project from_____, if the auditor's role is so defined in the project.
    a. Control Perspective
    b. System Perspective
    c. Audit Perspective
    d. Process Perspective

15. _____ is a PC platform which is similar to RDBMS having SQL languages and other features built into the platform.
    a. MS-Access
    b. MS-Excel
    c. MS-Word

d. MS-PowerPoint

16. _____ ascertains the options and action plans to enhance opportunities and mitigate threats.
    a. Risk Identification
    b. Risk Monitoring
    c. Risk Response Planning
    d. Risk Control

17. _____ involves overseeing the effectiveness of risk responses, monitoring residual risks, identifying and documenting new risks, and assuring that risk management processes are followed.
    a. Risk Identification
    b. Risk Monitoring & Control
    c. Risk Response Planning
    d. Risk Management Planning

18. The _____ is a UK based project management method, which mandates the use of product based planning.
    a. Prince2
    b. Pride
    c. LogFRAME
    d. None of the above

19. MITP - Managing the Implementation for the Total Project - this is _____ Project Management Methodology. (Reframe this question)
    a. Metasoft
    b. Infotronix
    c. Microsoft
    d. IBM

20. _____ Methodology was originally developed by the United States Department of Defense and is an analytical tool that is used to plan, monitor and evaluate projects.
    a. Logical Framework
    b. Physical Framework
    c. Network Framework
    d. System Framework

21. For effective monitoring of a project, activities in the critical path have to be_____.
    a. Loosely Monitored
    b. Closely Monitored
    c. Tightly Monitored
    d. Weekly Monitored

Answers:

1. (a)	2. (d)	3. (d)	4. (c)	5. (d)	6. (c)
7. (b)	8. (a)	9. (d)	10. (a)	11. (b)	12. (c)
13. (b)	14. (a)	15. (a)	16. (c)	17. (b)	18. (a)
19. (d)	20. (a)	21. (b)			

5 Specialised Systems

Learning Objectives

1. An understanding of AI (Artificial Intelligence) that includes:
    - Characteristic features of AI applications
    - AI applications like expert systems, neural systems, robotics, etc.
2. An insight into expert systems, their components, merits and shortcomings
3. An overview of data warehouse and data mining
4. An understanding of DSS (Decision Support Systems) that includes:
    - DSS frameworks
    - Design, development and implementation issues in DSS
    - DSS trends
5. PoS (Point of Sale) Systems
6. ATMs (Automatic Teller Machines)
7. EDI (Electronic Data Interchange), E-Commerce, and ERP (Enterprise Resource Planning) Systems

Artificial Intelligence (AI)

A computer is an electromechanical machine that contains no live elements. However, it is used for simulating human working which involves thinking and reasoning, solving simple and complex problems, calculating, etc.

Computer history shows that computers are good at making calculations of repetitive nature speedily. In fact, in the beginning, computers were used mainly for this purpose. Soon, after researchers evolved different techniques to make computers work like human beings.

Artificial Intelligence is an attempt to create intelligent human behaviour in a computer system on the basis of predetermined set of rules. But human beings are better than computers in several ways:

- Humans can think of things which they may not have experienced by improvising (e.g., story writing)
- They can use reasoning to solve problems

- They can learn from experience: That is, if a person burns his hands on a small fire, he will be careful next time.
- They can be creative and use their imagination to make paintings, songs, stories, films, etc.
- They can handle ambiguous or incomplete information. For example, in the word “informtion”, “a” is missing, but human beings can still understand it. AI tries to achieve the same through a computer.

AI Applications

The applications of AI can be classified into three major categories: Cognitive Science, Robotics and Natural Languages.

Cognitive Science: This is an area based on research in disciplines such as biology, neurology, psychology, mathematics and allied disciplines. It focuses on how human brain works and how humans think and learn. Applications of AI in the cognitive science are:

- **Expert Systems**: These are information systems with reasoning capability. These are discussed in detail later.
- **Learning Systems**: These systems modify their behaviour based on the information they acquire as they operate. Chess playing system is one such popular application.
- **Fuzzy Logic**: These systems can process ambiguous and incomplete data. This helps them to solve unstructured problems. These systems are 'trained' to learn imprecise terminology such as those normally used by humans in their interactions (e.g. cooler, faster etc). Many embedded systems as in washing machines, refrigerators, auto-focus cameras and energy efficient air-conditioners use fuzzy logic.
- **Neural Networks**: These are computing systems modeled after the human brain, and work through the mesh like network of interconnected processing elements. Though the architecture is much simpler than that of the human brain, it permits them to recognize patterns. These get more and more refined with data input. For example, credit risk determination could be handled by neural network systems. With data on creditworthy customers and defaulting customers, the network can generate patterns of users who are likely to be creditworthy or defaulters.
- **Intelligent Agents**: Intelligent agents are software that uses built-in and learned knowledge base about a person or process to make decisions and accomplish tasks in a way that fulfils the intentions of users. Most of the modern word processing packages have a facility to observe user operations, correct mistakes and provide useful hints on the problem at hand. This facility is an intelligent

agent. Wizards found in MS Office are intelligent agents. Wizards are built-in capabilities that can analyze how an end user uses a software package and offer suggestions on how to complete various tasks.

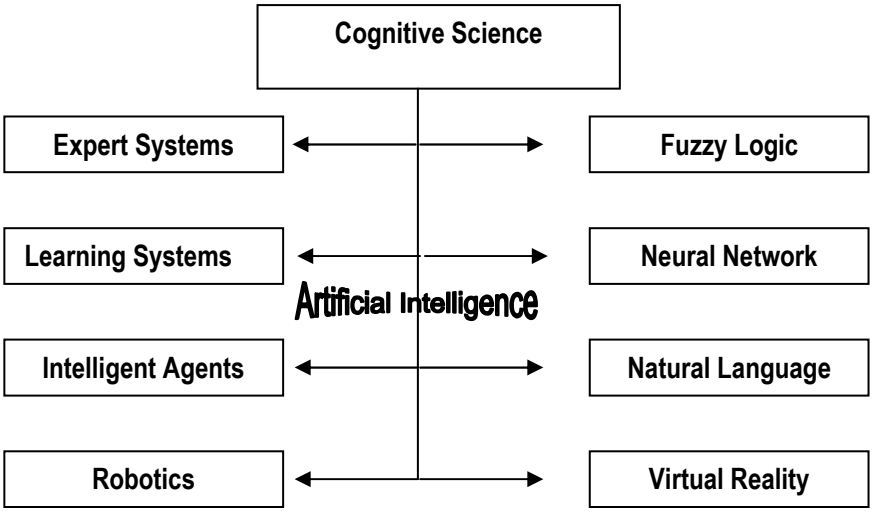


Fig 5.1 Branches of AI (Artificial Intelligence)

- **Robotics**: This technology produces robot machines with computer intelligence and human-like physical capabilities. This area includes applications that give robots visual perception, capabilities to feel by touch, dexterity and locomotion.
- **Natural Languages:** Being able to 'converse' with computers in human languages is the goal of research in this area. Interactive voice response and natural programming languages, closer to human conversation, are some of the applications. Virtual reality is another important application that can be classified under natural interfaces.
- **Virtual Reality**: Virtual reality involves using multi-sensory human-computer interfaces that enable humans to experience computer simulated objects, space and activities, as they actually exist. Flight simulation for training pilots, surgery imulation for training doctors are some of the applications of virtual reality.

Auditor's Role

IS auditor needs to be conversant with the controls relevant to these systems when used as the integral part of the organizations business process or critical functions and the level of experience or intelligence used as a basis for developing software. The errors produced by such systems would be more critical as compared to the errors produced by the traditional system.

Expert Systems

One of the most practical and widely implemented applications of artificial intelligence in business is the development of ES (Expert System). An expert system is a knowledge- based information system that uses its knowledge about a specific, complex application area to act as an expert consultant. They not only provide answers in the specific application area, but also explain their reasoning process and conclusions.

Components of Expert Systems

The components of expert systems are:

- **User Interface**: This allows the user to interact with the system. The user presents the problem for which solutions are delivered to him. There are different ways in which a dialogue takes place between the user and the system. The most common instance is the one in which the system runs a session where a set of questions are fired to the user. The user in turn has to reply with relevant information. Here all the initiative lies with the system because it runs the whole show. The ultimate interface is the one that allows the user to take the initiative. For example, consider the following questions to know what a patient is suffering from?

  System: What is the patient's name?
  User: Mithil Joshi
  System: What is the patient's age?
  User: 35 years
  System: Is the patient male or female?
  User: Male
  System: Does the patient have fever?
  User: Yes.
  System: Does he get fever with lot of shivering?
  User: Yes.

System: It seems that the patient Mithil Joshi is suffering from Malaria. He should be given Malaria treatment of chloroquine.

Interface Engine: The interface engine is that part of the program which reasons and determines the application of knowledge in the knowledge base to the facts presented in the user interface. Interface engine is the active component of an expert system and its main job is to mimic human reasoning so that the user can understand why the system has chosen a particular step. In the above example, the questions asked and the answers given are part of the Interface Engine.

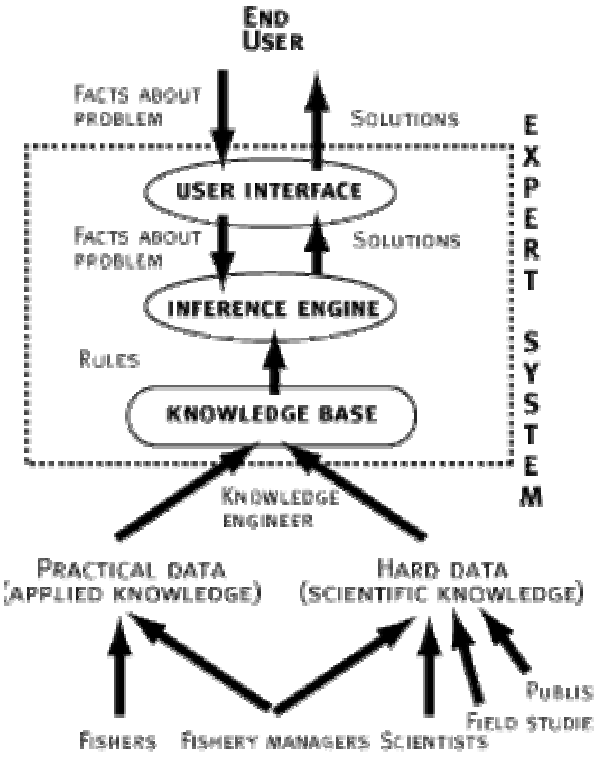


Fig 5.2 Components of the Expert System

Knowledge Base: This is the most important element of an expert system since it holds the problem solving knowledge. The key to this base is the way knowledge is represented. The knowledge acquired from the expert is represented formally. Knowledge representation deals with structuring of information and manipulating it to infer additional data. In the above example, based on the answers given by the system, it goes through its knowledge base (database) and finds out that if the patient has fever and also feels shivering, then he must be suffering from malaria.

Advantages of Expert Systems

- The knowledge and experience of the expert is captured before he leaves the organization.
- The codified knowledge in a central repository makes it easy to share it with the less experienced in the application area.
- This ensures consistent and quality decisions.
- It also enhances personnel productivity.

Limitations of Expert Systems

- Expert systems perform well in solving specific types of problems in a limited domain. When the problems involve multiple domains, it becomes difficult to construct expert systems.
- Because they do not have the capacity to learn, their knowledge is limited. . E.g., in the above example, a new disease (say Dengue) which may have similar symptoms like malaria may not be predicted unless some more questions are asked and the knowledge base is updated.
- Usage of specialized languages renders maintenance of expert systems difficult.
- Development costs of expert systems are high. This is because one may have to work with multiple experts to update the knowledge base.

Applications of Expert Systems

Some of the typical applications of expert systems are:

- Portfolio analysis
- Insurance
- Demographic forecasts
- Help Desk operations
- Medical diagnostics
- Maintenance scheduling
- Communication network planning
- Material selection

Applications of Expert Systems in IS Audit

The expert systems that can be used by the auditors fall in one of the following areas:

1. **Risk Analysis**: This system evaluates materiality and various types of risks associated with the audit assignment.
2. **Evaluation of Internal Control**: This system identifies the likely exposures in the given control area.
3. **Audit Program planning**: This system recommends a set of audit procedures to be conducted on the basis of subject characteristics and helps to evaluate the internal control system of an organization.
4. **Technical Advice**: This system assists in providing technical advice during audit. E.g., helping to confirm that financial statements conform to the statutory regulation.

Data Warehouse

Data Warehouse, as defined by W. H. Inmon, "is a Subject-oriented, integrated, time-variant, non-volatile, collection of data in support of management's decision making process."

Another definition given by Wayne Eckerson is that "It is a Central Repository of clean, consistent, integrated & summarized information, extracted from multiple operational systems, for on-line query processing."

The following diagram explains this concept:

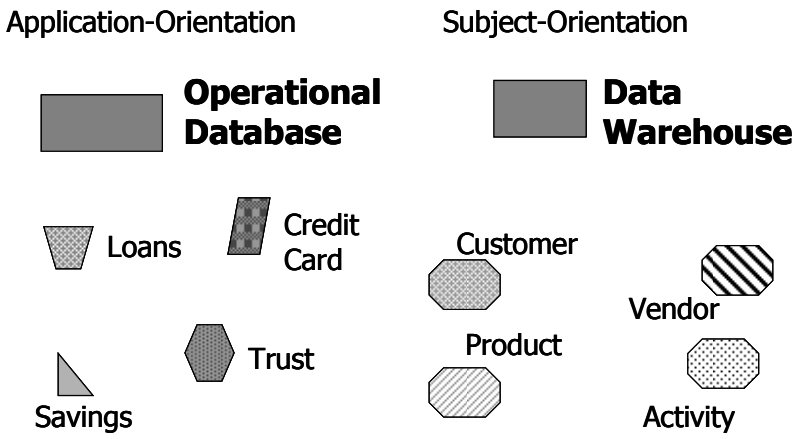


Fig 5.3 Creation of Data Warehouse

Both the definitions indicate that Data Warehousing system is used for getting valuable information for making management decisions. Generally, data is processed by TPS (Transaction Processing System), also known as operational systems. These systems are responsible for day-to-day functioning of business transactions. .Customers depositing and withdrawing money, applying for loans, opening accounts in a bank are examples of Transactions Processing System. The data associated with these transactions is stored in database management system and presented to users or programs whenever required. These systems process data either in a batch mode or in a real-time manner (e.g., when we book a railway seat we are presented with a ticket immediately). However, these systems have a limited use in management decisions. Consider the following simple query expected from a system.

"Which customer placed the biggest order for shoes during December?"

This query can be answered by writing a simple program or through a simple SQL query, on the operational database. However, consider the following query:

"What kinds of other items in other departments does a shoe purchaser buy on the same day?"

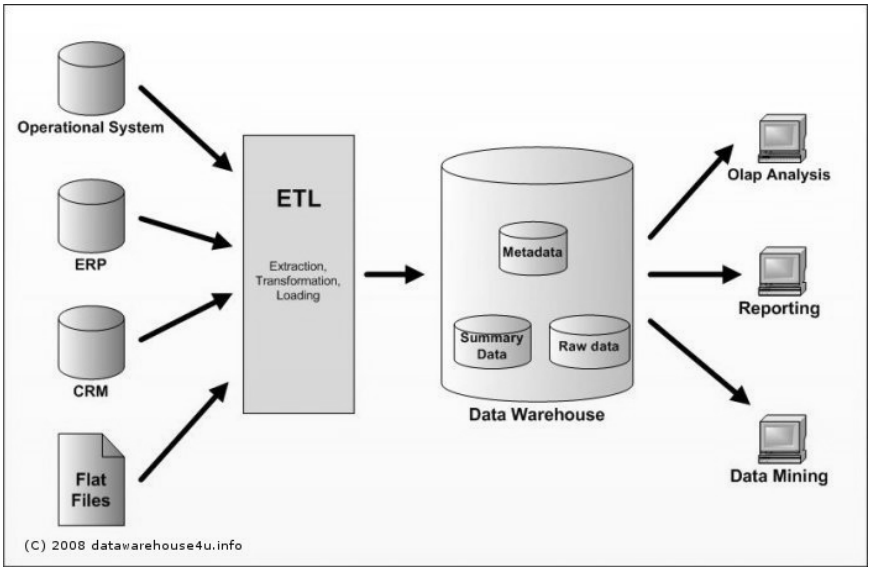


Fig 5.4 Architecture of Data Warehouse

This is a complex query as a program will have to be written (or SQL query) which fetches data from multiple tables, joins it and processes it. This is different from the way in which transaction processing system databases are organized. Therefore a separate organization of database and system is required to process this kind of DSS query. This is achieved through Data Warehousing.

Features of Data Warehouse

- It is a Stand-alone application.
- It has a repository of information which may be integrated from several, heterogeneous operational databases.
- It stores large volumes of data which are frequently used for DSS.
- It is physically stored separately from organization's databases.
- It is relatively static, and has infrequent updates.
- It is "Read-Only" application.

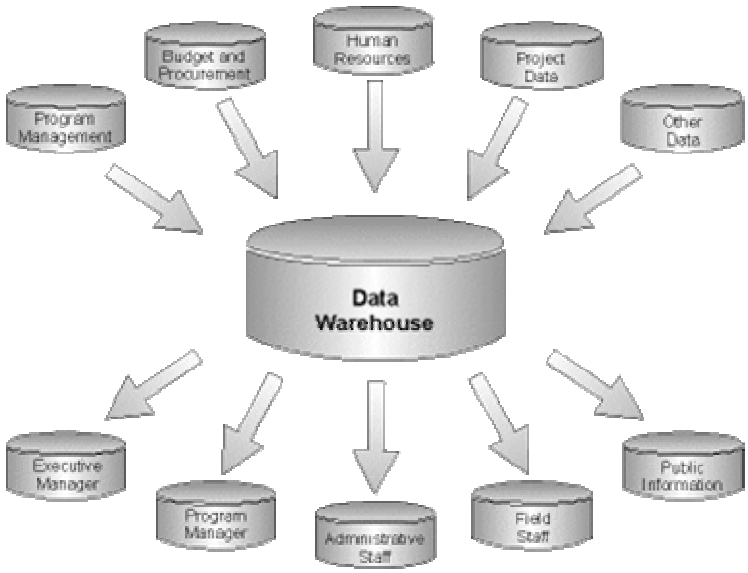


Fig 5.5 Various Domains of Data Warehouse

Preparation of Data Warehouse

Data is copied from ERP or other transaction processing systems and before uploading it in a data warehouse, it is aggregated, summarized & filtered for suitable analysis. End users run queries against this data to identify trends, patterns and correlations hidden in the data.

The following is the complete life cycle of a Data Warehouse:

1. Prepare data
2. Transform data
3. Load data
4. Model data
5. Establish access to data warehouse
6. Retrieve data
7. Analyse data
8. Archive data
9. Destroy data from data warehouse

Assume that in a transaction processing system in operation at various locations, the following is the case:

1. Users (may be at different locations) have entered values as “Dozen” or “Doz” or “Dz” or “12”. All these will have to be converted to a uniform value: that is Doz”

2. Non-standard data format e.g. Phone nos as 91-22-4308630 or 24308630 or 9819375598 or 98193-75598
3. Non-atomic data fields name as “Sachin Ramesh Tendulkar” instead of “Sachin” in first name, “Ramesh” in second name and “Tendulkar” in Surname

So, all these data items need cleaning and transformation. All the details presented here, which includes items 1 to 3, need to be restated clearly.

In contrast to the OLTP (Online Transaction Processing) applications of operational database, the data warehouse is subject to OLAP (Online Analytical Processing). OLAP permits many operations such as:

- **Consolidation**: Grouping and aggregation of data
- **Drill-down**: Looking into layers of details from summary, depending on the need.
- **Slicing and dicing**: Looking at the database from different viewpoints.

All these capabilities, coupled with the graphical output capability, support the top management in their strategic decision making.

Data warehouses find application in areas such as market analysis. The management attempts to identify patterns from the warehouse and make decisions on that basis. The process of identifying patterns in data warehouse is called data mining. When a data warehouse is not created for the entire enterprise, but only for a part of it for a specific function, it is called data mart.

Auditor's Role:

IS Auditor should keep in mind the following while auditing data warehouse:

1. Credibility of the source data
2. Accuracy of the source data
3. Complexity of the source data structure
4. Accuracy of extraction and transformation process
5. Access control rules
6. Network capacity for speedy access

Data Mining

Data Mining is a process of recognizing the patterns among the data in the data warehouse. IS Auditors are likely to place more reliance on data mining techniques to assess audit risk and to collect and evaluate audit risk by:

1. Detecting errors and irregularities
2. Knowledge discovery by better assessing safeguarding of assets, data integrity and effective and efficient operation of the system

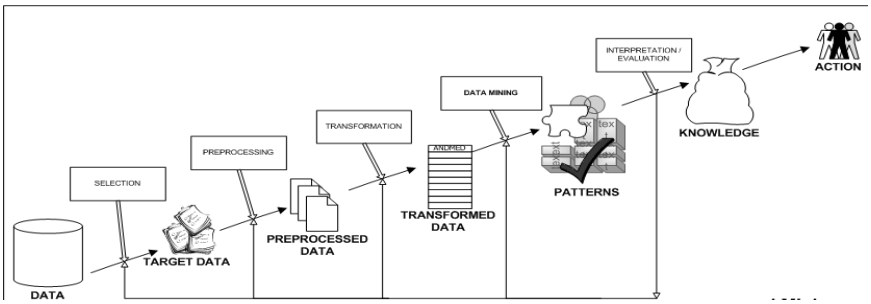


Fig 5.6 Data Mining

DSS (Decision Support System)

DSS (Decision Support System) are information systems that provide interactive information support to managers with analytical models. DSS are designed to be ad hoc systems for specific decisions by individual-managers. These systems answer queries that are not answered by the transactions processing systems.

Typical examples are:

1. Comparative sales figures between two consecutive months for different products with percentage variation to total sales.
2. Revenue and Cost projections on the basis of a product mix.
3. Evaluation of different alternatives, leading to the selection of the best one.

DSS Frameworks

Frameworks are generalizations about a field that help to put specific cases and ideas into perspective.

DSS framework will depend upon:

- The extent to which the problem can be structured, and
- The level of management that will use the model. This corresponds to the type of decision making involved: operational, control, or strategic.

MANAGEMENT LEVEL DIMENSION	DECISION STRUCTURE DIMENSION
1) Operational Control	1) Structured
2) Management Control	2) Semi structured
3) Strategic Planning	3) Unstructured

While a structured problem is amenable to automation, a highly un-structured problem involves greater complexity in unearthing the requirements. Similarly, as the

problem moves up the management decision-making hierarchy, the complexity of the model increases.

Design and Development

Prototyping is the most popular approach to DSS design and development. The prototype is iteratively refined on user feedback. Therefore this can be a cost effective and speedy method.

Effectiveness of the model will depend upon:

- Identification of all the related factors,
- The interdependencies, and
- The extent of their impact on the dependent variable.

Implementation and Use

The difficulty in implementing DSS is because of the subjective factors that are a part of the model. It is often looked upon as the product of an individual manager's subjective views and its implementation is resisted. It is here that the change management assumes importance.

Assessment and Evaluation

The true test of DSS lies in its capacity to improve the manager's decision-making, which is difficult to measure in tangible terms. It has to be recognised that DSS is evolutionary in nature. DSS systems do not have a defined completion date. Hence, it would not be advisable to evaluate DSS at the end, instead, DSS should be evaluated in the incremental steps for tangible results at each step.

DSS Trends

- With more sophisticated capabilities built into simple packages such as MS-Excel, the use of DSS is now within the reach of small and medium sized organizations.
- Such easy-to-use packages have increased the availability of design talent.
- The capability of PC based packages for graphical outputs have made the output formats user-friendly.
- With capabilities to incorporate expert systems in DSS, the quality of the models is significantly better.

PoS (Point of Sale System)

As the name indicates, a PoS (Point of Sales) system is intended to capture data at the time and place of transaction which is being initiated by a business user. It is often attached to scanners to read bar codes and magnetic cards for credit card

payment and electronic sales. They provide significant cost and time saving as compared to the manual methods. They also eliminate errors that are inherent in manual system (when a user is subjected to make transcription error while entering data from a document into system). POS may involve batch processing or online processing. These are generally observed in the case of big shopping malls or departmental shops.

In the case of batch processing, the IS auditor should evaluate the batch controls implemented by the organization, check if they are in operation, and review exceptional transaction logs. The internal control system should take care to ensure the accuracy and completeness of the transaction batch before updating it on the corporate database. In the case of online updating system, the IS auditor will have to evaluate the controls for accuracy and completeness of transactions; these controls should be more effective and exhaustive as compared to the controls in batch transfer.

ATM (Automatic Teller Machines)

An ATM (Automated Teller Machine) is a specialized form of the point of sale terminal. It is designed for unattended use by a customer of a financial institution. The ATMs generally allow cash deposits, cash withdrawals and a range of banking operations like requesting cheque books or account statements. ATMs are generally used for use after the closing hours of the financial institution and can be located either adjacent to the location of the financial institution or at a distant place. The facility of ATM can be within a bank, across local banks and amongst the banks outside a region. ATMs transfer information and money over communication lines. These systems provide a high level of logical and physical security for both the customer and the ATM machine.

Auditor's Role

The following are the guidelines for internal controls of ATM system which the auditor shall have to evaluate and report:

a. Only authorized individuals have been granted access to the system.
b. The exception reports show all attempts to exceed the limits and reports are reviewed by the management.
c. The bank has ATM liability coverage for onsite and offsite machines
d. Controls on proper storage of unused ATM cards, Controls on their issue only against valid application form from a customer, Control over custody of unissued ATM cards, Return of old/ unclaimed ATM cards, Control over activation of PINs
e. Controls on unused PINs, Procedure for issue of PINs, Return of PINs of returned ATM cards.

f. Controls to ensure that PINs do not appear in printed form with the customer's account number.
g. Access control over retrieval or display of PINs via terminals
h. Mail cards to customers in envelops with a return address that do not identify the Bank. Mail cards and PINs separately with sufficient period of time (usually three days) between mailings.

EDI (Electronic Data Interchange), E-Commerce, ERP (Enterprise Resource Planning) Systems

EDI (Electronic Data Interchange) Systems

EDI (Electronic Data Interchange) is the oldest form of transmitting business transactions between business partners with dissimilar computer systems. EDI is used to transmit and exchange business documents like purchase orders, request for proposals, invoices and shipping notices in a standard machine readable format.

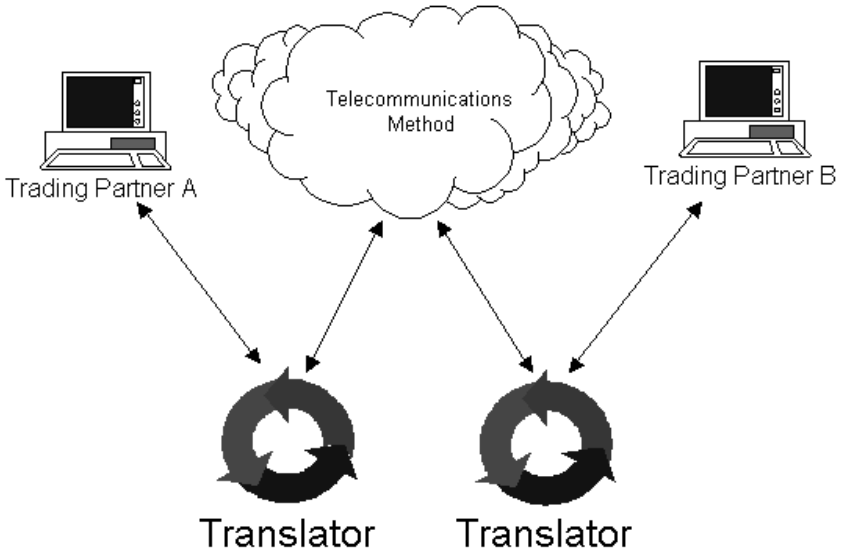


Fig 5.7 Electronic Data Interchange

The advantages of EDI are:

1. Reduction in paperwork
2. Improved flow of information
3. Less errors while transmitting / exchanging information
4. Speedy communication due to electronic transmission
5. Improvement in carrying out a business process

How does the EDI system function?

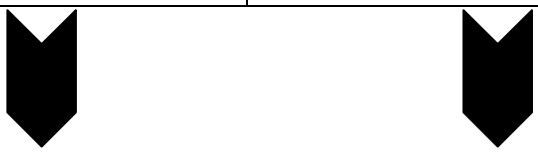
The EDI comprises the following three elements:

1. **Communication Software:** Communication software moves the data from one point to another point and marks the start and the end of the EDI transmission. It also decides how the acknowledgements are transmitted and reconciled.
2. **Translation Software**: Translation software involves conversion of data from a business application into a standard format, to be transmitted over the communication network, and converted back from the EDI format into the proprietary format of the receiver organization.
3. **EDI Standard**: EDI standard specifies standards for the transmission of business documents like invoices, purchase orders, etc.

Traditional EDI process generally involves three functions within each trading partner's computer system.

1. **Communication Handler:** Process for transmitting and receiving electronic documents between trading partners via Dial up lines, Public switched network, Multiple dedicated lines or a value-added network. VAN uses computerized message switching and storage capabilities to provide electronic mail box services similar to that of a post office. It receives all the outbound transactions from an organization, sorts them by destination and passes them to the recipients when they log to check their mail box. VAN may also perform translation and verification services.
2. **EDI Interface:** Interface function manipulates and routes data between the application system and the communications handler. EDI interface may generate and send functional acknowledgements, verify the identity of the partners and check the validity of transactions by checking the transmission information against the trading partner master file. The interface consists of two components:
    - **EDI Translator:** This device translates the data between the standard format and the trading partner's proprietary format.
    - **Applications Interface:** This interface moves the electronic transactions to or from the applications systems and performs data mapping.
3. **Application System**: The programs that process the data sent to, received from, the trading partner. E.g., Purchase orders from a purchasing system.

COMPANY A	COMPANY B
Business Application System	Business Application System
Internal Format Conversion	Internal Format Conversion
EDI Translator	EDI Translator
EDI envelope for document messaging	EDI envelope for document messaging



Value Added Network, Internet

The Layers of EDI transmission are:

EDI LAYERS	CONSTITUENTS	
EDI Semantic Layer	Application level services	
EDI Standard Layer	EDIFACT business form standards	
	ANSI X.12 business form standards	
EDI Transport Layer	Electronic Mail	X.435 MIME
	Point to Point	FTP, TELNET
	World Wide Web	HTTP
Physical Layer	Dial Up lines, Internet	

EDI Standards

There are two competing and mutually incompatible standards for EDI:

- ANSI ASCX.12 (American National Standards Institute-Accredited Standards Committee), and
- UN/EDIFACT (United Nations / Electronic Data Interchange for Administration Commerce and Trade) standards.

Features of ANSI ASCX.12:

1. This standard was developed by ANSI, and has been adopted in the USA and some Pacific Rim countries. Standards for 250 transactions are currently available.
2. It is relatively rigid and inflexible when compared to EDIFACT

Features of UN/ EDIFACT:

1. This standard was originally developed in Europe and adopted by United Nations.
2. This is more flexible than X.12
3. Flexibility leads to frequent versions. Different Companies may have different versions leading to conflicts.
4. Adopted in areas where X.12 was not adopted

Both the above standards are relatively expensive and have found acceptance in large organizations, but do not address the needs of small and medium size enterprises.

UN/XML

Now the effort is on to replace both X.12 and EDIFACT with XML (eXtensible Markup Language). XML messages can be run on any browser (like Microsoft Internet Explorer). This reduces the cost of EDI and benefits SMEs. These initiatives go by various names like UN/XML and ebXML (electronic business XML). The following are the features of this emerging world standard:

1. Flexible, with a simple format
2. Suitable for humans and applications
3. Allows workflow implementation
4. Fusion of UN/EDIFACT and national standard (X.12)
5. Allows multi-lingual usage
6. Suitable for SMEs as well as large enterprises

Web Based EDI

Web based EDI has become popular because of the following reasons:

1. Substantial reduction in the cost for small size organizations, which is because trade partners can use VPN on Internet as against dedicated communication lines.
2. Its ability to attract new partners via web based sites
3. Improvement in the traditional EDI format
4. New security products are available to address issues of confidentiality, integrity, authentication and non repudiation.

EDI Risks and Controls:

RISK OF EDI	EDI CONTROLS
Improper transaction authorization	Controls for inbound / outbound transactions
Integrity of EDI transactions	Encryption
Loss or duplication of EDI transactions	Use of sequence number for each transaction
Loss of confidentiality	Encryption
Non Repudiation	Digital signature

Auditor's Role in Auditing EDI

The IS auditor should ensure that all the controls related to the Inbound and Outbound EDI transactions are functioning. For this he has to ensure that:

1. Inbound EDI transactions are correctly received, translated and passed on to the correct application and are processed only once.
2. Outbound EDI transactions are authorized and sent to authenticated trade partners only.

To confirm the above, the auditor must:

a. Make sure that there is a control on functional acknowledgements. He has also to verify that the software matches the incoming functional acknowledgements against outbound transmissions.
b. Identify the controls for handling transaction sequencing. He also has to ensure that a unique transaction sequence is set for each trading partner and type of transaction, and verify that there are procedures to detect out of sequence transaction conditions.
c. Obtain a file of trading partners and verify that the software performs authorization check against the trading partner master file to ensure that the trading partner number and authorized transaction types are valid.
d. Verify that the software extracts data from the application or a flat file and reformats it in the EDI data formats in the version and release agreed upon by the Company and its trading partners. Also to ensure that the translator separates transactions and routes them to the appropriate application systems.
e. Check whether the maintenance agreement for the software includes updates to new versions and releases of the X.12 standard as part of the annual maintenance and obtain a copy of the agreement.
f. Ensure that the fields are checked for reasonableness and edit checks and to see that the data is accurate before it is routed to the appropriate application.

g. Ensure that all inbound transactions are logged. The total of the inbound transactions received by the EDI system must be equal to the number and value of transactions-forwarded and processed. Discrepancies, if any, must be accounted for and satisfactorily explained and reconciled.
h. Ensure the continuity of message serial numbers.
i. Review the functioning of the EDI system to ensure that functional acknowledgements are properly sent. The sender should match the functional acknowledgements with the log of the EDI messages sent.

e-Commerce (Electronic Commerce)

Other than buying and selling goods on the Internet, E Commerce (Electronic Commerce) involves information sharing, payment, fulfillment and service and support.

1. Information Sharing:

The WWW provides an effective medium for trading. Organizations can devise a Web Site to include product catalogs that can be searched electronically by prospective customers and can obtain data on which products were requested in searches, and how often these searches were made. A request can also be made to visitors to the website to provide information about themselves which will help the web site to provide information to the user based on the demographic data related to product searches.

2. Payment:

Payments can be made through the internet by using:

a. a credit card,
b. an electronic cheque, which has all the features of a paper check. It functions as a message to the sender's bank to transfer funds, and like a paper cheque, the message is given initially to the receiver who in turn endorses the cheque and presents it to the bank to obtain funds.
c. digital cash system in which the currency is nothing more than a string of digits. Banks issues these strings of digits and debit the account with the withdrawal equal to the value of the currency (tokens) issued. They validate each token with its digital stamp before transmission to the customer's account. When the customer has to spend some e-cash, he has to transmit the proper amount of tokens to the merchant who then relays them to the bank for verification and redemption. To ensure that each token is used only once, the bank records the serial number of each token as it is spent. The small denomination digital tokens used for payment are called microcash. Microcash can be used for items such as a stock quote, a weather report, an image or even a chapter from electronic book.

3. Fulfillment:

Many companies make money by generating, transferring or analyzing information. Documentation, program patches and upgrades are also well suited to Internet based distribution.

4. Service and Support:

Rarely does a company's relationship with a customer end with a sale. In fact, sale may be only the beginning of a long and fruitful relationship with a customer. Not only might the customer need some sort of assistance with the product or service, but the seller may electronically obtain the feedback, suggestion and registration of the product.

The Advantages of E Commerce are:

1. Saving in cost
2. Saving in transaction time
3. No limitations of geographical boundaries.
4. Large customer base for suppliers and large choices for customers
5. No restriction of timings
6. Reduction in Storage or holding cost
7. Different roles for the intermediaries

The E Commerce risks are risks connected with the transfer of any information through Internet.

RISK	CONTROL
1. Confidentiality of message Encryption with receiver's public key	Encryption with receiver's public key
2. Identity of the sender	Encryption by sender's private key (Digital Signature)
3. Integrity of the message	Comparison of the message digest sent by the sender with the message digest converted from the received message
4. Non Acceptance of confidentiality by receiver	Encryption with receiver's public key obtained by the sender from Certifying Authority
5. Non Repudiation by sender of having sent the message	Decryption with sender's public key obtained by the receiver from Certifying Authority

Types of E Commerce Models

1. B2B (Business to Business) Relationship:

B2B (Business to Business) sites link different businesses or parts of a business. Transactions on these sites take place between industrial manufacturers, wholesalers or retailers. These transactions are of high value per customer, lesser in number. They have a secured payment process and the information is kept private and confidential.

2. B2C (Business to Consumer) Relationship:

B2C (Business to Consumer) sites sell products or services directly to the consumers. Transactions on these websites are in large number having small value per customer. These sites enable the customers to set prices. Business enterprises bid to offer products and services to the customers.

3. B2E (Business to Employee) Relationship:

B2E (Business to Employee) sites are developed for employees of big enterprises. They give information from the business to employees and vice-versa.

4. B2G (Business to Government) Relationship:

B2G (Business to Government) site covers all transactions between government to business like making available information, circulars or forms applicable to the business. It involves the transactions like payment of taxes by the business to the government and refund of excess taxes by the government back to business.

5. C2C (Consumers to Consumers) Relationship:

C2C (Consumers to Consumers) sites enables the consumer to buy and sell through auctions or through other similar sites. For example, the vehicle of one consumer can be bought by another consumer.

6. C2G / G2C (Citizen to Government / Government to Citizen) Relationship:

C2G (Citizen to Government) and G2C (Government to Citizen) uses the internet for accessing government policies, tax forms, filing Electronic information, etc.

7. X2X (Exchange to Exchange) Relationship:

X2X (Exchange to Exchange) arrangement is multiple B2B relationship between two merchant exchanges. For example, the relationship of metal exchanges between two countries can be facilitated through such sites.

ERP (Enterprise Resource Planning)

ERP (Enterprise Resource Planning) Systems are fully integrated corporate solutions focusing on business applications like finance and control, production planning, sales,

warehousing and logistics. Presently, there are many ERPs available in the market like SAP, Oracle Applications, BAAN, People Soft, etc.

Fig 5.8 ERP (Enterprise Resource Planning)

As ERP systems are implemented by enterprises because of their acceptability, it requires lots of changes in the present system of documentation and steps. The IS auditor can evaluate the process by which the management worked out:

- their suitability and cost benefit analysis,
- the change management including conversion of data from the legacy system into ERP,
- the training to the users on the new ERP system.

The ERP systems save a lot of time by recording the business transaction only once. For example, the raw materials received shall be recorded only by the receiving department, while the dispatches shall be recorded by Warehouses on the basis of physical outward movement of the finished goods. Collections from the customers or debit or credit notes relating to sales shall be entered by Sales Department. And, the job of accounting and finance department shall be that of reconciliation and control. The information is accessible to anybody who has authorized access under rules.

Fig 5.9 Various Modules of ERP

Auditor's Role

1. The IS Auditor can refer to the implementation or customization procedure to ensure that appropriate controls available in ERP are enabled while customization.
2. As the information is recorded only through authorized persons, responsibility can be fixed for the transactions recorded and cancelled.
3. The Auditor is required to evaluate internal controls in the operation of the ERP as designed by the management and available in the respective documentation. With the help of interviews and observation he ensures that these are actually followed at the departmental level.
4. A lot of learning of ERP under consideration, especially, terminology and understanding of navigation through the system is required for conducting IS audit through the ERP system.

Summary

Specialized computer-based application systems or intelligent systems refer to a framework that integrates both hardware and software infrastructure in such a way that it serves as a complete solution to problems that entail human intelligence. In other words, these systems simulate the human brain in problem-solving methodologies. Continuing research and study in the field of specialized computer-based systems has resulted in the evolution of AI (Artificial Intelligence), ES (Expert System), and DSS (Decision Support System). AI is an attempt to simulate intelligent

behaviour in computer systems that makes a computer think and reason. Further AI will make use of this reasoning for solving problems.

An ES is a knowledge-based information system that uses its knowledge about a specific, complex application area to act as an expert consultant. It not only provides answers in the specific application area, but also explains the reasoning process and conclusions.

Typically, an expert system comprises user interface, interface engine and a knowledge base.

Expert systems are employed in portfolio analysis, insurance, demographic forecasts, help desk operations, medical diagnostics, maintenance scheduling, communication network planning and material selection.

DSS are information systems that provide interactive information support to managers with the use of analytical models. DSS are designed as ad hoc systems and modeled for specific decisions of individual managers. For example, a spreadsheet package can be used for creating DSS models.

Establishing and ensuring proper functioning of intelligent systems involves efficient data object management. This calls for a data management design that is capable of linking different interfaces (both software and hardware), and managing historical and statistical data. Such data management systems are called data warehouses.

Questions:

1. _____ is an attempt to duplicate intelligent behaviour of the human beings in computer system on the basis of predetermined set of rules.
   a. Artificial Intelligence
   b. Expert System
   c. Fuzzy Logic
   d. Intelligent System
2. _____ are the systems that can modify their behaviour based on information they acquire as they operate. Chess playing system is one such popular application.
   a. Expert System
   b. Fuzzy Logic
   c. Intelligent System
   d. Learning System
3. _____ are systems that can process data that are ambiguous and incomplete. This permits them to solve unstructured problems.
   a. Learning System

b. Fuzzy Logic
c. Neural Network
d. Intelligent Agents

4. Credit risk determination could be a good application for _____ systems.
   a. Fuzzy Logic
   b. Expert System
   c. Neural Network
   d. Learning System

5. _____(are?) software that use built-in and learned knowledge base about a person or process to make decisions and accomplish tasks in a way that fulfils the intentions of user.
   a. Intelligent Agent
   b. Fuzzy Logic
   c. Expert System
   d. Artificial Intelligence

6. Interactive voice response is an application of _____.
   a. Fuzzy Logic
   b. Expert System
   c. Natural Language
   d. Robotics

7. _____ involves using multi sensory human-computer interfaces that enable humans to experience computer simulated objects, space and activities, as they actually exist.
   a. Interactive voice response
   b. Expert System
   c. Virtual Reality
   d. Fuzzy Logic

8. In Expert System _____ is that part of the program which reasons and determines the application of knowledge in the knowledge base to the facts presented in the user interface.
   a. User Interface
   b. Interface Engine
   c. Knowledge Base
   d. None of the above

9. _____ application of expert system will recommend a set of audit procedures to be conducted on the basis of subject characteristics and may help to evaluate the internal control system of an organization.
   a. Risk Analysis
   b. Evaluation of Internal Control
   c. Audit Program Planning
   d. Technical Advice

10. _____ is a Subject - oriented, integrated, time-variant, non-volatile, collection of data in support of management's decision making process.
   a. Data Warehouse
   b. Data Mining
   c. Both (a) and (b)
   d. None of the above

11. _____ is the ability to look into layers of details from summary, depending on the need. (Reframe this question.)
   a. Slicing
   b. Dicing
   c. Drill-down
   d. Consolidation

12. DSS stands for _____.
   a. Decision Support System
   b. Decision Service Support
   c. Decision Support Service
   d. Decision Service System

13. _____ is a specialized form of the point of sale terminal.
   a. PIN
   b. PAN
   c. ATM
   d. None of the above

14. EDI stands for _____.
   a. Electronic Digital Data
   b. Electronic Data Definition
   c. Electronic Data Interface
   d. Electronic Data Interchange

15. _____ involves conversion of data from a business application translated into a standard format, to be transmitted over the communication network, and convert this data back from the EDI format into the proprietary format of the receiver organization. (Change. Badly worded and confusing)
    a. Communication Software
    b. Translation Software
    c. EDI Standards
    d. None of the above
16. Payments can be made through Internet by using
    a. Electronic Cheque
    b. Digital Cash
    c. Credit Card
    d. All of the above
17. In _____ form of e-commerce, transactions sites take place between industrial manufacturers, whole sellers or retailers.
    a. Business to Consumer
    b. Consumer to Business
    c. Business to Business
    d. Business to Government
18. E-commerce risk includes _____.
    a. Identity of the sender
    b. Integrity of the message
    c. Non Acceptance of confidentiality by receiver
    d. All of the above
19. In _____, currency is no more than a string of digits.
    a. Digital Cheque
    b. Credit Card
    c. E-Wallet
    d. Digital Cash
20. In _____, an organization can devise a WebSite to include the product catalogs that can be searched electronically by prospective customers and it can obtain data on products requested in searches.
    a. Information Transfer
    b. Information Sharing
    c. Information Processing
    d. None of the above

21. The errors produced by AI systems would be _____ as compared to the errors produced by (the traditional system of what?).
    a. Critical
    b. Negligible
    c. Proportionate
    d. None of the above

Answers:-

1. (a)	2. (d)	3. (b)	4. (c)	5. (a)	6. (c)
7. (c)	8. (b)	9. (c)	10. (a)	11. (c)	12. (a)
13. (c)	14. (d)	15. (b)	16. (d)	17. (c)	18. (d)
19. (d)	20. (b)	21. (a)			

6 Auditing the System Development Process

Learning Objectives

- To understand the role of IS auditors in System Development,
- To understand the IS auditor's role in reviewing development phases of SDLC,
- To discuss the role of IS auditors in Project Management, and
- To discuss various checklists for Systems Development.

IS Auditor's Role in System Development, Acquisition and Maintenance

The different development models discussed earlier make us wonder how the IS auditor will check the compliance of systems. For each system, the auditor elicits the information about the methodology used for development and assesses its suitability for the system. This can be done by interviewing the project management team. After he is convinced about the suitability of the process, he can check the compliance of the process with the existing system. This process generally includes the following:

- Identifying subsystems and modules, their goals and user functionality expectations. The auditor does not undertake this task of identification himself. He goes through the analyst's findings and crosschecks them with users. Identification of these details becomes necessary to suggest the required controls.
- Checking if the control recommendations are appropriate for the risks identified in the project. The auditor reviews the risk analysis done by the analyst, and verifies if all the risks have been covered through user interviews.
- Advising the design team on incorporating control measures. The user also needs to be convinced about the control measures that have to be in place, and this too is the auditor's responsibility.
- Verifying if the recommendations he has made are properly implemented. To achieve this objective, he has to monitor the design process.
- Apart from ensuring control aspects, the auditor also has to ensure that the systems help to meet the organizational objectives and hence auditor must know

correctly that 'what are the organization's objectives'. This can be achieved by discussing with the senior management before starting audit.

- Reviewing the system development methodology to ensure the quality of the deliverables. He has to check for adherence to stated methodology documentation and other deliverables.
- Reviewing the change management process and effectiveness of its implementation.
- In the post implementation phase, the delivery and project management team may cease to exist, so the auditor will have a greater role to play in assessing the effectiveness of the system.
- Reviewing the maintenance procedure and ensuring that adequate documentation has been maintained for related activities. Lax controls in this phase can also have an adverse effect on the systems, so the auditor should continue to monitor the maintenance procedures.
- Ensuring production source integrity during the maintenance phase.

IS Auditor's Role in Reviewing Developmental Phases of SDLC

The IS auditor also has to review all the phases of the system development life cycle, such as:

1. Feasibility study
2. System requirement definition
3. Software acquisition
4. Detailed design and programming
5. Testing
6. Implementation
7. Post-implementation and maintenance
8. System change procedures and program migration process

The responsibilities of the IS auditor in each of these phases is given below:

Feasibility study

While reviewing the feasibility study phase, the IS auditor has to consider the following questions:

- Has there been an agreement in the definition of the problem among all stakeholders?
- Was there a genuine need for solution established?
- Were alternate solutions considered? Or was the feasibility assessed on the basis of a single solution?
- What was the basis for choosing the solution?

- What is the extent of the problem perceived and how extensive is the impact of the solution likely to be? These give valuable inputs for evaluating feasibility.

Feasibility is incomplete without establishing clearly, the requirement and the technology proposed. In other words, the auditor has to ensure that the suggested technology is viable before implementing it in the development process.

In cost-benefit analysis, there could be a tendency to understate costs and overstate benefits. There could also be inaccuracies in identification and quantification of benefits. The auditor can provide a valuable input in evaluating the cost-benefit analysis.

Requirements definition

- The auditor should collect a copy of System Requirements Specifications (SRS) document and review it in terms of :
    - Problem definition
    - Information flows
- The auditor can also evaluate the methodology employed and the compliance level.
- The auditor should also check whether CASE (Computer Aided Software Engineering) tools were used, because the quality of work is likely to be better in CASE environments.

Software acquisition process

- The decision to acquire the software should flow from the feasibility study. The auditor should ensure that it is so.
- The auditor should also ensure that the software acquired would meet the overall design goals of the proposed system, identified during requirement analysis phase.
- RFP (Request for proposal) should be checked for adequacy. Details such as transaction volume, data base size, turn around time and response time requirements and vendor responsibilities should clearly be specified in RFP
- The auditor should also check the criteria for pre-qualification of vendors.
- Similarly, there should be sufficient documentation available to justify the selection of the final vendor / product.
- The auditor may also collect information through his own sources on vendor viability, support infrastructure, service record and the like.
- The auditor should thoroughly review the contract signed with the vendor for adequacy of safeguards and completeness. The contract should address the contingency plan in case of vendor failures such as, source code availability and

third party maintenance support He should also ensure that the contract went through legal scrutiny before it was signed.

Detailed design and programming phases

- CASE environment simplify the tasks that have to be done by the IS auditor. This is because the CASE tools ensure quality and consistency of design, a major concern of auditors. In non-CASE environments, the auditor may have to undertake a detailed design review:
- The design diagrams should be checked for compliance with standards
- Any change that has been incorporated in the design stage should have appropriate approvals and this should be checked.
- The auditor should check the design for modularity.
- The auditor should review the input, processing and output controls of systems.
- Design of input screens and output reports is an area, in which auditors can offer valuable suggestions. The auditor should check the user interface design for usability, appropriateness, compliance with standards and acceptance by users.
- Audit trails are important the auditor should ensure its availability and adequacy in the design
- In the recommendation of hardware and software choices, the auditor should look for compatibility, interoperability and scalability conditions.
- Flow charts and other such tools should be checked for key calculations. Their implementation in programs also should be checked with the help of a programmer who is knowledgeable about the programming language.
- Exception data handling is an area that the auditor has to focus on. He has to test the design and program for such data.
- Unit test results should be reviewed. The auditor has to ensure that the 'bugs' have been fixed.

Testing phase

- The auditor should review the test plans for completeness.
- Cyclical processing such as month-end reports should be verified.
- Security functions of the system also have to be verified.

Implementation phase

- The documentation on parallel run, if available, should be reviewed for effectiveness.
- Operating procedures should be checked for clarity and accuracy
- System and user documents should be checked for adequacy, clarity and currency.

- It should be ensured that data conversion has been completed and all past data are available in a format readable by the new software.

Post-implementation review

This review mainly addresses the issue of system's ability to fulfill objectives that were specified initially. Apart from this, the auditor has to check the following aspects:

- Compliance with change control procedure
- Functioning of controls in accordance with design
- Review of operator error logs

System change procedures and program migration process

On a periodic basis, the auditor should check the following:

- Procedures for authorizing, prioritizing and tracking system changes
- Appropriateness of authorizations for selected change requests
- Existence of program change history
- The match program and documentation versions
- Access control procedures on source and executable codes in production directory
- Procedure for emergency changes
- Security of emergency login ids.
- The match between current version of source code and executable code in production directory

IS Auditor's Role in Project Management?

Apart from the phases of SDLC, project management is also subject to audit - particularly with respect to recognition and management of risks during different phases. The risk management process includes the measures undertaken to mitigate the risks at costs commensurate with the level of risks. Not recognizing risks or providing exorbitantly a costly mitigation measure for trivial risks should be avoided and this is the main focus of IS audit of project management activities.

To achieve the objective, the IS auditor can undertake the following activities:

- Collect documentation of each phase and check for adequacy and completion.
- Attend project meetings to check the compliance of the development process.
- Advise the team on adequate and cost effective control measures.
- Represent the management interest in the team by continuously assessing the ability of the team to meet targets that have been set.

Systems Development Project - Audit Checklist

A generalized Systems Development Project - Audit Checklist as contributed by Judy Condon (Ref : JCONDON@bcbsm.com) is given below :

Corporate Policies and Practices

1. Does corporate policy establish in writing detailed practices to govern systems development projects?
2. Is there a corporate organization which clearly establishes responsibilities and authority for each component of the organization relating to systems development and data processing functions?
3. Is there a management steering committee which is assigned specific responsibilities for systems development projects?
4. Is there an appropriate systems development methodology which provides for periodic milestone events?
5. Does corporate policy require sign-off at milestones by appropriate executives before proceeding?
6. Is there a project management and control system which requires preparation of time and cost budgets and then measurement of actual vs. planned results?
7. Is there an independent quality assurance function which monitors in detail systems development projects?
8. Is a project manager assigned with overall responsibility for direction/ coordination of systems development?
9. Are there adequate standards for complex systems development projects and functions?
10. Do documentation standards provide detailed guidance for each step and each product during systems development?
11. Is there a corporate data security function which monitors systems development, maintenance and operation?
12. Is there a corporate data administration function and have detailed responsibilities and authority been established?
13. Is there a corporate data dictionary and is its use required during systems development and modification?
14. Are feasibility, impact, cost/benefit, and risk analysis required to be prepared, approved and maintained during systems development projects?
15. Are internal control and security features included with systems design?
16. Are systems acceptance criteria and acceptance test plans prepared during documentation of systems requirements and design and updated with each phase prior to sign-off?

17. Are there turn-over standards for use by computer operations prior to acceptance of developmental systems for production?
18. Is there review, approval and control of changes during development, maintenance and operation of computerized systems?
19. Is the systems design required to provide for off- site backup and recovery measures and are they required to be tested before the system is accepted for operation in a production mode?
20. Is the internal auditor required to monitor systems development projects, sign-off at milestones, and review and approve acceptance test results?

User Requirements

1. Are user requirements documented well and clearly?
2. Is the responsible user executive specified?
3. Have cognizant user executives approved the requirements?
4. Is a priority for implementation requested?
5. Is the project included in the long- or short-range systems plan?
6. Are the business objectives expressed clearly?
7. Is the scope of the project defined well?
8. Are benefits claimed supported?
9. Is a solution or solutions to business objectives proposed?
10. Does the requirements study include potential needs for the future?
11. Does the requirements study consider potential use in meeting common needs?
12. Are existing systems to be replaced or interfaced identified clearly?
13. Are existing systems to be replaced or interfaced documented adequately and accurately?
14. Have other departments which will be involved in systems development and operation been consulted during preparation of the requirements and have recommendations been included?
15. Do user requirements include security, control and privacy measures?
16. Do benefits claimed appear to be reasonable?
17. Do user requirements appear to reflect actual needs?

Feasibility Analysis

1. Is the feasibility analysis documented well and clearly?
2. Have departments which will be involved in systems development and operation been consulted during the feasibility analysis and have recommendations been included?
3. Does the feasibility analysis reflect any significant differences from original objectives, boundaries, and interfaces?

4. Is the preliminary design in sufficient detail to support adequately the time and cost estimates, cost/benefit analysis, and impact study?
5. Does the preliminary design meet user requirements?
6. Does the preliminary design reflect corporate standards?
7. Has a project plan been prepared?
8. Have preliminary time and cost estimates been prepared?
9. Has a preliminary impact study been prepared?
10. Are the conclusions and recommendations supported by the feasibility analysis?
11. Do the recommendations conform with corporate policies and practices?
12. Has the feasibility analysis report been submitted to the management steering committee for action?
13. Have responsible departments signed off for the feasibility phase?
14. Has the internal auditor prepared a milestone report with opinions and recommendations for the feasibility analysis phase?

Systems Design

1. Is the systems design documented well and clearly?
2. Have significant changes to the preliminary design-been controlled and approved by cognizant authority?
3. Has a detailed work plan been prepared for the design phase?
4. Has the systems development methodology been used effectively?
5. Has the project management and control system been used effectively?
6. Has actual accomplishment been reasonably close to estimates?
7. Are systems development team resources adequate to accomplish objectives?
8. Have time and cost estimates, cost/benefit analysis, and impact study been updated?
9. Have significant changes to project scope been approved by the management steering committee?
10. Do detailed functional design features reflect accurately approved detailed user requirements?
11. Is it reasonable to expect the designed system to be implemented satisfactorily within the user and data processing environments?
12. Does the design provide adequately for internal control and data security?
13. Does the design provide adequately for requested audit features?
14. Have the requirements for hardware and systems software been developed and can they be met satisfactorily with resources available or approved for installation?
15. Does the design provide adequately for corporate standards and practices?
16. Have systems design acceptance criteria been prepared?
17. Has a preliminary systems test plan been prepared?
18. Does the design provide adequately for offsite backup and recovery measures?

19. Has data administration reviewed the systems design?
20. Has data security reviewed the systems design?
21. Has quality assurance reviewed the systems design?
22. Has data processing operations reviewed the systems design?
23. Have cognizant user departments reviewed the systems design?
24. Has a risk analysis been conducted?
25. Is the input defined in detail?
26. Is the output defined in detail?
27. Is the functional logic defined in detail?
28. Is the logical file structure defined in detail?
29. Has the systems design been submitted to the management steering committee for action?
30. Have responsible departments signed off for the systems design?
31. Has the internal auditor prepared a milestone report with opinions and recommendations for the design phase?

Systems Specifications

1. Are systems specifications documented well and clearly?
2. Have significant changes to systems design been controlled and approved by cognizant authority?
3. Has a detailed work plan been prepared for the systems specifications phase?
4. Has the systems development methodology been used effectively during development of systems specifications?
5. Has the project management and control system been used effectively?
6. Has actual accomplishment during development of systems specifications been reasonably close to estimates?
7. Are systems development team resources adequate to accomplish objectives?
8. Have time and cost estimates, cost/benefit analysis, and impact study been updated?
9. Have significant changes to project scope been approved by the management steering committee?
10. Do systems specifications reflect accurately approved functional design features and user requirements?
11. Is it reasonable to expect the systems specifications to be implemented satisfactorily within user and data processing environments?
12. Do the systems specifications provide adequately for internal control and data security?
13. Do the systems specifications provide adequately for requested audit features?
14. Has an appropriate configuration for hardware and software been selected for implementation of the systems design and specifications?

15. Have the hardware and software selected been reviewed for adequacy of internal control, data security, integrity, and dependability?
16. Do systems specifications provide adequately for corporate standards and practices?
17. Have systems acceptance criteria been updated?
18. Has the systems test plan been updated?
19. Has data administration reviewed systems specifications?
20. Has data security reviewed systems specifications?
21. Has quality assurance reviewed systems specifications?
22. Has data processing operations reviewed systems specifications?
23. Have cognizant user departments reviewed systems specifications?
24. Has the risk analysis been updated?
25. Have systems specifications been submitted to the management steering committee for action?
26. Have responsible departments signed off for systems specifications?
27. Has the internal auditor prepared a milestone report with opinions and recommendations for the systems specifications phase?

Systems Development

1. Has a detailed work plan been prepared for the systems development phase?
2. Has the systems development methodology been used effectively during the systems development phase?
3. Has the project management and control system been used effectively during the systems development phase?
4. Has actual accomplishment during systems development been reasonably close to estimates?
5. Have significant changes to systems specifications been controlled and approved by cognizant authority?
6. Are systems development team resources adequate to accomplish objectives of systems development phase?
7. Have time and cost estimates, cost/benefit analysis, impact study, and risk analysis been updated?
8. Have significant changes to project scope been approved by the management steering committee?
9. Do program specifications and user procedures reflect accurately approved systems specifications?
10. Do program specifications and user procedures provide adequately for internal control and data security?
11. Do program specifications and user procedures provide adequately for requested audit features?

12. Are data elements, including interfacing data sets, entered in the data dictionary?
13. Have procedures and/or programs been developed and documented for:
    a. Loading data files?
    b. Initializing data files?
    c. Systems conversion?
    d. Year-end processing?
    e. Onsite backup and recovery?
    f. Offsite backup and recovery?
14. Is there a detailed, written training plan?
15. Is there a detailed, written test plan, including:
    a. Unit test?
    b. Integrated test?
    c. Systems test, including interfaces?
    d. Pilot test?
    e. Acceptance test?
    f. Parallel test?
16. Has a test coordinator been assigned?
17. Are tests documented well?
18. Have all tests been reviewed in detail by at least one level?
19. Have the test results been reviewed by the internal auditor and are they satisfactory?
20. Do products of the systems development phase conform with corporate standards and practices?
21. Have products of the systems development phase been submitted to the management steering committee for action?
22. Have responsible departments signed off for products of the systems development phase?
23. Has the internal auditor prepared a milestone report with opinions and recommendations for the systems development phase?

Implementation

1. Has a detailed work plan been prepared for the systems implementation phase?
2. Are the results of the pilot test and acceptance test satisfactory?
3. Has data processing operations conducted a systems turnover evaluation and is the result satisfactory?
4. Is the system documented adequately?
5. Has internal control review been made?
6. Is the level of internal control satisfactory?
7. Are the results of the parallel test satisfactory?
8. Is the result of the test of backup and recovery tests satisfactory?

9. Have responsible departments approved the system for implementation?
10. Has the management steering committee approved the system for implementation?
11. Has the internal auditor prepared a milestone report with opinions and recommendations for systems implementation?

Post-Implementation

1. Has the internal auditor conducted a detailed review of the system and its environment at about six months after initial implementation?
2. Is the level of internal control and security adequate?
3. Does the system meet original objectives satisfactorily?
4. Has documentation been maintained current?
5. Has change control been maintained?
6. Has systems logic been evaluated using statistical sampling techniques?
7. Has the internal auditor prepared a report with opinions and recommendations?

Summary

An IS auditor gathers insights from the methodology used for development and assesses its suitability for the system. This can be also done by interviewing the project management team. Once the auditor is convinced about its suitability, he can check compliance with the stated process.

The reviewing process generally entails

- Identifying subsystems and modules, their goals and user functionality expectations Identifying and reviewing risks involved in the project
- Incorporating controls in the design
- Advising on control aspects of a design and ensuring that these recommendations are implemented
- Ensuring that the system meets organizational objectives
- Maintaining the quality of the delivered products and reviewing the change management process
- Assessing the effectiveness of the system
- Reviewing maintenance procedures and ensuring that adequate documentation is maintained during change implementation
- Ensuring production and source integrity

Multiple Choice Questions:

1. The auditor should collect a copy of System Requirements Specifications (SRS) document and review it in terms of:
   a. Problem definition
   b. Information flows
   c. Both (a) and (b)
   d. None of these

2. The decision to acquire the software should flow from the:
   a. Feasibility study
   b. Requirements definition
   c. Designing
   d. Programming

3. While reviewing the feasibility study phase, the IS auditor has to consider the following question/s:
   a. Has there been an agreement in the definition of the problem among all stakeholders?
   b. Was there a genuine need for solution established?
   c. Were alternate solutions considered? Or was the feasibility assessed on the basis of a single solution?
   d. All of the these

4. In the Post-implementation Review, the auditor has to check the following aspects:
   a. Compliance with change control procedure
   b. Functioning of controls in accordance with design
   c. Review of operator error logs
   d. All of these

5. In testing phase, the auditors should check the following point/s:
   a. Test plans for completeness
   b. Cyclical processing such as month-end reports
   c. Security functions of the system
   d. All of these

6. In Implementation phase, the auditors should check the following point/s:
   a. The documentation on parallel run, if available, should be reviewed for effectiveness.
   b. Operating procedures should be checked for clarity and accuracy.
   c. Both (a) and (b)
   d. None of these

7. On a periodic basis, the auditor should check the following:
   a. Procedures for authorizing, prioritizing and tracking system changes
   b. Appropriateness of authorizations for selected change requests
   c. Existence of program change history
   d. All of these

8. The auditor does not undertake this task of identification himself. He goes through the _____ findings and crosschecks them with _____.
   a. analyst's, users
   b. users, analyst's
   c. programmer's, users
   d. users, programmer's

9. In the post implementation phase, the delivery and project management team may cease to exist, so the auditor will have a greater role to play in assessing the _____ of the system.
   a. Correctiveness
   b. Usefulness
   c. Effectiveness
   d. None of these

10. _____ controls can also have an adverse effect on the systems, so the auditor should continue to monitor the maintenance procedures.
    a. Lux
    b. Lax
    c. Pux
    d. Pax

11. In _____ analysis, there could be a tendency to understate costs and overstate benefits.
    a. cost-benefit
    b. financial
    c. cost
    d. benefit

12. In the recommendation of _____ and _____ choices, the auditor should look for compatibility, interoperability and scalability conditions.
    a. Input, Output
    b. Hardware, Software
    c. Data, Information
    d. Design, Review

13. For each system, the auditor:
    a. elicits the information about the methodology used for development
    b. assesses its suitability for the system
    c. Both (a) and (b)
    d. None of these

14. In the term 'CASE Tools', 'CASE' stands for:
    a. Computer Aided Software Engineering
    b. Case Aided Software Engineering
    c. Computer Aided Security Engineering
    d. Case Aided Security Engineering

15. RFP should be checked for_____:
    a. Adequacy
    b. CASE Tools
    c. Both (a) and (b)
    d. None of these

Answers:

1. (c)	2. (a)	3. (d)	4. (d)	5. (d)	6. (c)
7. (d)	8. (a)	9. (c)	10. (b)	11. (a)	12. (b)
13. (c)	14. (a)	15. (a)			