Course material

INFORMATION SYSTEM 1

Narcisse TALLA

Department of Computer Science engineering
University Institute of Technology Fotso Victor
University of Dschang
narcisse.talla@univ-dschang.org

2017-2018

SOMMARY

Chapter I. THE INFORMATION SYSTEM

1.1. Definitions

- An information is the result of a process of formatting and materialization to communicate a fact or set of facts to an audience.
- A system is a set of dynamically interacting elements organized according to a goal.
- An **open system** is a system in permanent interaction with its environment: It is a reservoir that fills up (energy, matter, information) and empties (entropy, energy waste) at the same speed to stay within a given state. It is in constant interaction with its environment, modifying it and in return being modified.
- A **closed system** is a system that doesn't exchange matter, information, or energy with its environment: it lives on its internal energy and as feedback, increases its entropy which becomes maximum.
- A **complex system** is a system made of a wide variety of components or parts with specialised functions:
    - Elements organised in **hierarchical levels**,
    - Elements and levels connected by multiple **links**,
    - **Non linear** interactions,
    - Difficult and **unpredictable** behavior
    - **Great resistance** to change.

Any system has two aspects:

- The structural aspect (structural or spatial description).
- The functional aspect (temporal description).

1.2. The concept of "information system".

The concept of information system is a product of the general systems theory, developed in the 1940s by Van Bertalanffy and developed in the 1950s, notably in France by Jean-Louis Le Moigne. What characterises the systemic approach can be seen as a reaction to an analytical approach typical of the Cartesian logic and it should be noted that systemic stresses on the concept of totality.

For example, According to Saussure, a system is an organised whole, made of elements that can only be described relatively to each other, according to their place in this totality. According to Von Bertalanffy, it is a "set of units in mutual interrelations". Lesourne to his side talks of a "set of elements linked by a set of relations."

In this course, we consider a system as a set of entities in interactions, achieving a specific goal.

The systemic concept distinguishes systems and subsystems. For example in a car, the engine will be a subsystem. In the human body, the digestive system is a subsystem.

1.3. The systemic

To examine the infinitely small and the infinitely large, scientists use tools like the microscope and the telescope that have led to significant leaps of knowledge. Today, we are confronted with the **infinitely complex** and there are no tools to study the tremendous complexity of the systems we are elements and particles. The systemic approach aims to develop a symbolic instrument called **macroscope**, to study this complexity. However, the systemic approach does not pretend to explain everything or to solve everything and is not intended to provide models of the world that claim to encompass everything.

The natural tendency of the human spirit is unifying, reductive, approximating; It gives apparently satisfactory model, but dangerous: one filters and eliminates all what is different from the unified model and comes out with the worst intransigence. The systemic models are starting points for reflection; they must be confronted with reality, they must be attacked, destroyed, to be better rebuilt because they can **evolve** only in **confrontation**.

1.4. Some methods of analysis

- In computer science analysis, there are several methods organised in three main categories, notably: Relational oriented models including MERISE, SA, SART SADT;
- Functional oriented models including FAST, GRAFCET, APTE, RELIASEP;
- Object oriented models including OMT, UML.

1.4.1. Functional oriented models

1.4.1.1. The method FAST (Function Analysis Systems Techniques)

General principle: The analysis method FAST is a graphical communication instrument among stakeholders. It focuses on the relationships between products and functions. This instrument allows representing the logical relationships between tasks by repetition of "why? / How? / When?" posed at each stage of the analysis. FAST presents the relationship between products and functions within a strictly delimited domain.
Originalities: FAST produces a diagram that allows the designer to explain and justify the technical solutions.

1.4.1.2. The method APTE (APplication des Techniques d'Entreprise)

General principle: The APTE method is a set of methodological tools available to project teams to control the consistency between the various elements of a project, to make performed and reliable technology choices to satisfy the needs at lower cost. This method provides common logics of formalisation for conceptual phases of complex projects.
Originalities: The APTE method is very systemic-oriented in the sense that the product has been designed from its surrounding environment, favouring the point of view of the designer and the user. APTE has its own internal validation tools (need, functions, constraints...). APTE offers an effective method of group animation.

1.4.1.3. The method GRAFCET

General principle: The GRAFCET method proposes a graphical representation of awaited behaviour of a logical system. "Make a GRAFCET" consists of describing sequential ongoing, starting from functional specifications of an automated equipment. A GRAFCET allows to describe the "command" part of the automated system.
Originalities: The GRAFCET method is used only for automatism. The GRAFCET language is used to program programmable automates.

1.4.1.4. The method RELIASEP

General principle: From the need expressed in terms of functions, the RELIASEP method finds recursively the sub-functions required to satisfy the need. This process generates the so-called functional tree. This research is progressive and adapted to the development phases.

Originalities: Very oriented operating security, the functional tree of the RELIASEP method is used as basis to analyse failure modes, their effects and their criticality. This method allows visualising the functional path of deterioration. RELIASEP is effective for the study of complex systems more material-oriented than software-oriented.

1.4.2. Relational oriented methods

1.4.2.1. The method SA/RT

General principle: From an analysis based on relevant questions and the graphical representation of this analysis, the SA/RT method allows the development of:

- A model of the analysed system;
- A model of the static process attached to it;
- A dynamic control model that will help for the method use.

Originalities: This method considers the dynamic aspect of the analysed system. It is based on an old method (Structured Analysis -SA) that had been widely used.

- SA / RT is well suited for applications with high dynamic behaviour.
- SA / RT is rather focused on sharing the work and on human-machine interface.

1.4.2.2. The SADT method

Definition: The acronym SADT stands for System Analysis and Design Technic. This method has been developed by the Softech Company in the United States. SADT is a method of analysis by successive levels of descriptive approach of a given set.

General principle: The SADT method consists in modelling existing or future systems to understand the functioning and to envisage adequate solutions. This modelling is based on the actions of the analysed system, called actogram, and on the data the system can manage, called datagram, in an arborescent structure of the system.

Originalities: Rather oriented information system, softwares and automatisms, SADT defines two types of stakeholders: the author who conceives the system and the reader who criticises. These works allow to precise and to optimise the system by iterative approach.

SADT allows a good traceability and internal controls of the coherence. It can be summarised as:

- A graphical tool of representation;
- It obliges to record the decisions of the work team. This progressively allows to create a complete documentation of the system.
- It's a team effort that requires discipline and coordination. Indeed, SADT is a product for communication and diffusion.
- Its formalism leads to and ascendant or descendant structured representation.
- If SADT is completely used (actigram and datagram), it allows to program directly an automated system.

1.4.2.3 The method MERISE

General principle: The MERISE principle consists in: (1) representing an existing system by easiest invariant elements (sub-systems of order n) which compose the system, (2) optimising links and relations required for the functioning of the system.

Originalities: MERISE is very information system and software-oriented. The next chapter is focus on the MERISE method.

Chapter II. Study of a modelling method : MERISE

2.1. Introduction

The conception of an information system is not obvious. Indeed, one must think about the whole organisation to be put in place. The conception phase requires methods allowing building the basic model. The modelling consists of creating a virtual representation of a reality in order to highlight the interesting points. This kind of method is called *analysis*. There are several methods of analysis and MERISE is one of the most used.

2.2. Presentation of MERISE

2.2.1. Definition and generalities

MERISE is a method of conception, development and realisation of automated systems. The purpose of this method is to conceive an information system. MERISE is based on the separation of data from treatments to be done in several conceptual and physical models called abstraction cycle of MERISE.

The separation of data from operations insures longevity to the model. Indeed, the arrangement of data doesn't often need to be reshuffled, while operations or treatments are being frequently.

Regarding the company, MERISE distinguishes 3 sub-systems:

- ❖ Steering system;
- ❖ Information System;
- ❖ Operating system.

Where **operating systems** concerns basic operations of the enterprise (production of goods for sale for example) and the **steering system** points to the organisation and decision instances. Between the two, MERISE situates the **information system**.

2.2.2. Abstraction cycle of Information System conception

The conception of information system is done by steps, in order to lead to a functional information system reflecting a physical reality. The purpose is to validate one by one each step, considering the results of the previous steps. Since data are separated form operations, it is required to verify that all data required by the system are present and that there is no superfluous data. This succession of steps is called “abstraction cycle” for the conception of information system:

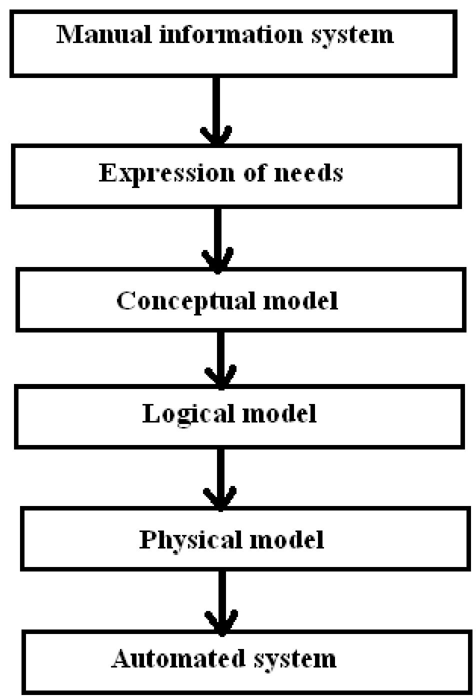


Figure1. Abstraction cycle of MERISE.

2.2.2.1. The Manual information system

The manual information system is a set of manuscript and oral information used for the management of the structure. This information is generally presented in the form of forms to fill or in the form of report to interpret. In the analysis process, this step consists in realising interviews beside different services managers or beside system managers and in collecting samples of each type of information handled in the system.

2.2.2.2. The Expression of needs

This step consists in defining what is expected from, and what is required by the automated information system. Therefore, it is required to:

- Do the inventory of required elements for the information system;
- Identify the limits or difficulties of the system management;
- Delimit the system by getting information beside the future users;
- Define the expected functionalities of the automated system.

2.2.2.3 The Conceptual model

This step consists in realising different conceptual models among which the fundamentals are: the Conceptual Model of Data (CMD), the Operating Conceptual Model (OCM) and the Operating Organisational Model.

The CMD is the model on which are based the data of the system. It describes the architecture of the database and presents the rules and constraints to consider while building the database.

After the expression of needs, the specification of the Conceptual Model of Communication (CMC) which defines the information flows to consider can be done. These information flows start from an actor to another actor of the system and specify the nature or the type and the orientation of the transmitted information. The Operating Organisational Model (OOM) describes the constraints due to the environment (organisational, spatial and temporal).

2.2.2.4 The Logical Model

The Logical Model represents a software choice for the information system. It is deduced from the CMD. It helps in writing queries to access databases. It presents the database in the relational form and expresses the associated functional dependences in the form of relations.

2.2.2.5 The Physical Model

The physical model of an information system reflects the matter choice (for example, Ms Access, Oracle, Paradox, Mysql, etc.) for the information system. It presents the organisation of the created tables and the different relations between them.

2.3. The Conceptual Model of Communication (CMC)

2.3.1. Definition of the organisation

The first step of this model is to isolate the system by delimiting its domain. The purpose is to define the system and the external elements which exchange information flows with the system. These external elements are called external stakeholders/actors or partners. Information exchanged can be either in only one direction (from external actor to the system exclusively or from the system to an external actor exclusively) or in double direction (simultaneously from the external actor to the system and vice versa).

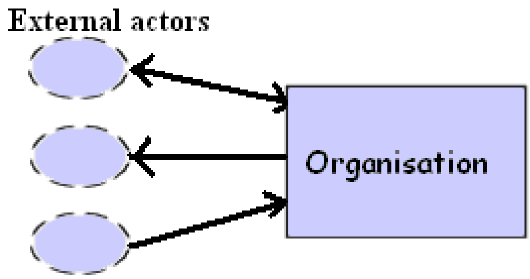


Figure 2. Context diagram formalism.

The second step consists in cutting up the organisation into entities called internal stakeholders/actor (or domaine). When the domain of an organisation is very large, it can be cut up again into sub-domains.

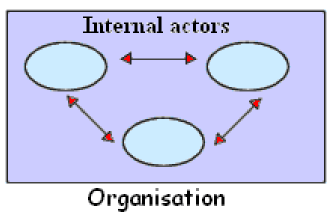


Figure 3. Cutting-up the system domain.

- The last step is the analysis of the information flows, that is, the definition of process.

2.3.2. The Context Diagram

The context diagram aims to represent the information flows between the organisation and the external actors accordingly to a standardised representation in which each object has a name. Notably:

- ❖ The organisation is represented by a rectangle,
- ❖ External actors are represented by ellipses in dotted line,
- ❖ The information flow is represented by an arrow which orientation specifies the direction of the information flow.

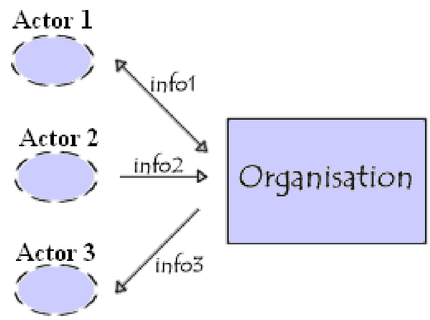


Figure 4. A context diagram of a system.

No internal actor is involved in this diagram. In this diagram, the external environment of the organisation is made of three actors, namely Actor 1, Actor 2 and Actor 3.

2.3.3. Conceptual Diagram of information flow

This diagram (also called Conceptual Model of Communication) helps to complete the context diagram by breaking down the organisation into a series of internal actors. In this diagram, the standard representation is as follow:

- ❖ Internal actors are represented by ellipses in continuous line;
- ❖ Internal or external messages are represented by oriented arrows;

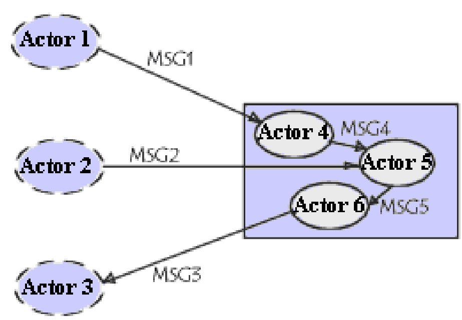


Figure 5. Conceptual Model of Communication formalism.

This diagram can be put in a matrix form called **information flow matrix.** The structure of the said matrix is the following:

	Actor 1	**Actor 2**	**…………**	**Actor n**
Actor 1				
Actor 2				
…				
Actor n				

Each cell localized by the line of an actor J and the column of an actor I specifies the information sent by the actor J to the actor I. If the cells of the matrix are symmetrically filled then all the arrows of the CMC are bidirectional.
The diagonal is black-colored because an actor cannot send an information to himself.

2.4. The Conceptual Model of Data (CMD)

2.4.1. The purpose

The CMD aims to describe formally the data to be used in the information system. It is question here of a data representation, easily comprehensible, allowing describing the information system using entities.
This conceptual model is the main purpose of this course. Each student, by the end of this course, should be able to build the conceptual model of data of any information system.

2.4.2. Entities and class of entities

An **entity** is the representation of a material or immaterial element having a role in the information system that one wishes to describe. It is anything, material or logical that one can manipulate.
A **class of entities** is a set of entities of the same type, (having the same definition). An entity is an instantiation of the class. Each entity is composed of properties, elementary data describing the entity. A class of entities is an abstraction description of properties that all the related entities have in common.
Let us consider the example of a Toyota Advensis, a Mercedes ML and a Renault Megane. These are three entities belonging to the class "*Car*". The Toyota Advensis is then an instantiation of the class "*Car*". Each entity can possess the properties: colour, year, number of seats and model.
A class of entities is represented by a rectangle separated into two fields as shown in the following figure:

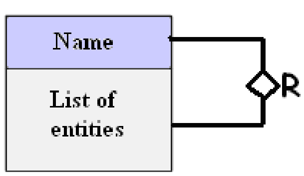
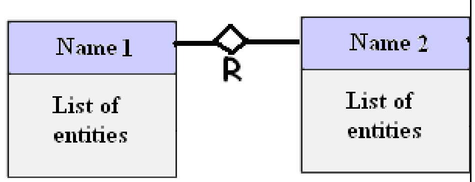
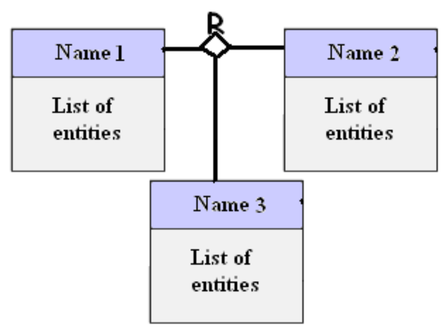
Figure 6. Formalism of a class of entities

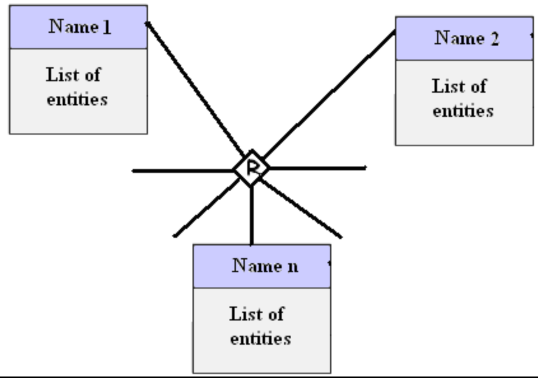
The upper field contains the name of the class. This name is generally an abbreviation for a matter of writing simplification. The question here is to verify that, to each class of entities corresponds one and only one name and vice versa, for a given system.

The lower field contains the list of properties of the class.

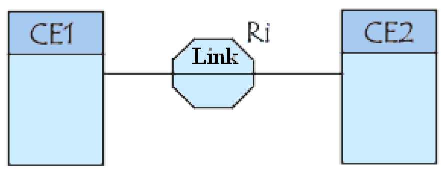
2.4.3. Relations and class of relation

A *relation* (also called *association*) represents the semantic links that can exist between several entities. A *class of relation* contains relations of the same type (It links entities belonging to the same class of entities). A class of relation can link more than two classes of entities. According to the number of participants in a relation, we have the following denominations:

		
A recursive (reflexive) class of relation links a class of entity to itself.	A binary class of relation links two classes of entity	A ternary class of relation links three classes of entity

		
	An order n class of relation links n classes of entity.	

Classes of relation are represented by *hexagons* (at times by *ellipses*). The name (generally a verb) describes the type of the relation that links the classes. For each class of relation, an identifier R_i is defined. This identifier defines one and only one class of relation.

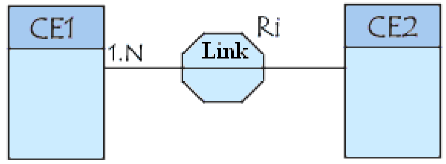


One can eventually add properties to classes of relation. These properties are called *association properties*. These properties cannot belong to any class of relation taking part to the association. For example, the mark is an association property between the class of entity *Course* and the class of entity *student*.

2.4.4. The cardinality

Cardinalities characterise the link between an entity and an associated relation. The cardinality of a relation is a couple made up of a minimum limit and a maximum limit defining the interval in which the cardinality of an entity can take its value:

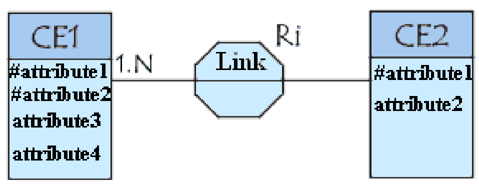
- The minimal limit (generally 0 or 1) describes the minimum number of times that an entity can participate to the associated relation.
- The maximal limit (generally 1 or n) describes the maximum number of times that an entity can participate to the associated relation.



A cardinality 1.N means that each entity belonging to a class of entity participates at least one time to the associated relation (link).
A cardinality 0.N means that each entity belonging to a class of entity does not absolutely participate to the associated relation (link).

2.4.5. Identifiers

An identifier is a set of properties (one or several) describing one and only one entity. The original definition is the following:
The identifier is a particular property (set of properties) of an object so that it's impossible to find two occurrences of the said object for which this (these) property(ies) should have the same value.
Attributes of a class of entity identifying one and only one entity are called *absolute identifiers*. The MCD formalism proposes to precede identifiers with # or to underline them.



Then, each class of entity must own at least one identifier attribute and the whole set of its identifiers attributes must be field during the creation of the entity.

2.4.6. Aggregation (or relative identification)

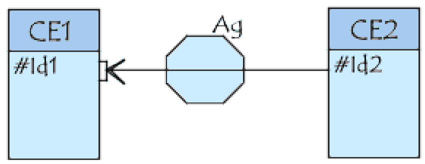
When an entity identifier is constitute only of intrinsic attributes (attributes that doesn't refer to another entity), it is called *absolute identifier*. Entities made up of absolute identifiers can be defined independently to other entity occurrences. In this case, entities are *independent*.

However, some entities can only be identified through others. This is why we talk of relative identifier. For example, one can talk of the 4th door of the 2nd floor of the building B instead of door N°3451…

Thus, an aggregation specifies that an entity is required for the identification of another one.

- The class of entity required for the identification of another is called the aggregator class.
- The class identified is called the aggregated class.

The representation of this type of association is as follow:



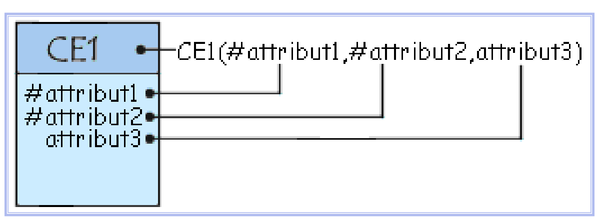
2.5. The logical model of data (MLD)

2.5.1. Description

The logical model of data consists in describing the structure of data used, without however referring to a particular programming language. The purpose is then to precise the type of data used during the process. Thus the logical model depends to the type of data base used.

2.5.2. Translation of a class of entity

Each class of entity of the conceptual model becomes a table in the logical model. The identifiers of the class of entity are called keys of the table while the standard properties become attributes of the table. Each attribute defines a column in the table.

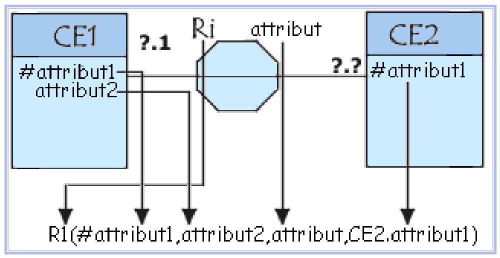


2.5.3. Translation of a class of relation

The passage from the conceptual model to the logical model related to the classes of relation is done accordingly to the cardinalities of each entity associated to the association.

If a class of entity possesses a weak cardinality, the table should have as attributes the properties of the class with weak cardinality, the properties of the relation and finally the primary key of the other class.

If the two classes of entity possess high cardinalities, the table should have as attributes, the primary keys of the two classes and the properties of the association.



2.5.4. Translation of an aggregation class

In the case of an aggregation class, the aggregated entity class has as additional attribute, the primary key of the aggregator class.

2.6. The Operating Conceptual Model (OCT)

2.6.1. Introduction

The operating conceptual model allows studying the dynamic of the information system. This model allows representing graphically the activity of the information system without considering organisational choices or execution means. It defines simply what is supposed to be done without specifying neither the date, nor how, nor the place.

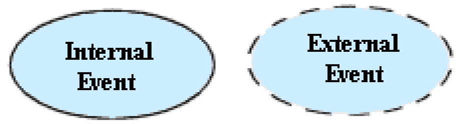
2.6.2. The concept of event

An event represents a change in the external universe of the Information system or in the information system itself.

- An external event is a change of the external universe of the information system

- An internal event is a change in the information system

Events are represented by ellipses in continuous line for internal event and dotted line for external event.



2.6.3. Process

A process is a sub-set of the entreprise activity. Indeed, the activity of the enterprise is made up of a set of process. A process itself is made up of *operations*.

2.6.7. Operation

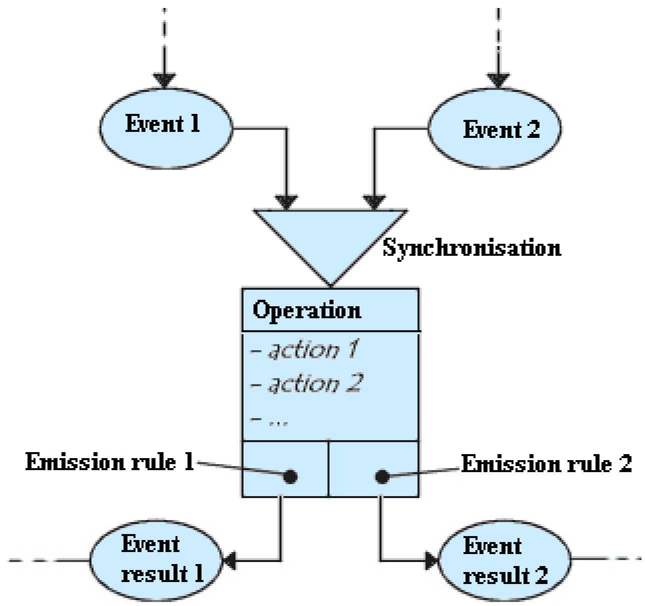
An operation is a set of actions proceeded by the system after an event or a conjunction of events. This set of actions is uninterruptible until the operation is accomplished.

2.6.8. The synchronisation

The synchronisation of an operation defines a Boolean condition on events that contribute to trigger the operation.

2.6.9. The construction of the operating conceptual model (MCT)

The operating conceptual model allows representing graphically the management of events.



2.7. The Operating Organisational Model (MOT)

2.7.1. Description

The operating organisational model describes the properties of operations not considered by the conceptual model of data, notably the time, resources and the place.
The MOT consists in representing the MCT in a table which specifies in its columns the duration, the place, the personal chair and the resources required by an action.

2.7.2. The table of functional procedures

The first step of the MOT consists in cutting up operations into functional procedures (FP), a succession of operations triggered by an event. The purpose is to associate in a table:

- The functional procedures,
- The beginning and the ending time,
- The place of the workstation,
- The personal chair of the workstation,
- The resources of the workstation

Procedure	Time		Workstation		
	Begin	End	Place	Personal chair	Resources

2.8. The physical model of data

This step consists in implementing the model in a specific System of data Base Management. This means to translate into a specific definition language. The commonly used language for this type of operation is the SQL (Structured Query Language).

2.9. Exercises

2.9.1. Automated management of a library

You have to create a data base for a library. The librarian asks you to be able to manage the following information: Books characterised each by a title, the authors (name and surname), an editor, an edition date and the kinds (for example: love, fiction science, politic, roman, news,…) and the customers (name and surname) each with an address (street, postal code and town), the books borrowed, the expected date for book return and the effective date of book return.
You don't know more and you are asked to do reasonable hypothesis and choices required by the system. Conceive then the data base.

2.9.2. The management of an urban transport company

You must create a data base for a bus company. The company manager gives you the following indications: each bus is numbered. It realises ways made of a departure, several stops and a terminus. Each bus can every day be allocated to several ways and it can do many journeys on each way with eventually distinct drivers. The management of customers is done separately (in another data base not concerned for this project) but the management of buses (date of the next revision, the mileage, number and name of the depot,…) and drivers (name, surname, grade, seniority, salary, buses droved, ways done,…) is included in this data base.

Chapter III. EXAMPLE OF DATA BASE CREATION

3.1 Problem

3.1.1 Specification

Let consider the case of realisation of the data base of the students of the national higher school of applied statistics and economy (ENSEA) of Abidjan in Côte d'Ivoire. In this data base, we are called to manage notably: (1) students characterised each by a registration number, a name, a surname, the year of first registration, the date of bird and the sex; (2) courses characterised each by au code, a name, a level and the hourly quota; (3) marks of students, teachers characterised each by a code, a name, a surname, a speciality, a grade and a sex; (4) the option (Mathematic, Economy,…), the category (ISE, ITS,…), the level and the function (delegate or not) of each student; The constraints are as follow.

1. The registration code belongs to one and only one student.
2. The registration code is sufficient to determine all other characteristics of the student.
3. The mark depends on the student, the course and the year.
4. Each course has only one principal teacher.
5. The same teacher can be principal teacher for many courses, independently to the level of the course.

3.1.2 The expected result.

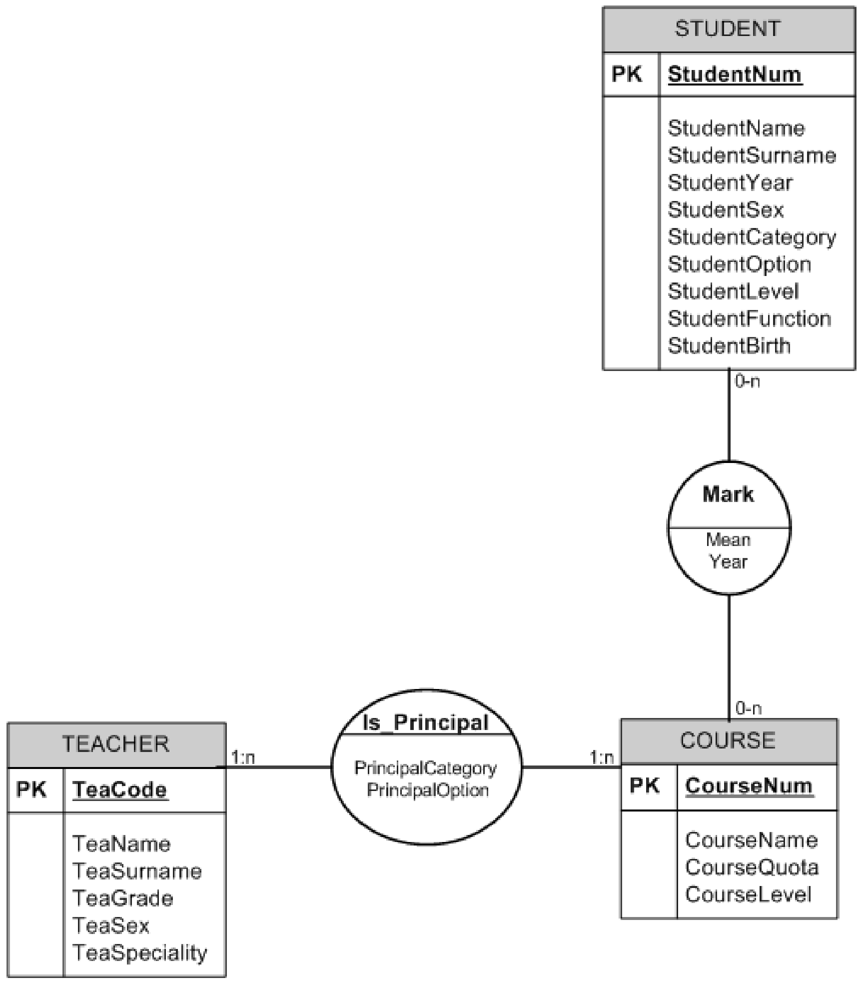
The data base should produce answers to the following preoccupations:

- Extract all female students who have a mark greater than 12 in Information System in 2009.
- Extract all the principal teachers for the course Data base since 2007.
- Extract and order by level and by option all the delegates.

3.2 Solution

3.2.1. Analysis

One can denote that there are three entities, notably: student, teacher and course. The related conceptual model is the following:



<u>Figure 2:</u> The conceptual model of data (MCD)

The equivalent **logical model** of data is the following:

Student (**<u>StudentNum</u>**, StudentName, StudentSurname, StudentYear, StudentSex, StudentCategory, StudentOption, StudentLevel,StudentFunction, StudentBirth)
Teacher (**<u>TeaCode</u>**, TeaName, TeaSurname, TeaGrade,TeaSex, TeaSpeciality)
Cours (**<u>CourseNum</u>**, CourseName, CourseQuota, CourseLevel)
Mark (**<u>CodeMark,</u>** StudentNum,CourseNum, Mean,Year)
Principal (**<u>CodeTeach,</u>** TeaCode, CourseNum, PrincipalCategory, PrincipalOption)

BIBLIOGRAPHY

[1] Jean-Claude. COURBON, 1997, Dunod, "Systèmes d'information : structuration, modélisation et communication",288 pages, ISBN : 2729604227.

[2] André MULLER, 2005, Presses Universitaires de France (PUF), "L'informatique dans l'entreprise", 127 pages, 2ème édition, ASIN : 2130445926.

[3] Steven ALTER, 2001, Prentice Hall, "Information Systems: Foundation of E-Business", 587 pages 4ème édition, ISBN: 0130617733.