# Real-Time Systems

Real-Time systems are becoming pervasive. Typical examples of real-time systems include Air Traffic Control Systems, Networked Multimedia Systems, Command Control Systems etc. In a Real-Time System the correctness of the system behavior depends not only on the logical results of the computations, but also on the physical instant at which these results are produced. Real-Time systems are classified from a number of viewpoints i.e. on factors outside the computer system and factors inside the computer system. Special emphasis is placed on hard and soft real-time systems. A missed deadline in hard real-time systems is catastrophic and in soft real-time systems it can lead to a significant loss. Hence predictability of the system behavior is the most important concern in these systems. Predictability is often achieved by either static or dynamic scheduling of real-time tasks to meet their deadlines. Static scheduling makes scheduling decisions at compile time and is off-line. Dynamic scheduling is online and uses schedulabilty test to determine whether a set of tasks can meet their deadlines.

Real-Time systems span several domains of computer science. They are defense and space systems, networked multimedia systems, embedded automative electronics etc. In a real-time system the correctness of the system behavior depends not only the logical results of the computations, but also on the physical instant at which these results are produced. A real-time system changes its state as a function of physical time, e.g., a chemical reaction continues to change its state even after its controlling computer system has stopped. Based on this a real-time system can be decomposed into a set of subsystems i.e., the controlled object, the real-time computer system and the human operator. A real-time computer system must react to stimuli from the controlled object (or the operator) within time intervals dictated by its environment. The instant at which a result is produced is called a deadline. If the result has utility even after the deadline has passed, the deadline is classified as soft, otherwise it is firm. If a catastrophe could result if a firm deadline is missed, the deadline is hard. Commands and Control systems, Air traffic control systems are examples for hard real-time systems. On-line transaction systems, airline reservation systems are soft real-time systems.

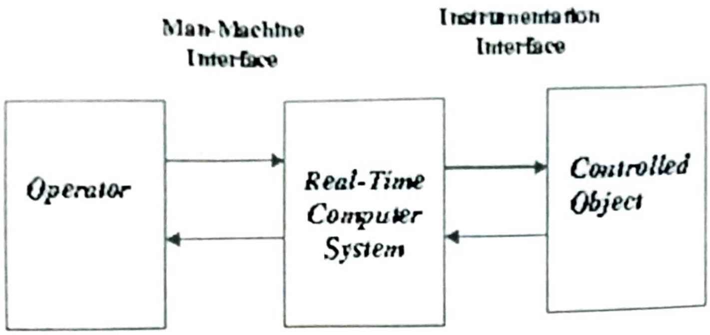
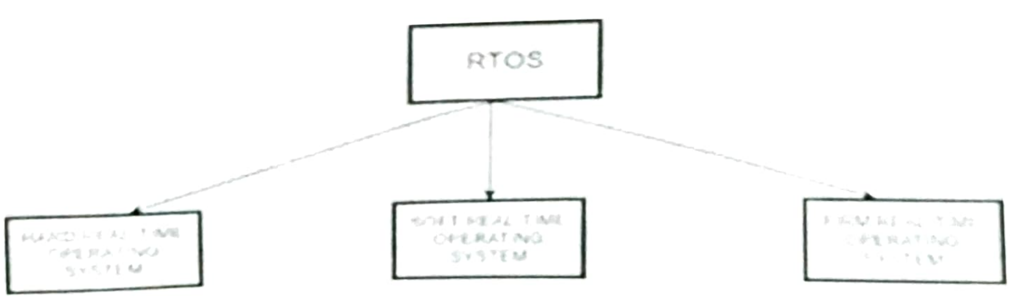



Figure 1: Real-Time System

Real-time **operating systems (RTOS)** are used in environments where a large number of events, mostly external to the computer system, must be accepted and processed in a short time or within certain deadlines. such applications are industrial control, telephone switching equipment, flight control, and real-time simulations. With an RTOS, the processing time is measured in tenths of seconds. This system is time-bound and has a fixed deadline. The processing in this type of system must occur within the specified constraints. Otherwise, This will lead to system failure.
Examples of the real-time operating systems: Airline traffic control systems, Command Control Systems, Airlines reservation system, Heart Pacemaker, Network Multimedia Systems, Robot etc.

The real-time operating systems can be of 3 types –



**Hard Real-Time operating system:**

These operating systems guarantee that critical tasks be completed within a range of time.
For example, a robot is hired to weld a car body. If the robot welds too early or too late, the car cannot be sold, so it is a hard real-time system that requires complete car welding by robot hardly on the time., scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

**Soft real-time operating system:**

This operating system provides some relaxation in the time limit.
For example – Multimedia systems, digital audio systems etc. Explicit, programmer-defined and controlled processes are encountered in real-time systems. A separate process is changed with handling a single external event. The process is activated upon occurrence of the related event signalled by an interrupt.

Multitasking operation is accomplished by scheduling processes for execution independently of each other. Each process is assigned a certain level of priority that corresponds to the relative importance of the event that it services. The processor is allocated to the highest priority processes. This type of schedule, called, priority-based preemptive scheduling is used by real-time systems.

**Firm Real-time Operating System:**

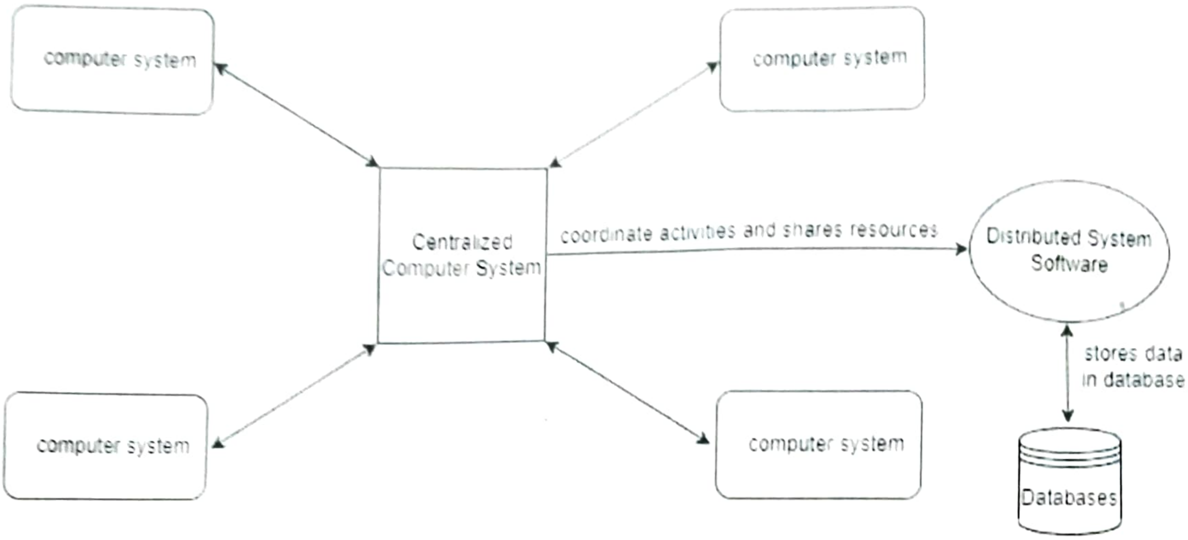
RTOS of this type have to follow deadlines as well. In spite of its small impact, missing a deadline can have unintended consequences, including a reduction in the quality of the product. Example: Multimedia applications.

# Distributed System?

Distributed System is a collection of autonomous computer systems that are physically separated but are connected by a centralized computer network that is equipped with distributed system software. The autonomous computers will communicate among each system by sharing resources and files and performing the tasks assigned to them.

**Example of Distributed System:**

Any Social Media can have its Centralized Computer Network as its Headquarters and computer systems that can be accessed by any user and using their services will be the Autonomous Systems in the Distributed System Architecture.




- **Distributed System Software:** This Software enables computers to coordinate their activities and to share the resources such as Hardware, Software, Data, etc.
- **Database:** It is used to store the processed data that are processed by each Node/System of the Distributed systems that are connected to the Centralized network.

# SSADM (Structured Systems Analysis and Design Method)

SSADM (Structured Systems Analysis and Design Method) is a widely used computer application development method in the United Kingdom, where its use is often specified as a requirement for government computing projects. SSADM is in the public domain and is formally specified in British Standard BS7738.

SSADM divides an application development project into modules, stages, steps and tasks. It provides a framework for describing a project in a way that's suited to managing it. SSADM's objectives are to do the following:

- improve project management and control;
- make more effective use of experienced and inexperienced development staff;
- develop better quality systems;
- make projects resilient to the loss of staff;
- enable projects to be supported by computer-based tools such as computer-aided software engineering systems (CASE); and
- establish a framework for good communications among project participants.

Consultants working for Learmonth & Burchett Management Systems developed SSADM v1 in 1981. The methodology covers system lifecycle aspects from the feasibility study to the production of a physical design. It's generally used in conjunction with other methods, such as PRINCE2 (Projects in Controlled Environments), which focuses on broader aspects of project management.

SSADM is applicable to projects with technical and business requirements that are unlikely to change. It uses a top-down waterfall project management approach to information systems projects. SSADM isn't suitable for projects in a volatile business environment, so Agile should be used instead.

## SSADM techniques

The following three data modeling techniques are used in SSADM:

**1. Logical data modeling.** A logical data model depicts the logical data structure of the system and deals with documenting the data requirements of the system. The model describes:

- entities that are objects in the system and have information recorded about them;
- attributes that are descriptors of the entities; and
- relationships that are the associations between the entities

**2. Data flow modeling.** Data flows are identified, modeled and documented. Data flow diagrams consist of the following:

- processes that transform data;
- external entities that send data from a system and receive data from another system;
- data stores that hold data at rest; and
- data flows to move data among entities.

**3. Entity behavior modeling.** This technique focuses on identifying, modeling and documenting the events that occur in the system. It looks at how events affect the entity and the sequence in which they occur.

## What are the stages of SSADM?

SSADM uses a cascade or waterfall view of system development, in which a series of steps are followed. This contrasts with the rapid application development method that conducts steps in parallel.

The following steps are part of SSADM.

**1. Feasibility stage.** A feasibility study investigates the goals and implications of a project to determine whether the project is achievable. The existing system is analyzed for problems that must be solved during development. There are four main areas considered in a feasibility study.

- A technical feasibility study to determine if the project is technically possible.

- Organizational feasibility to evaluate whether the project is compatible with existing systems.
- An ethical feasibility step to establish if the project is socially and ethically acceptable.
- A financial feasibility study to verify if the project is financially practical.

**2. Requirements analysis.** This step investigates the current environment and examines existing business systems to see what modifications are needed. This includes a cost-benefits analysis of business system options and the potential effect of the new system on the organization and employees.

**3. Definition of requirements.** The requirements specification for the new system may include a look at the life histories of the entities involved, user role and function matrices, and process identity modeling, data catalogs and correspondence diagrams.

**4. Logical system specification.** This step consists of two stages:

- Technical system options define possible implementation methods and components, such as staffing costs, the type of software and hardware required, physical limitations of the area that houses the system and requirements for the human-computer interface. At the end of this stage, a technical system option is chosen.
- Logical design defines the logic used in the new system, with a specific focus on human-computer interaction and logical dialog with the system.

**5. Physical design.** The system's logical specifications are matched to physical architecture. The function and structure of the database component is specified based on outputs from the logical design and technical system option steps.

For each stage, SSADM sets out a series of techniques, procedures and conventions to record and communicate information, both in textual and diagrammatic form. SSADM is supported by numerous CASE tool providers.

*SSADM is used today for some large government IT projects primarily in the U.K. Unlike smaller more dynamic software development projects, large public sector projects often require a standard methodology to upgrade legacy systems and manage processes. Learn some business process management best practices to ensure continuous improvement in the enterprise and compare them with the SSADM methodology.*