**CHAPTER 1: CONCEPTS OF SYSTEM ANALYSIS AND DESIGN**

**WHAT IS A SYSTEM?**

The word System is derived from Greek word Systema, which means an organized relationship between any set of components to achieve some common cause or objective.

***A system is an orderly grouping of interdependent components linked together according to a plan to achieve a specific goal.***

**CONSTRAINTS OF A SYSTEM**

A system must have three basic constraints −

* A system must have some **structure and behavior** which is designed to achieve a predefined objective.
* **Interconnectivity** and **interdependence** must exist among the system components.
* The **objectives of the organization** have a **higher priority** than the objectives of its subsystems.

**Examples of systems**

* Traffic management system
* Payroll system
* Automatic library system
* Human resources information system.

**PROPERTIES OF A SYSTEM**

A system has the following properties −

***i. Organization***

Organization implies structure and order. It is the arrangement of components that helps to achieve predetermined objectives.

***ii. Interaction***

It is defined by the manner in which the components operate with each other.

For example, in an organization, purchasing department must interact with production department and payroll with personnel department.

***iii. Interdependence***

Interdependence means how the components of a system depend on one another. For proper functioning, the components are coordinated and linked together according to a specified plan. The output of one subsystem is the required by other subsystem as input.

***iv. Integration***

Integration is concerned with how a system components are connected together. It means that the parts of the system work together within the system even if each part performs a unique function.

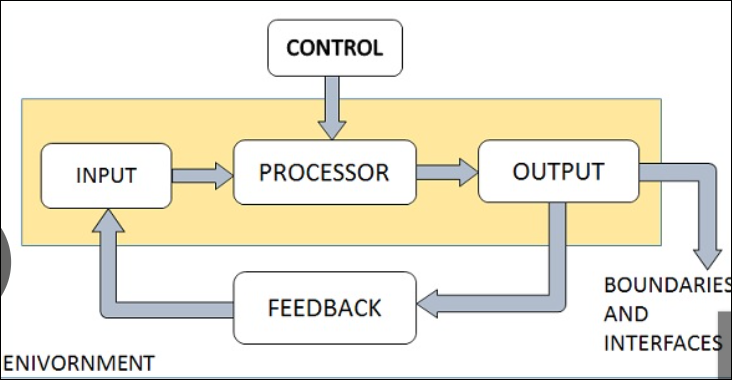
***v. Central Objective***

The objective of system must be central. It may be real or stated. It is not uncommon for an organization to state an objective and operate to achieve another.

The users must know the main objective of a computer application early in the analysis for a successful design and conversion.

**ELEMENTS OF A SYSTEM**

The following diagram shows the elements of a system −



***i. Outputs and Inputs***

* The main aim of a system is to produce an output which is useful for its user.
* Inputs are the information that enters into the system for processing.
* Output is the outcome of processing.

***ii. Processor(s)***

* The processor is the element of a system that involves the actual transformation of input into output.
* It is the operational component of a system. Processors may modify the input either totally or partially, depending on the output specification.
* As the output specifications change, so does the processing. In some cases, input is also modified to enable the processor for handling the transformation.

***iii. Control***

* The control element guides the system.
* It is the decision making subsystem that controls the pattern of activities governing input, processing, and output.
* The behavior of a computer System is controlled by the Operating System and software. In order to keep system in balance, what and how much input is needed is determined by Output Specifications.

***iv. Feedback***

Information about the output that is sent back to the system to measure performance and make adjustments.

***v. Environment***

* The environment is the super system within which an organization operates.
* It is the source of external elements that strike on the system.
* External area that influences a system's operation and consists of elements like vendors, customers, competitors, and other external factors that the system interacts with

***vi. Boundaries and Interface***

* A system should be defined by its boundaries. Boundaries are the limits that identify its components, processes, and interrelationship when it interfaces with another system.
* Each system has boundaries that determine its sphere of influence and control.

**Example: An e-commerce system**

* Inside the boundary: The website's user interface, the shopping cart functionality, payment processing, inventory tracking, and the customer database.
* Outside the boundary (external entities): The customer, the external bank or payment gateway, the shipping company, and the marketing department.

**Interfaces** are the points where the system interacts with its environment, exchanging information, services, or resources. While boundaries define what is inside and outside the system, interfaces specify how the system interacts with the outside world.

**Example: The e-commerce system interfaces**

* Customer Interface: The website's user-facing pages, such as the product search screen, shopping cart, and checkout page.
* Payment Gateway Interface: The set of rules and protocols that allow the e-commerce system to communicate with an external bank or credit card company to process a transaction.
* Shipping API Interface: The specification that defines how the system sends shipment information (e.g., customer address, package weight) to an external delivery.

**SYSTEM ANALYSIS**

System analysis is the initial phase of a software development project where the requirements of the system are gathered, analyzed, and documented. It involves understanding the problem domain, identifying the stakeholders, and defining the scope and objectives of the system.

**Key Activities in System Analysis**

1. **Requirement Gathering**− Identifying the needs and expectations of the users and stakeholders.
2. **Requirement Analysis**− Analyzing the gathered requirements to ensure consistency, feasibility, and completeness.
3. **Feasibility Study**− Assessing the technical, economic, and operational feasibility of the proposed system.
4. **Process Modelling**− Creating diagrams and models to represent the current and proposed business processes.
5. **Data Modelling**− Defining the data entities, attributes, and relationships within the system.

**Techniques Used in System Analysis**

1. **Interviews**− Gathering information from stakeholders through face-to-face or online interviews.
2. **Surveys**− Collecting data from a large number of respondents using questionnaires.
3. **Observation**− Observing the current system in operation to understand its processes and workflows.
4. **Document Analysis**− Examining existing documents, reports, and manuals.
5. **Prototyping**− Creating simplified models or mock-ups of the system to gather feedback and refine requirements.

**SYSTEM DESIGN**

System design is the subsequent phase where the detailed specifications of the system are developed. It involves designing the architecture, components, interfaces, and data structures that will implement the requirements defined in the analysis phase.

**Key Activities in System Design**

1. **Architectural Design**− Determining the overall structure and components of the system.
2. **Component Design**− Designing individual components and their interactions.
3. **Interface Design**− Specifying the interfaces between components and with external systems.
4. **Data Design**− Designing the database schema and data structures.
5. **Detailed Design**− Creating detailed specifications for each component, including algorithms and data flow.

**Techniques Used in System Design**

1. **Unified Modelling Language (UML)**− A standardized modelling language used to visualize, specify, construct, and document software systems.
2. **Data Flow Diagrams (DFDs)**− Diagrams that illustrate the flow of data through a system.
3. **Entity-Relationship Diagrams (ERDs)**− Diagrams that represent the entities and relationships between them in a database.
4. **Decision Trees**− Diagrams that show the possible outcomes and decisions in a process.
5. **State Transition Diagrams**− Diagrams that represent the different states a system can be in and the transitions between them.

**Key Differences Between System Analysis and System Design**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.No.** | **Feature** | **System Analysis** | **System Design** |
| 1 | Focus | Understanding the problem domain and gathering requirements. | Specifying the solution and designing the system. |
| 2 | Output | Requirements document | System design specifications |
| 3 | Techniques | Interviews, surveys, observation, document analysis | UML, DFDs, ERDs, decision trees, state transition diagrams |
| 4 | Level of Detail | High-level understanding | Detailed specifications |

**THE RELATIONSHIP BETWEEN SYSTEM ANALYSIS AND SYSTEM DESIGN**

System analysis and system design are closely interconnected. The output of the analysis phase (the requirements document) serves as the input for the design phase. The design specifications must align with the requirements to ensure that the developed system meets the needs of the users and stakeholders.

**TYPES OF SYSTEMS**

The systems can be divided into the following types −

**i. Physical or Abstract Systems**

# Physical Systems - are tangible systems that can be **touched, seen, and physically interacted with**.

**Traits:**

* Made up of physical components (machines, hardware, documents, people).
* Exist in the real world.
* Often the **infrastructure** supporting an information system.

**Examples in SAD:**

* A **computer system** with CPU, monitor, keyboard.
* **Bank ATMs** processing cash withdrawals.
* **Hospital equipment** connected to a health information system.

***Advantage***: Concrete, easy to identify and observe.

***Disadvantage***: Cannot fully explain intangible processes (like data flow).

# Abstract Systems are Conceptual, intangible systems made up of ideas, relationships, or logical representations rather than physical objects.

**Traits:**

* Focus on **relationships, models, and processes.**
* Cannot be touched but can be visualized or represented.
* Useful for system modeling and analysis.

**Examples in SAD:**

* **Data Flow Diagram (DFD)** showing how information moves.
* **Mathematical models** (e.g., forecasting demand).
* **Conceptual frameworks** of an information system.

***Advantage***: Helps simplify and analyze complex systems.

***Disadvantage***: May be too theoretical, harder for non-technical users to understand.

**ii. Open or Closed Systems**

* ***An open system*** is a system that **interacts with its environment**, exchanging inputs, outputs, and feedback.

**Characteristics:**

* + Accepts **inputs** (data, resources, energy) from the environment.
  + Produces **outputs** (information, products, services) back to the environment.
  + Influenced by external factors (technology, users, regulations, competitors).
  + Adaptable and flexible.

**Examples in SAD:**

* + A university student information system (takes student data, processes it, produces results/transcripts).
  + An e-commerce platform (interacts with customers, suppliers, banks).

***Advantage*:** More realistic, adaptable, responsive to changes.

***Disadvantage:*** More complex to design and manage due to external dependencies.

* ***A closed system*** - A system that does not interact with its environment; it is self-contained and isolated.

**Characteristics:**

* No exchange of inputs or outputs with the environment.
* Operates independently, with little external influence.
* More stable but less adaptable.

**Examples in SAD:**

* A calculator (works independently, no external interaction after input is given).
* A standalone desktop program (not connected to the internet or other systems).

***Advantage:*** Simpler, easier to design and control.

***Disadvantage:*** Unrealistic for most real-world systems, limited adaptability.

**iii. Adaptive and Non Adaptive System**

* ***Adaptive Systems*** that can adjust themselves in response to environmental changes (internal or external).

**Key Traits:**

* Flexible and self-regulating.
* Take **feedback** from the environment and make necessary adjustments.
* Can evolve over time to meet new requirements.

**Examples in SAD:**

* A modern **traffic control system** that adjusts signals based on real-time traffic flow.
* **E-commerce recommendation systems** (Amazon, Netflix) that adapt to user behavior.
* **Self-healing cloud systems** that reallocate resources when failures occur.

***Advantage:*** Resilient, sustainable in dynamic environments.

***Disadvantage:*** More complex and costly to design.

* ***Non-Adaptive Systems***

***Non-Adaptive Systems*** areSystems that cannot adjust automatically to environmental changes. They continue operating in the same manner regardless of feedback.

**Key Traits:**

* Rigid and static.
* Fail or perform poorly if the environment changes significantly.
* Depend on manual intervention for updates.

**Examples in SAD:**

* A **basic calculator** (same logic, no adjustment to context).
* A **manual payroll system** that requires human updates for new rules.
* Legacy systems that do not update automatically with new business policies.

***Advantage***: Simpler, cheaper to build and maintain.

***Disadvantage:*** Less flexible, quickly outdated.

**iv. Permanent or Temporary System**

* ***Permanent system*** is a system that is designed to **exist and operate indefinitely**, usually with ongoing maintenance and upgrades.

**Key Traits:**

* Long-term lifespan.
* Continuously used in the organization.
* Evolving through updates, patches, or enhancements.
* Integral to business operations.

**Examples in SAD:**

* **Banking information systems** (core banking solutions used for decades).
* **University student management systems.**
* **ERP systems** in companies (SAP, Oracle).

**Advantages:** Reliable, sustainable, supports long-term goals.

**Disadvantages:** Requires constant maintenance, higher cost over time.

# Temporary System – A system created for a short duration to achieve a specific objective or support a temporary activity.

**Key Traits:**

* Limited lifespan (ends when its purpose is fulfilled).
* Often discarded or replaced once the task is complete.
* Lower investment and minimal maintenance.

**Examples in SAD**

* A **survey system** created to collect data for one-time research.
* A **conference registration system** used only during an event.
* A **pilot project system** for testing feasibility before full deployment.

**Advantages:** Low cost, purpose-specific, quick to deploy.

**Disadvantages:** Not sustainable, lacks scalability, often discarded.

**vi. Deterministic or Probabilistic System**

# *Deterministic System* - **Is a**system in which the **output is predictable** and determined precisely by the input and process.

**Key Traits:**

* No randomness or uncertainty.
* Always produces the same output for the same input.
* Easier to analyze, test, and design.

**Examples in SAD:**

* A **payroll system** → given salary rules + employee data = predictable output.
* A **calculator** → 2 + 2 will always = 4.
* An **inventory system** → deducts exact number of sold items.

**Advantages:** Reliable, consistent, easy to maintain.

**Disadvantages:** Cannot handle uncertainty or unpredictable events.

**SYSTEMS MODELS**

A **system model** is a simplified representation of a system that helps analysts, designers, and stakeholders understand how the system works, how components interact, and how information flows.

Models are important because they **reduce complexity** and make communication easier.

**Types of System Models in SAD**

**i. Physical System Model**

* Focuses on the actual physical components of the system.
* Deals with hardware, devices, and physical interactions.
* Example: Diagram of servers, routers, and user devices in a banking system.

**ii. Abstract/Conceptual Model**

* A logical representation of the system without physical details.
* Emphasizes ideas, functions, and relationships.
* Example: Data Flow Diagram (DFD) showing how information moves between processes.

**iii. Deterministic vs Probabilistic Models**

* Deterministic: Output is predictable (e.g., payroll system).
* Probabilistic: Output is uncertain, based on probabilities (e.g., weather forecasting system).

**iv. Static vs Dynamic Models**

* Static Model: Shows the system at a fixed point in time (e.g., ER diagram of database).
* Dynamic Model: Shows changes in the system over time (e.g., state transition diagram, activity diagram).

**v. Mathematical Models**

* Represent systems using formulas, statistics, or algorithms.
* Useful for optimization, simulation, and performance analysis.
* Example: Formula to calculate waiting times in a queuing system.

**vi. Functional Models**

* Focus on what the system does.
* Include:
  + Data Flow Diagrams (DFD)
  + Use Case Diagrams (UML)

**vii. Object-Oriented Models**

* Focus on objects (entities with attributes & behaviors).
* Include:
  + Class Diagrams
  + Sequence Diagrams
  + Object Diagrams

**Why Use System Models in SAD?**

1. Simplify complex systems
2. Communicate ideas clearly to stakeholders
3. Identify gaps and errors early
4. Provide a blueprint for design and development
5. Help in documentation and future maintenance

**IMPORTANCE OF SYSTEM ANALYSIS AND DESIGN IN SYSTEMS DEVELOPMENT**

1. Identifying Requirements: System analysis helps determine the specific needs and requirements of the users and the business. This includes understanding the current processes, identifying gaps, and gathering information from stakeholders.

2. Enhancing Efficiency and Reducing Costs: By analyzing existing systems and processes, system analysis can pinpoint inefficiencies and areas where costs can be reduced. This can involve streamlining processes, automating tasks, and optimizing resource allocation.

3. Improving System Quality: A well-designed system, built on a solid analysis, is more likely to be reliable, maintainable, and scalable. It can also be more user-friendly, leading to greater employee productivity.

4. Reducing Risks: System analysis can help identify potential risks and threats early in the development process, allowing for proactive mitigation measures. This can include security vulnerabilities, technical challenges, and integration issues.

5. Better Management and Planning: System analysis provides a framework for managing the development process, from planning to implementation. It allows for better communication between stakeholders, facilitates decision-making, and ensures that the final system aligns with the organization's strategic goals.