University of Msila
FACULTY OF MATHEMATICS AND INFORMATICS
DEPARTMENT OF COMPUTER SCIENCE

Modeling of dynamic discret events systems

(DDES)

Introduction

- **Modeling a system**: Modeling consists of defining the following points:
  - **The system**: existing (or not) to which the model refers.
  - **The model**: abstract representation of the system (simplified)
  - **The objective:** the purpose for which the model was developed.
  - **A profitability criterion**: an economic criterion that justifies the use of a model

- Modeling a dynamic system first requires

1. the definition of a time scale
2. the definition of an input variable u and an output variable y.
   - How to define the input and output of a system?
   - The general rule is to define as inputs all variables that can be controlled or modified and as outputs all variables that can be measured.

Introduction

- **Modeling methods:**

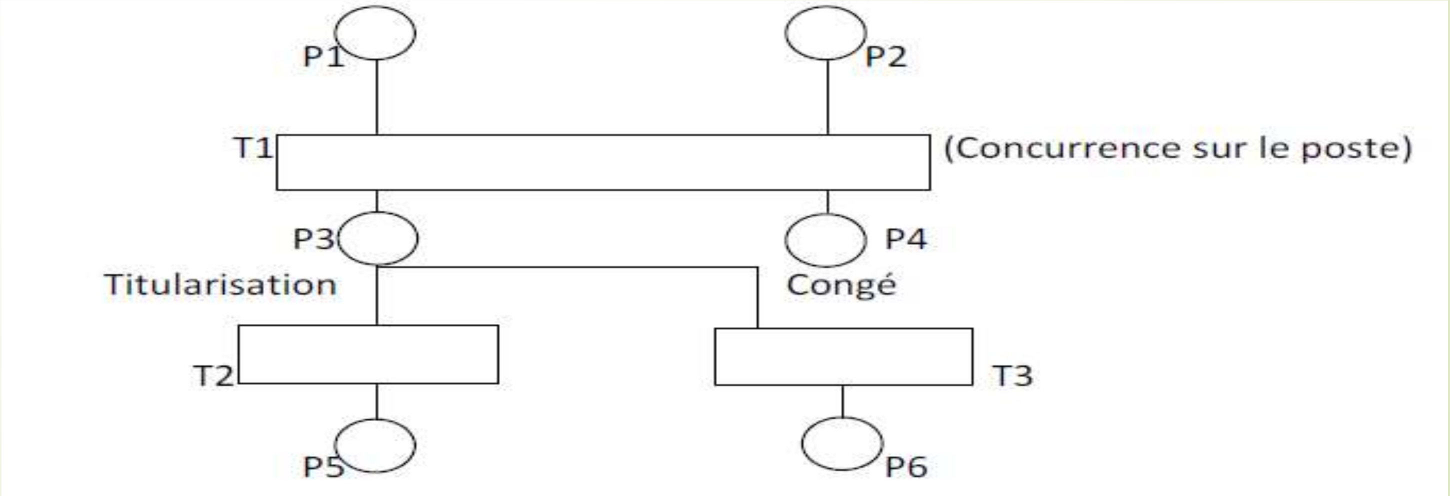
Several modeling methods are used today. Each of them being more or less well adapted to specific aspects of analyzing the performance of a given system.

The classes of modeling methods are divided into two classes:

1.descriptive methods : They make it possible to describe the logical behavior of a system generally without involving the temporal aspect.

An example of descriptive methods: Petri nets. They provide a graphical model for describing the operations of a computing machine. They are particularly suited to describing phenomena of competition, conflict and synchronization.

Example: Personnel management by RDP Anyone can request recruitment depending on the availability of budget positions. An employee can be tenured An employee can go on leave, training, etc.



Introduction

- **Analytical methods**: These are methods based on mathematical models, and their results are obtained by calculation. Much research in the field of modeling has focused on queuing theory. Several queuing models are established.

Example :

- **the M/M/1 model**: Arrival following Poisson's law Service according to the exponential law A waiter
- **the M/M/S model**: Same thing **S** servers
- **Example** of queue modeling Arrival of customers in front of a bank counter Goals :

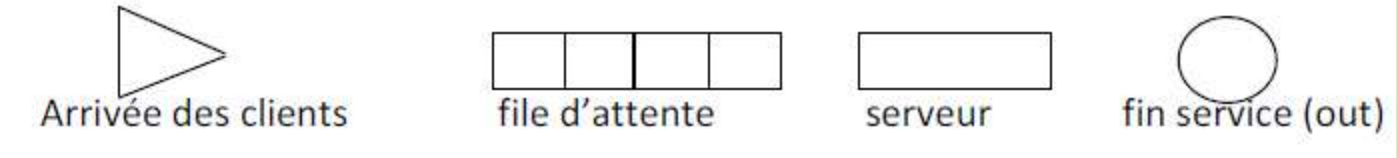
  - understand the function of the system

  -Reduce waiting time

Alternatives:

-One or two wickets?

-One or two queues? An analytical method may have aspects of a descriptive method
Example: representation of a queue.



Discrete event system

➢ **Components of a discrete event model**

A model is characterized by a static aspect linked to the structure of the modeled system, and by a dynamic aspect linked to the evolution of the system over time.

- **Static part** A discrete event model is composed of objects or entities (parts, machines, etc.) with which services are carried out over time, which can be active (activities or operations) or passive (waiting), and relationships between these objects (possibility for a part to pass through a machine, etc.).
- There are generally two types of objects:
  - customers or users: On whom the services are carried out (parts in a workshop, customers in a store, information in a transmission network, etc.); often, customers are circulating objects (particularly in systems of production).
  - resources: Are necessary to carry out the services; these resources can be composed of one or more units (number of identical machines, personnel with the same skill, trolleys, places in a queue, etc.).

Discrete event system

- **Dynamic part**

The dynamic aspect of a model refers to the mechanisms of state changes. When time evolves, interactions occur between the objects that make up the model and they then change state. These state changes can be done continuously, discretely, or a combination of the two. This is how we speak of “continuous” models, “discrete event” models, and “combined continuous discrete event” models.

CONCEPTS RELATED TO THE DISCRETE EVENT SYSTEM SIMULATION METHOD

- **SYSTEM STATE VARIABLES** The state variables of a system are all the information necessary to define or characterize what is happening in the system at a sufficient level of detail at a given time.

- **Example:** In a bank, the unoccupied time of the counters is of no interest or does not constitute a state variable of the system analyzed if we decide to assign a customer to the first free counter and this in a cyclical manner. On the other hand, if we decide to assign a customer as a priority to the counter which has remained unoccupied the longest, then the unoccupied time of the counters becomes a state variable of the system.

- In a discrete event model, state variables remain constant over time intervals and change values only at certain well-defined points called: Times or instants of events. On the other hand, continuous models have state variables defined by differential or difference equations which allow the variables to change continuously over time.

CONCEPTS RELATED TO THE DISCRETE EVENT SYSTEM SIMULATION METHOD

- **ENTITIES, RESOURCES, ATTRIBUTES**
  - **Entities** designate objects of the modeled system. The term itself can designate both passive objects which undergo operations and active objects which carry out operations.
  - Entities that move in the system (e.g.: customers) during the simulation are called dynamic while entities that do not move and simply serve other entities are called static (e.g.: server in a bank) .
  - A permanent entity is an entity that remains in the system even when the simulation is finished (e.g. machine, server).
  - A temporary entity is an entity that undergoes operations and leaves the system as soon as it completes its operations.
  - **Resources:** these are objects that execute operations and generally do not move within the system (Machine, Operator, Central Unit, etc.). However, it should be noted that it is entirely possible to encounter in practice resource type objects which move inside the system while executing the operation (cart transporting a part in a workshop).
  - There are several possible states for a resource. The two minimum states are: free or busy but other possibilities exist such as: broken down, blocked, starved, etc.

CONCEPTS RELATED TO THE DISCRETE EVENT SYSTEM SIMULATION METHOD

- **An object (entity or resource)** is characterized by one or more attributes to which values can be assigned. Thus the attributes can be considered as variables local to the object.
- we distinguish two types of attributes:

  •**Fixed:** contain the constant characteristics of the object (service life, date of arrival in the system, color of a part, etc.)

  •**Variables**: contain the changing characteristics of the object (state of a resource, length of the queue associated with a resource, machining time remaining for an entity, etc.)

- Attributes that are interesting in one investigation may not be interesting in another. For example, if we have red parts and blue parts to machine, the color attribute will certainly be an interesting attribute. On the other hand, if we are only interested in the residence time in the system for all the parts, the color of a part may not be an important attribute.

CONCEPTS RELATED TO THE DISCRETE EVENT SYSTEM SIMULATION METHOD

- **Event , Activity , Process**
  - **Event**

    An event is characterized by a **date** (event date) on which the system changes state. A distinction is made between **internal** and **external** events, also known as **endogenous** and **exogenous** events.

    Example:

    - The start of customer service is an **endogenous event**, since it is internal to the system we are simulating.

    - The arrival of a customer for service is an **exogenous event**, since it is outside the simulation. However, the arrival of a customer for the service impinges on the system and must be taken into account.

  - **Activity**

    Each time an event occurs, the objects concerned engage in operations. These operations, which are initiated (or terminated) at each event, are called **activities**. Every activity is bounded at its start and end by an event.

    Example:

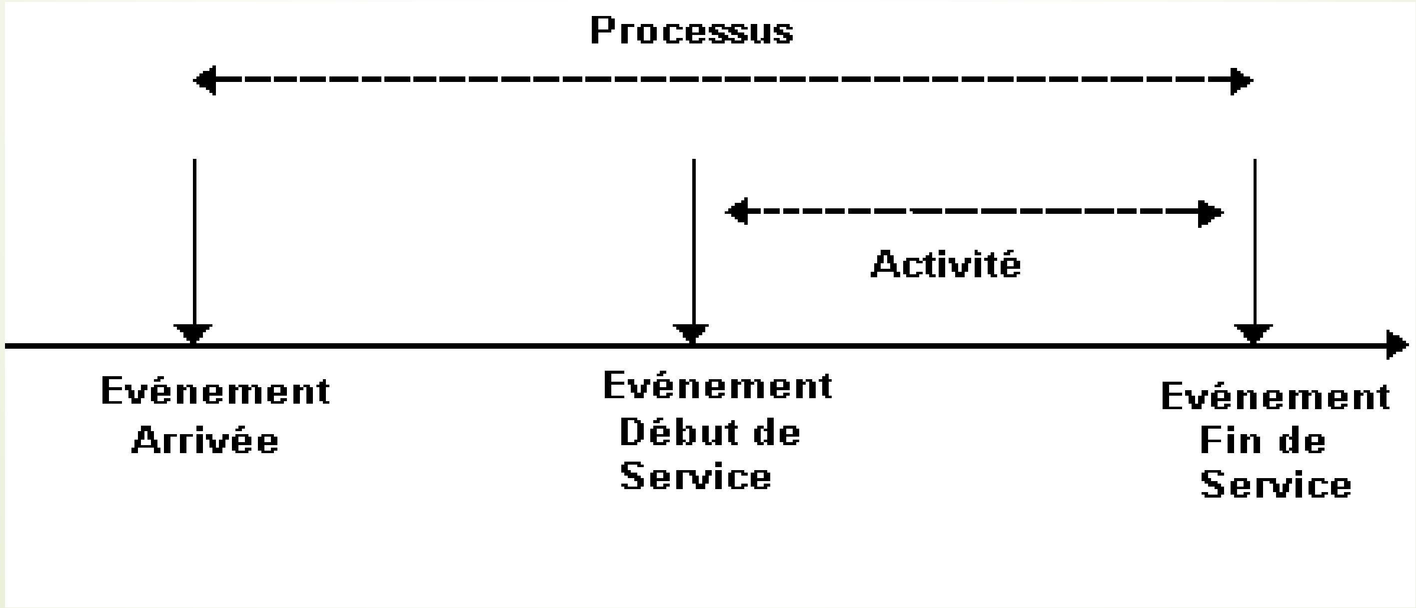
    - The start of service for a customer will initiate the following operations: extract customer from queue, serve customer

CONCEPTS RELATED TO THE DISCRETE EVENT SYSTEM SIMULATION METHOD

- **Process**

A process is the grouping of a sequence of events in the chronological order in which they will occur. As events can initiate activities, a process can also include activities.

- **Example**:- The arrival of a customer, his queuing, the start of his service, can constitute a process associated with the customer. Relationship between Event, Activity, Process



Building a discrete-event model

- To build a discrete-event simulation model, the designer must choose a modeling approach
- If the designer has a simulation language at his disposal, then the approach to be used is implicit, since each language normally offers one (or more) approaches. If, on the other hand, the designer is using a general-purpose programming language (Fortran, etc.), then the choice of approach is entirely his own responsibility.
- The three best-known approaches are
  1) Event-based approach
  2) Activity-based approach
  3) Process interaction approach

Building a discrete-event model

These three modeling approaches are characterized by the fact that they lead to models with a 3-level hierarchical structure:

- **level 1:**

Corresponds to the simulation control program, also known as the **executive** or kernel. The **kernel** is responsible for sequencing the operations (events, activities, processes) that take place as the simulation progresses. Thus, the kernel controls level 2 and includes routines responsible for :
- Identifying when the next event will take place
- Ensuring that the right operations take place at the right time

- In the case of a simulation language, the kernel is a part that is not directly accessible to the programmer, who really doesn't need to know all the details about the kernel. On the other hand, with a general-purpose programming language (Fortran, Pascal, C, etc.), you'll have to write the kernel yourself, and in this case you'll have to master all the kernel's operating details.

Building a discrete-event model

- **level 2**:

This level is the responsibility of the programmer and constitutes the simulation model from the programmer's point of view. It is therefore a set of instructions whose structure depends on the modeling approach adopted. These will be routines describing events in the case of an event-based approach, routines describing activities in the case of an activity-based approach, or routines describing processes in the case of a process-based approach.

- **level 3:**

Includes a set of utility routines called up by level 2. These include routines for generating random numbers, collecting statistics, producing results in report form, and so on. The programmer calls these routines when writing level 2.