Chapter IV: IS development methodology (MERISE)

1. IS development process

The IS development process is a structured set of activities that aim to achieve the objectives of an IS project set by the organization. These activities vary depending on the type of organization, the type of project and the type of system to be developed. This process must be clearly described in order to be managed correctly.

2. Software life cycle

The life cycle of an IS succinctly describes the phases through which an information system passes from the initial need to the retirement of the system.

3. IS development activities

The development of an information system involves several activities which vary depending on the type of project and the organization, the essentials of which are:

- **Specification** system requirements and constraints, establishment of specifications
- **Design** of the solution, production of a model of the system to be developed Implementation of the system
- **Test** of the system, verification of the adequacy between the implemented properties of the system and the specification of requirements
- **Facility** of the system at the customer and verification of its operation
- **Maintenance** of the system, repair of faults

4. IS development models

There are different models used in the creation of software. These models aim to apply the development activities cited above with a certain organization between these activities.

Among the best-known models, we cite: the Waterfall model, the V model, the spiral model and the incremental model. The application of these models can involve the use of an analysis and design method. Each method has its advantages and disadvantages and each method is adapted to a type of project (industrial, management, scientific, etc.). Among the existing methods we cite:

MERISE, SADT, SART, OMT and UML (although UML is not a method but a unified modeling language)

5. The MERISE method (in french : Méthode d'Étude et de Réalisation Informatique pour les Systèmes d'Entreprise)

Merise is a systemic method which appeared in 1979 as a result of a project launched in 1977 by the French Ministry of Industry whose aim was to provide companies with a design method which would enable them to succeed in their IT projects in expected costs and deadlines.

5.1. Characteristics of Merise

The major advantages of Merise are:

1. A global IS approach carried out in parallel on data and processing.

2. A description of the IS using a simple and rigorous formalism standardized by ISO for the presentation of data (Entity Association Model).

3. Separation of data and processing.

5.2. Cherry abstraction levels

5.2.1. Conceptual level

The conceptual level aims to answer the question WHAT? That is, what to do? with what data? without taking into account the organization of work and the equipment used. The two model results of this level are:

- The conceptual data model (CDM)
- The conceptual treatment model (CTM).

5.2.2. Organizational level

The organizational level aims to answer the questions WHO? WHERE? and WHEN? At this level, we integrate the organizational criteria of the work. We take into account (or propose) the distribution of processing between man and machine, the mode of operation (real time, deferred time). The resulting models from this level are:

- The logical data model (LDM).
- The organizational treatment model (OTM).

5.2.3. Operational (physical) level

It aims to fix the results of technical decisions taken according to technical objectives and constraints. It consists of answering the question HOW?

We study technical solutions (storage mode for data, division of programs for processing. The resulting models at this level are:

- The physical data model (PDM).
- The physical model of treatments (PMT).

The abstraction levels with their resulting models are summarized in the following table:

LEVEL	DATA	TREATMENTS
Conceptual *Management choice: What?*	Conceptual Data Model (CDM)	Conceptual Model of Treatments (CTM)
Organizational *Organizational choice: Who? Where? When?*	Logical Data Model (LDM)	Organizational Model of Treatments (OMT)
Physical *Technical choices: How?*	Physical Data Model (PDM)	Physical Model of Treatments (PMT)

5.3. Breakdown into stages

MERISE recommends dividing the project into four stages. This division is not specific to the Merise method, but it is generally recommended for carrying out any IT project. Each of the steps corresponds to a level of abstraction

There are five of these steps:

1) Preliminary study / master plan

2) Detailed study

3) Realization

4) Implementation

5) Maintenance

Breakdown of the MERISE method into stages
Master plan / Preliminary study (study of the existing)
Detailed study (in parallel by two teams if possible) CDM: Conceptual Data Model CTM: Conceptual Treatment Model OMT: Organizational Model of Treatment External views / Validation MLD: Logical Data Model
Realization(together) PDM: Physical Data Model PMT: Physical Model of Treatment
Implementation
Maintenance

6. Preliminary study (Study of the existing)

The objective of this phase is to:

- Learn in detail about the area for which new automation is planned.

- Identify the exhaustive set of objectives that the company is pursuing regarding this area.

- List management, organizational and technical constraints.

6.1. Collection of the existing:

All of what exists can only be collected from two entities which are:

“Management” and “workstation”. The technique used is naturally that of the interview.

It can be supplemented by questionnaires, surveys, documents, etc.

6.1.1. Management interviews:

- First knowledge of the problem posed
- Identify the requested objectives

- Identify the main workstations involved
- Define interfaces with other projects
- Delimit the field of study

6.1.2. Job interviews:

- Identify and describe the tasks performed
- Observe the flow of information
- Learn the business language.

6.2. Tasks and formalisms accompanying the study of the existing

During the existing study phase, we must carry out the following tasks:

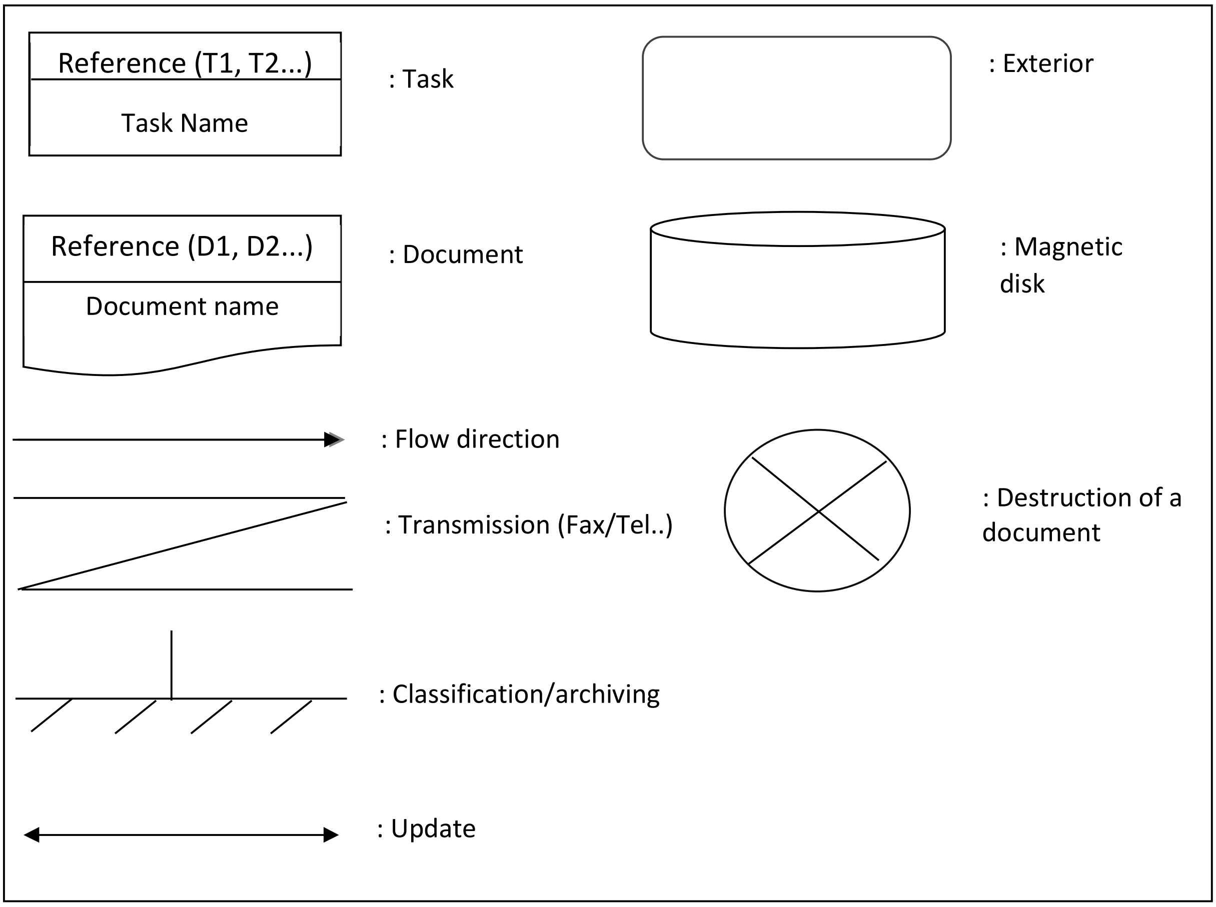
- Interviews visualized by task-document diagrams
- Establishing a flow graph
- Census of rules (management, organization and technique)
- Establishment of a data dictionary.

6.2.1. Task-document diagrams

Over the course of the interviews, we will construct task-document diagrams. These will visualize the sequence of tasks through the documents that trigger them and those that they produce.

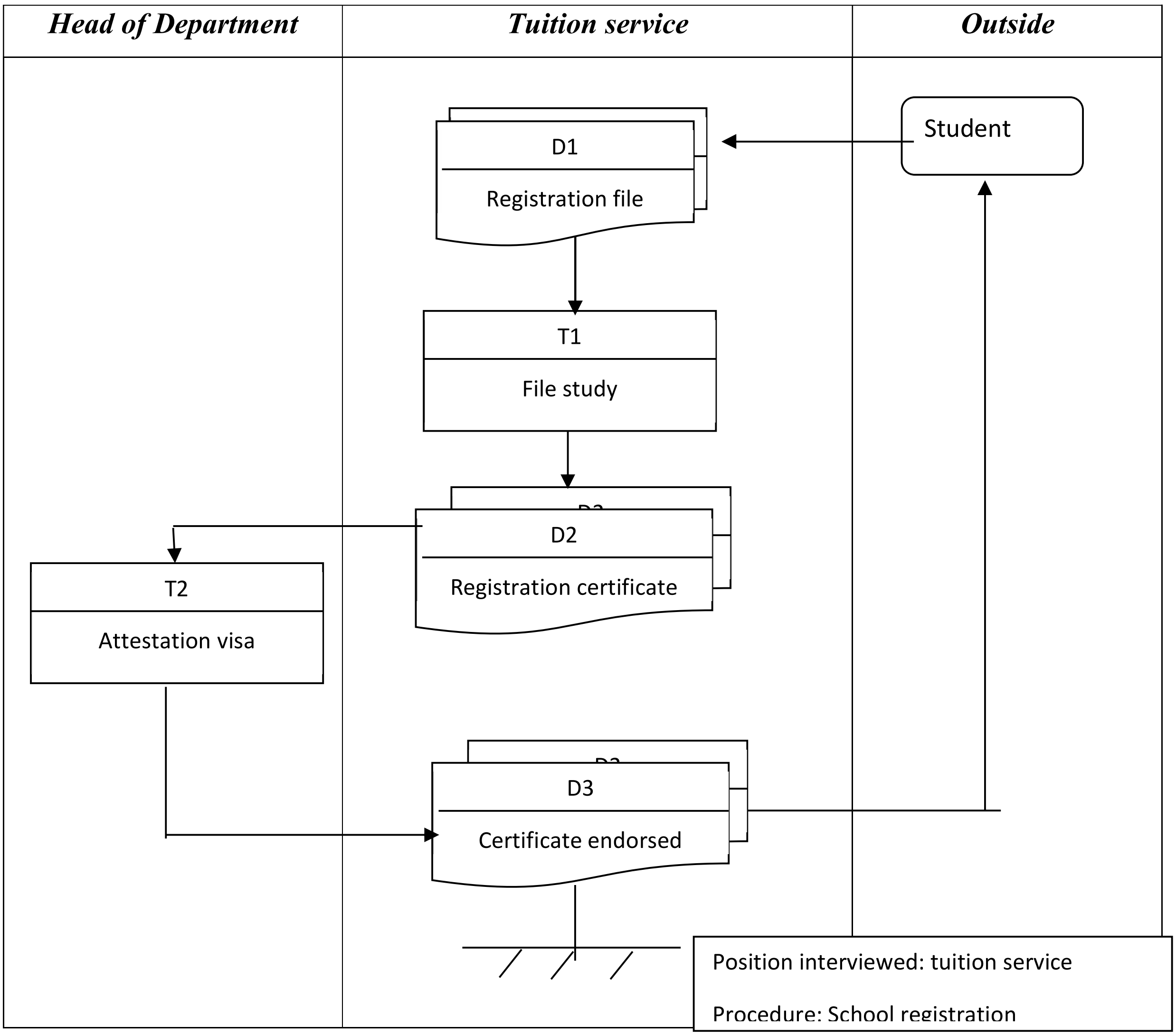
To simplify the diagrams and to put specific remarks related to the set of tasks and documents, we reference the latter to describe them separately.

The symbolism used in task-document diagrams is as follows:



For the organization of the diagram, we put the external actors (if they exist) in the rightmost column, the central column will be reserved for the position concerned (interviewed), and the leftmost columns will be reserved for actors who are part of our field of study (if they exist).

Example: Registration *at school level*



Document number	***Wording/role***	***Task numbers***
D1	Registration file: contains the documents necessary for registration such as….	T1
D2	Registration certificate: this is a visa-free certificate	T1, T2
D3	Certified certificate: registration certificate signed by the head of department	T2

Task number	*Description of the task*	*Workplace*	*Frequency and volume*	*Input document*	*Output document*
T1	Study of registration file. Check if the student's documents are compliant and complete	Tuition service	At the start of the academic year	D1	D2
T2	Visa of the certificate. Before signing the certificate we must check its conformity.	Head of Department	At the start of the academic year	D2	D3

6.2.2. Flow graph

The objective of this diagram is to show the circulation of documents between the different actors. This graph is already described in section 14 (page 17).

6.2.3. Census of rules

The execution of tasks and the manipulation of data within a company are always governed by a certain number of rules. In this phase, it is therefore a question of identifying the rules whose expression is often vague and/or confused in the description of the tasks or on the documents collected and relating them to the different levels of abstraction. There are different ways of expressing rules:

- Literal in everyday language
- By mathematical formulas
- By pseudo code
- With decision tables, flowcharts, etc.

There are three types of rules which are:

- Management rules.
- Organizational rules.
- Technical rules.

Management rules:

They describe the “what” of the company, that is to say they translate the objectives chosen and the constraints accepted by the company. They express the actions that must be accomplished and the regulations attached to these actions (laws, regulations, calculation rules, etc.). Management rules are of two types:

- **Action rule** which describes the actions that the company must perform.
  Examples:
  - Any product delivered will be entered into stock
  - A student cannot register in 2 different levels
- **Calculation rule** which describes how actions should be carried out.
  Example: Base salary = index x number of points.

Organizational rules:

They describe the "where", the "who" and the "when", that is to say they reflect the organization put in place in the company to achieve the fixed objectives.

Examples:

-Orders should not be placed on Friday.
- the closing of the cash register must be validated by the cashier every day at 4 p.m.

Technical rules

They describe the “how” that is to say they translate the technical conditions for implementing the tasks.

Example:

- The capacity of auxiliary memories will be at least 20 Gigabytes.

6.2.4. Data Dictionary

The DD represents all the properties appearing on the documents used by the different workstations in our field of study. The model used in establishing a DD can be as follows:

Coded	**Meaning**	**Type**	**Length**	**Syntactic rules**	**Semantic rules**
Name	Student Name	A	20		
Pre	Student first name	A	20		
Date_N	Date of birth	D	10	DD/MM/YYYY	
Avg_	Student Average	N	5	XX,XX	$0 \leq Avg \leq 20$
………					

Dictionary purification

After establishing the dictionary, it must be purified by eliminating synonyms and polysemes.

Synonyms: Different names designate the same reality.

Ex: - Order_Num and Order_Ref.

- Agent and employee.

Polysemes: The same name designates 2 or more distinct realities.

Ex: - Num_C: to designate the customer number and the order number

7. Detailed study

7.1. Conceptual Level

7.1.1. Data: CDM (Conceptual Data Model)

The objective of the MCD ("Conceptual Data Model") is to provide a static representation in schematic form of the data related to the management domain concerned. This model is based around the following concepts:

A. Property (Attribute)

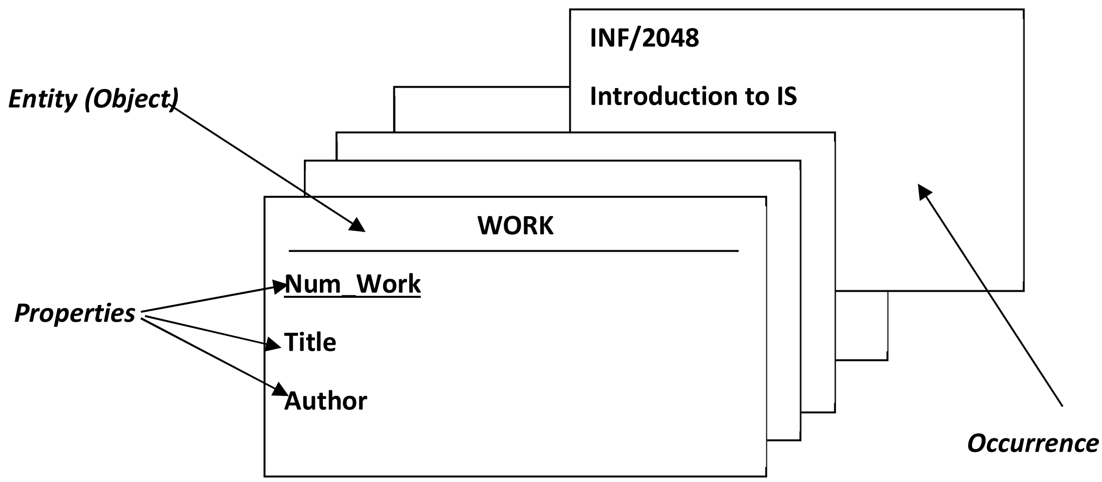
The finest level of data (called "unbreakable"). Which assumes that the base has been purified of polysemes and synonyms [22]. It is a property associated with entities and associations to describe them. The properties must conform to the company's management choices.

Example: Name_employee, Qty_stock, Address, Date_birth…etc.

B. Entity (Object)

Object of the real world (concrete or abstract), about which we want to record information and which has its own existence. It is formed from a collection of specific properties, where each occurrence is unambiguously identifiable using a particular property called an "identifier". Each value of this identifier corresponds to a single occurrence. "For each occurrence, all properties must take one and only one value" [22].

Example :

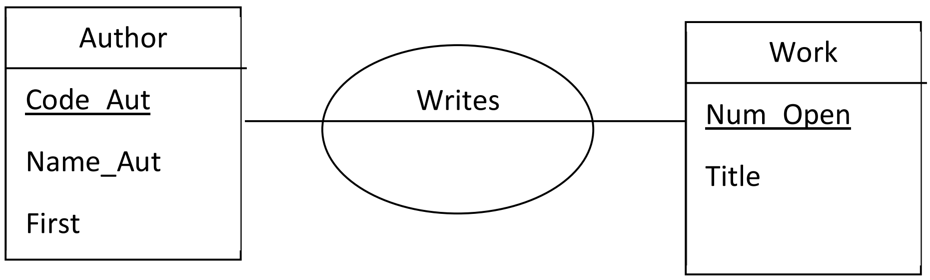


C. Association (Relationship)

Link between two or more entities where each of them plays a certain role in this relationship. Each occurrence of an association is identified by the concatenation of the identifiers of the entities united in this association. Associations are generally symbolized by conjugated verbs.

Examples:

- An author **writes** a work.
- A customer **Pass** An order.

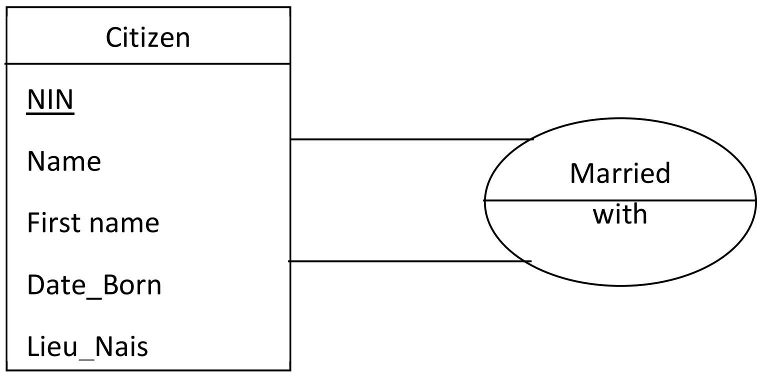


Dimensions of an association:

The dimension of an association is the number of objects (entities) involved in this association. Depending on the size, we can distinguish some particular associations, namely:

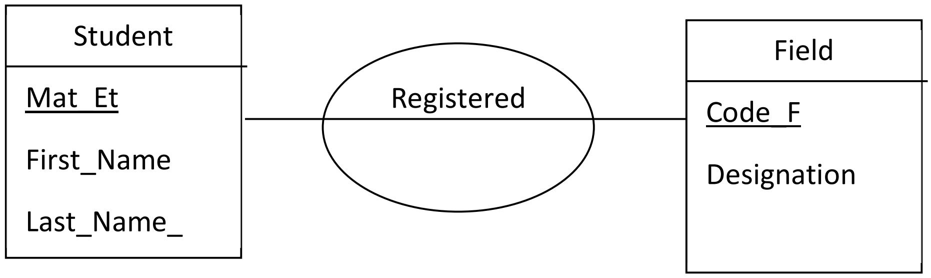
- **Dimension = 1 → reflexive association**

  A reflexive association (of dimension 1) connects an object to itself.



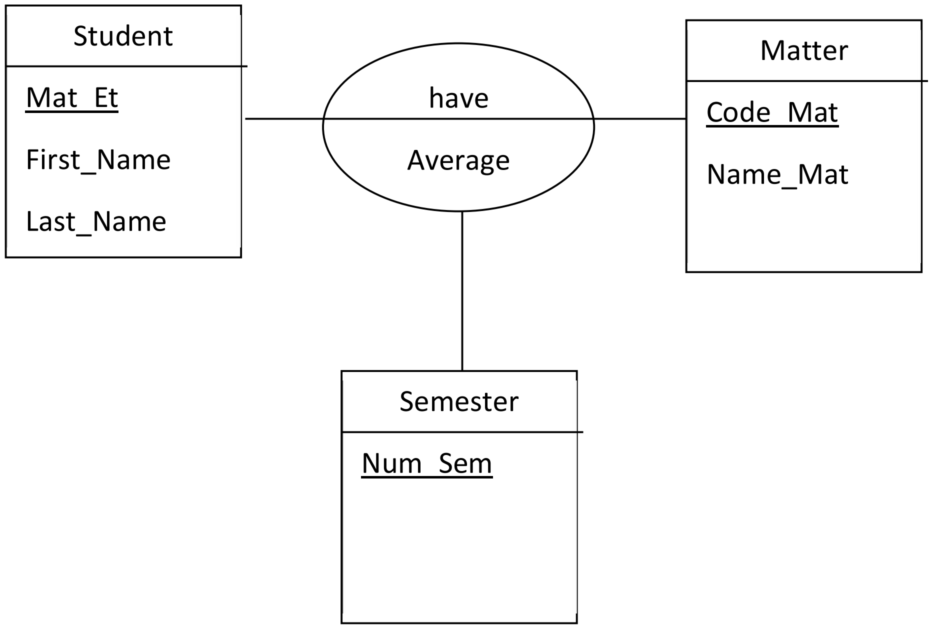
- **Dimension = 2 → binary association**

It connects two entities. This is the most common type of association.



- **Dimension = 3 → ternary association**

It connects three entities.



- **Dimension ≥ 4 → n-air association**

It connects four or more entities. These are rare types of associations whose use should be avoided. Their existence generally reflects poorly done design work. Therefore, it is preferable to defragment them into a set of ternary or binary associations.

D. Cardinality

The cardinalities of an entity measure, when we go through all the occurrences of this Entity, the minimum and maximum participation of an individual (occurrence of the individual) in the association. In other words, the cardinalities of an entity in an association express the number of times that an occurrence of this entity can be involved in an occurrence of the association, at minimum and maximum.

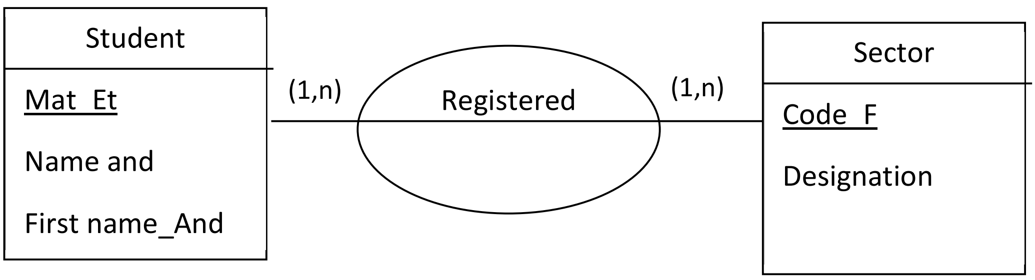
- A minimum cardinality of "0" will express the authorization of the non-participation of some occurrences of the entity in the association.
- A minimum cardinality of "1" will express the obligation of all occurrences of the entity to participate in the association.
- A maximum cardinality of "n" will express the authorization of some occurrences of the entity to participate several times in the association.
- A maximum cardinality of "1" will express the prohibition of all occurrences of the entity to participate several times in the association.

Important

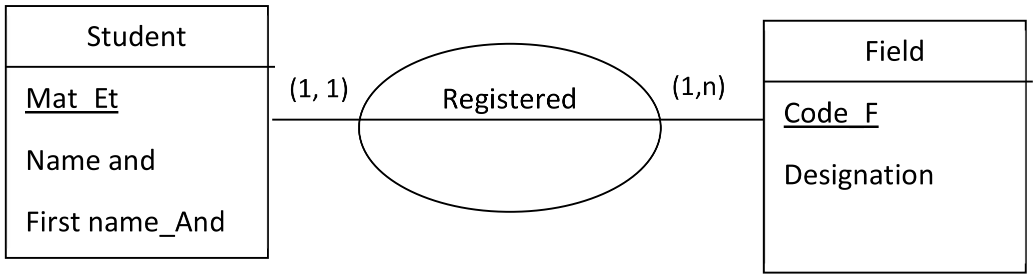
Business rules play a decisive role in determining cardinalities. Therefore, they must be collected and expressed carefully during the preliminary study phase.

Example

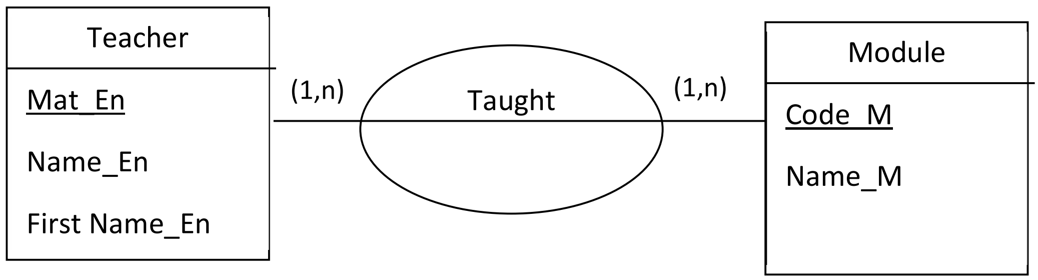
- Management rule 1: A student can enroll in several courses



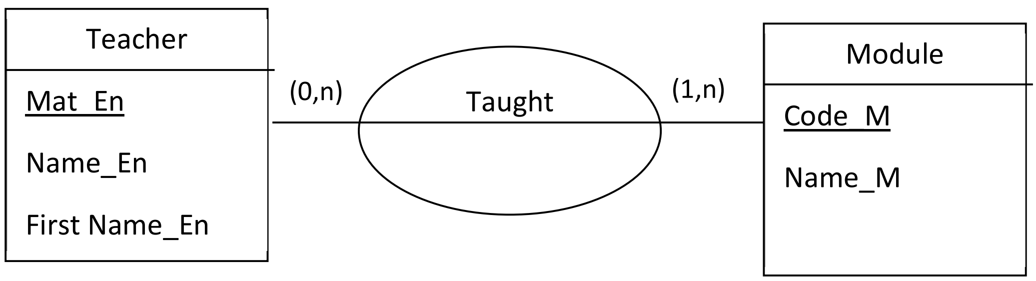
- Management rule 2: A student can only register in one and only one field.



- Management rule 3: A teacher must teach at least one module



- Management rule 4: administrator teachers may be exempt from teaching.



In fact, in the vast majority of cases, we only use 4 combinations of values for the cardinalities.

- 0.1: at most one
- 1.1: one and only one
- 1, n: one or more
- 0, n: zero or more

Example: Medicine distribution company

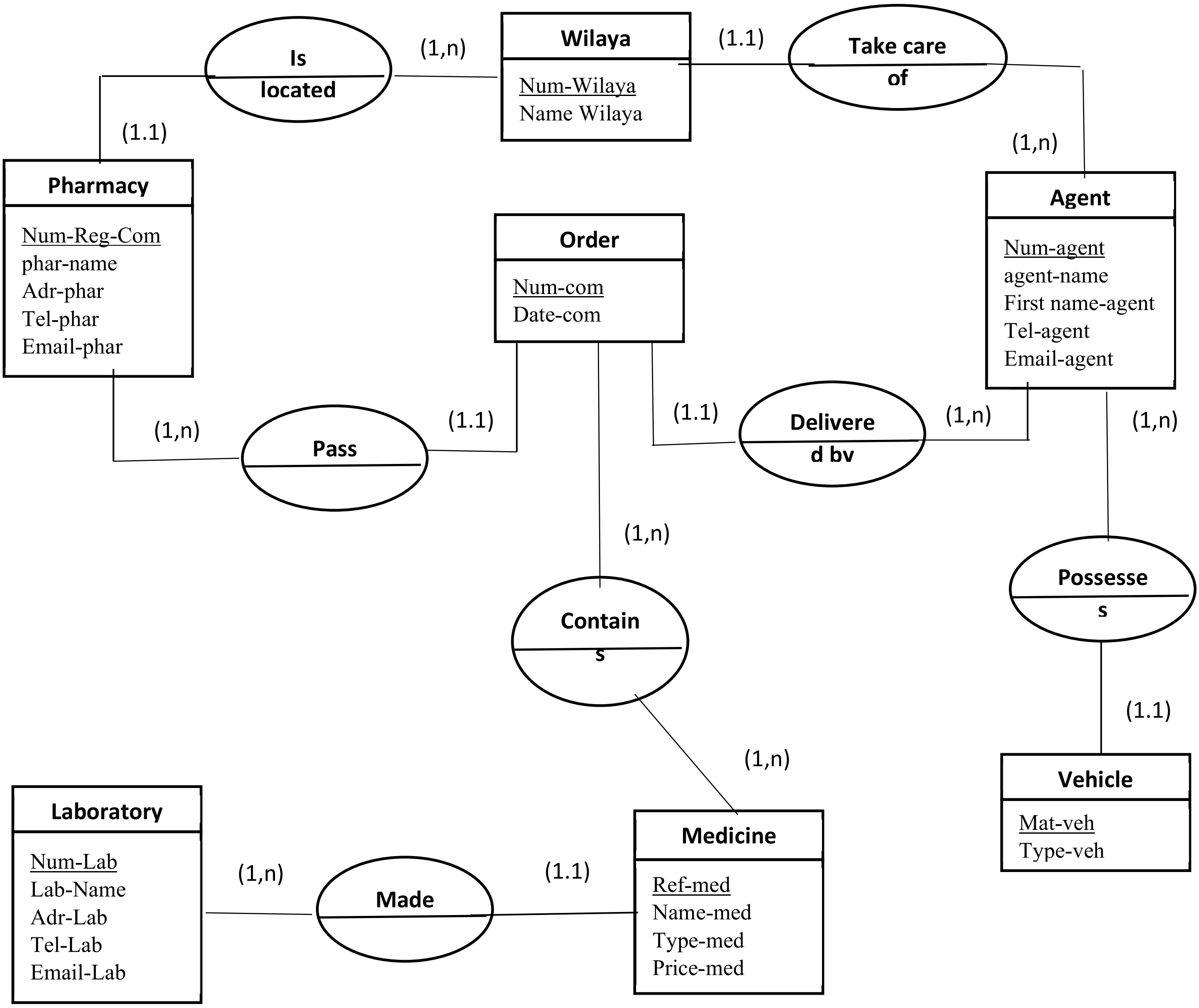
Al-Shifa company is a company specializing in the distribution of pharmaceutical products in different Wilayas. The company takes care of the pharmacies registered there, so each pharmacy is known by its commercial register number, name, address, telephone number and email. The company carries out its work through a group of distribution agents, so each agent is responsible for distributing orders in one or more Wilayas. Each agent is known by a registration number, first name, last name, telephone number and email. Each agent declares the means of distribution they use, which must be proportionate to the nature of the work (commercial cars with air conditioning), so the company must keep the registration and type of each car. The company deals with drug manufacturing laboratories where it must maintain the number, name, address, telephone and email of each laboratory. Each laboratory manufactures several drugs, and each drug is manufactured by one and only one laboratory. Each medicine is known by a reference number, a name, a type and a price.

Pharmacies place their orders with the company where the latter must keep the number and date of each order as well as the list of medications it contains with their desired quantities. Each order is delivered by a single Distribution Agent.

- the data dictionary

Coded	Designation	Kind	Length	Syntactic Rules	Semantic rules
Num-Reg-Com phar-name Adr-phar Tel-phar Email-phar	Commercial register number Pharmacy name Pharmacy address Pharmacy telephone Pharmacy email	N A AN N AN			
Num-agent agent-name First name-agent Tel-agent Email-agent	Agent number Agent name Agent first name Telephone agent Email agent	AN A A N AN			
Mat-veh Type-veh	Vehicle number Vehicle type	N AN			
Num-Wilaya Name Wilaya	Wilaya number Name Wilaya	N A			
Num-Lab Lab-Name Adr-Lab Tel-Lab Email-Lab	Laboratory number Laboratory name Laboratory address Laboratory telephone Laboratory email	AN A AN N AN			
Ref-med Name-med Type-med	Drug reference Drug name Drug type	AN AN A			

Price-med	Drug price	N			
Num-com Date-com Qty	Order number Order date Medication quantity	AN D N			



Treatments: CMT (Conceptual Model of Treatments)

The objective of the MCT is to represent the dynamic part of the domain of study, that is to say the activities carried out by the domain. It is also called “Event-Result Model”: The arrival of one or more events will trigger an operation which will produce a result. Its constitution is based on the following concepts:

A. Event

A real event whose occurrence has the effect of triggering the execution of one or more actions. In other words: events inform the information system that something is happening and that it must react. The event can be internal or external to the information system.

Example:

- Arrival of an order: event that triggers the execution of the order processing operation.
- Receipt of an invoice: event which triggers the execution of the operation concerning the payment of invoices.

B. Operation

Set of actions whose sequence is uninterruptible is not conditioned by the expectation of any event other than the initial trigger. An operation produces new events as output.

Example:

The “order preparation” operation brings together the following uninterruptible actions:

- Determination of missing products with the quantities to order,
- Choice of a supplier,
- Writing a purchase order.

C. Synchronization rules

Boolean condition, reflecting the management rules that events must verify to trigger an operation. Synchronization rules are the translation of management rules. They determine the conditions for triggering operations.

Synchronization rules are expressed using logical operators (mainly AND and OR).

Example: To engage in the establishment of an order (operation) you must have:

Either a stock shortage OR a request to satisfy (synchronization rule).

D. Emission rules

The emission rules translate the management rules to which the emission of the results of an operation is subject.

Example:

If the order is compliant, then...

Due to their complexities, and for the sake of readability, the emission rules are generally of type OK, not OK (or)

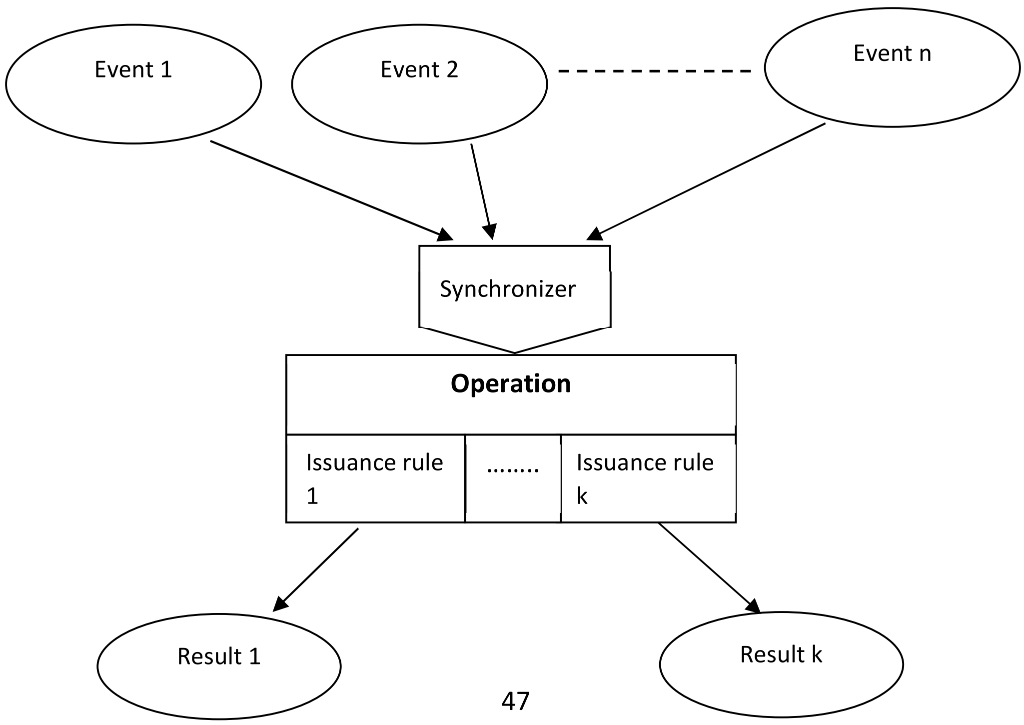
E. Result|

Product of the execution of an operation. The result is also a fact of the same nature as the event and it can be a trigger for another operation.

Examples:

- Order transmitted: result of the operation concerning the establishment of an order.
- File accepted: result of the operation concerning the verification and validation of a file.

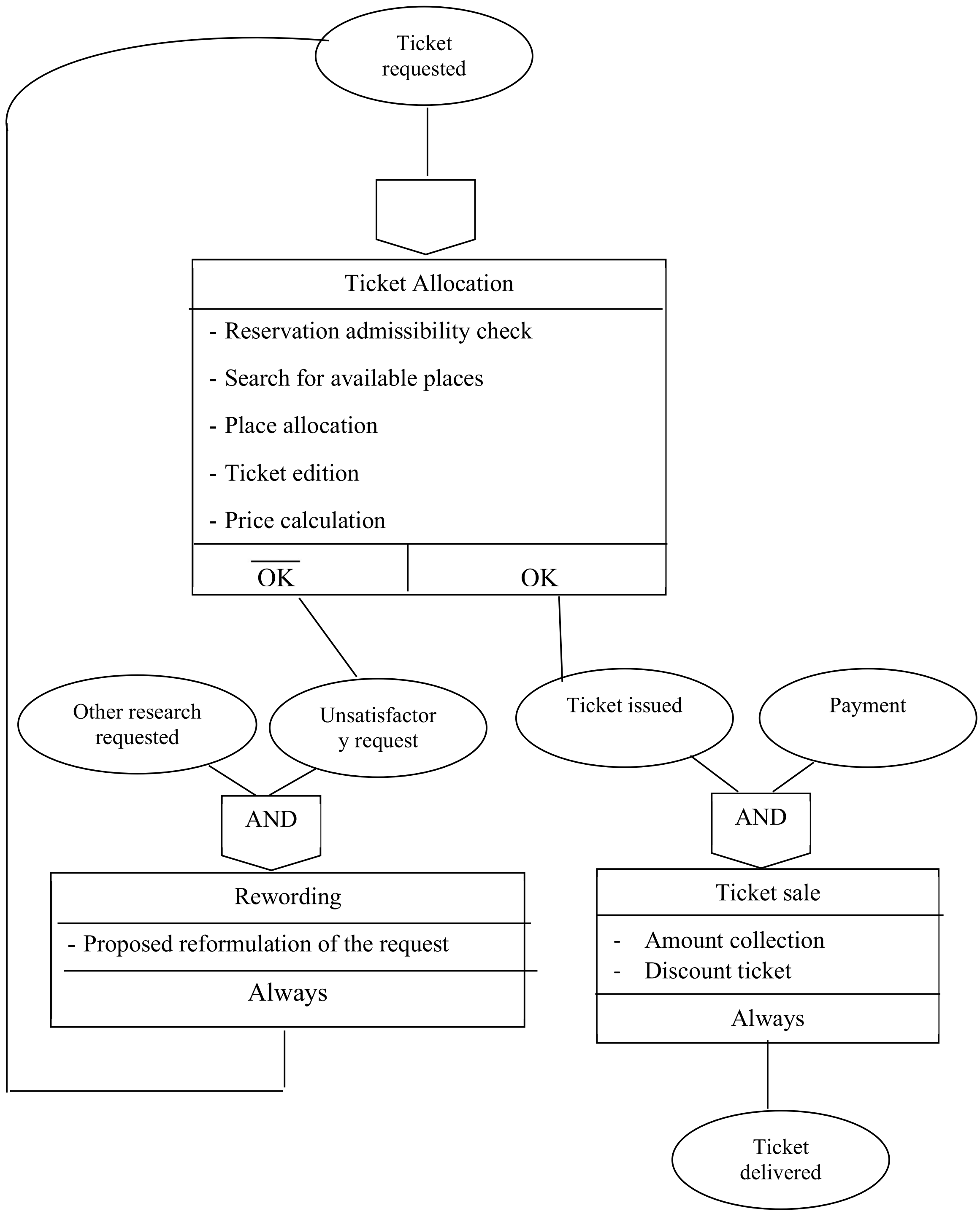
Graphic Representation



Example: Reserving seats in a theatre.

The reservation of seats in a theatre takes place according to the following management rules:

- The reservation counter can issue tickets in advance (reservations) or tickets for immediate entry,
- Place reservations are possible under certain conditions (less than two months in advance, etc.),
- For any seat allocation, a ticket must be issued and a search for available seats carried out,
- Reductions are granted upon presentation of proof (military, students, etc.),
- No ticket can be issued if payment has not been received beforehand,
- For immediate entries, tickets are issued without precise allocation of a place.


Ticket
requested
Ticket Allocation
- Reservation admissibility check
- Search for available places
- Place allocation
- Ticket edition
- Price calculation
OK
OK
Other research
requested
Unsatisfactor
y request
Ticket issued
Payment
AND
AND
Rewording
- Proposed reformulation of the request
Always
Ticket sale
- Amount collection
- Discount ticket
Always
Ticket
delivered

7.2. Organizational Level

7.2.1. Treatments: Organizational Model of Treatments (OMT)

Make technical and organizational choices with a view to creating the software. At this level, the problems of:

- material,

- personal work places,

- organization over time of the treatments to be carried out,

- financial and personnel resources.

In other words: Answer the questions: Where, Who, When

Where: workstation concerns and equipment,

Who: human and financial resources, automatic processing or not,

When: progress over time of the treatments to be carried out.

We don't worry about the how.

In brief MOT = MCT + Place + Time + Nature

A. Event

Concept and formalism identical to that of the MCT: Real fact whose occurrence has the effect of triggering the execution of one or more tasks.

B. Organization rule

Expression of the organization put in place in terms of workstation, nature of processing and chronology.

C. Synchronization

Concept and formalism identical to that of the MCT: Boolean condition, reflecting the management rules that the events must verify to trigger the tasks.

D. Functional Procedure (PF)

Set of actions whose uninterruptible sequence (taking into account the organization put in place) is not conditioned by the expectation of any event other than the initial trigger.

The workstation, the nature of the processing and its progress over time will therefore be common to all the tasks of the same functional procedure.

Noticed :an operation at the MCT level =ΣPFs at the MOT level (at least one).

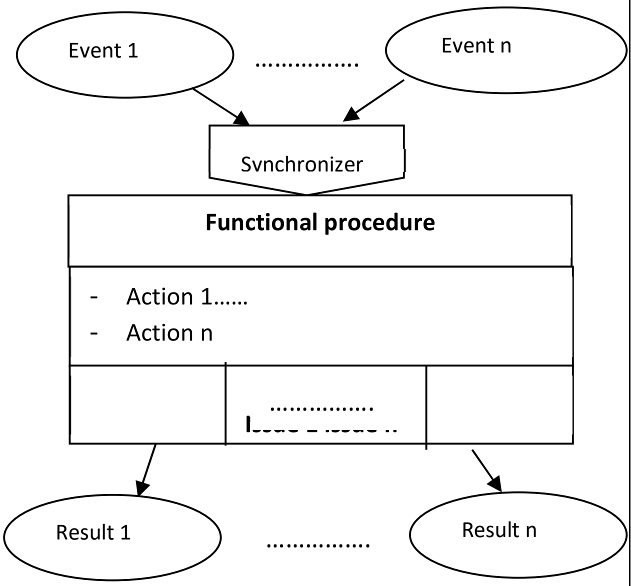
E. Emission rule

Concept and formalism identical to that of the MCT: Condition reflecting the management and organizational rules to which the issuance of the results of a PF is subject.

F. Result

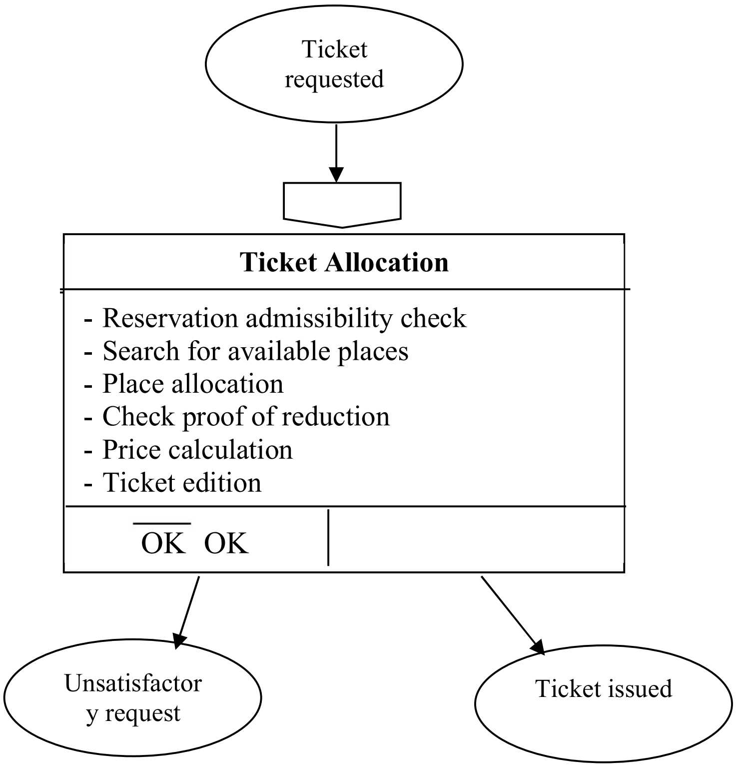
Concept and formalism identical to that of the MCT: Product of the execution of a PF. The result which is a fact of the same nature as the event can be the trigger of another FP.

Graphic Representation

Functional sequence of procedures	**Nature of treatment**	**Workplace**	**Chronological sequence**
	Nature of phase processing (manual, automatic)	Name of the workstation running the phase	Phase progress period

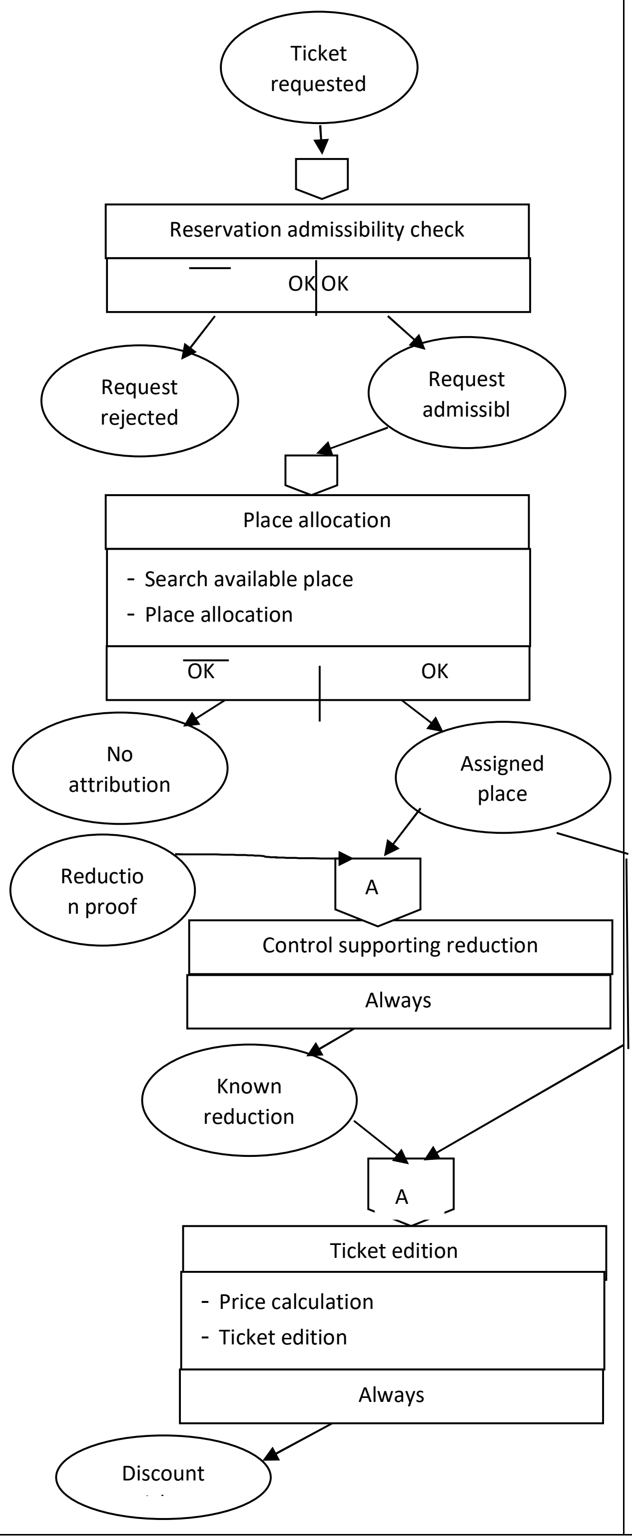
Example:

Taking the ticket allocation procedure from the previous example.



This procedure contains several actions that differ organizationally:

Action	Workplace	Nature of treatment	Timing/Timeline
Reservation admissibility check	Public counter	Manual (conversational)	As requests progress
Search for available places	Public counter	Automatic	As requests progress
Seat assignment	Public counter	Automatic	As requests progress
Control of proof of reduction	Public counter	Manual (conversational)	As requests progress
Price calculation	Public counter	Automatic	As requests progress
Ticket edition	Public counter	Automatic	As requests progress



Functional sequence of procedures	Nature of treatment	Workplace	Chronological sequence
	Manual	Public counter	As requests progress
	Automatic	Public counter	As requests progress
	Manual	Public counter	As requests progress
	Automatic	Public counter	As requests progress

7.2.2. Data: Logical Data Model (LDM)

The DCM is a representation of the data from our future static part of the IS. It is represented in a formalism understandable by the designers and not by the machine. It is therefore necessary to transform it into a representation that can be implemented using computer techniques. This transformation is called logical level. It consists of transforming the MCD obtained into a relational model by applying a few transition rules. Therefore, the following abbreviation is often used:

MLD: Logical Data Model and sometimes, the following abbreviations are also used: - LRDM: Logical Relational Data Model

- RDM: Relational data model

- LRDM: Logical relational data model

A. Rules for moving from an DCM to a relational type LDM

Rule 1: Properties

Each property becomes an attribute of the relational model.

Rule 2: Entities

Each entity becomes a relation (minimum 3rd normal form) of the relational model. The identifier of the entity becomes the primary key of the corresponding relationship.

Rule 3: Association

- ***Binary association of the father-son type***

  A binary father-son type association is an association characterized by:

  * Dimension 2

  * Cardinality of entities participating in the association:

  Father (0, N) or (1, N)

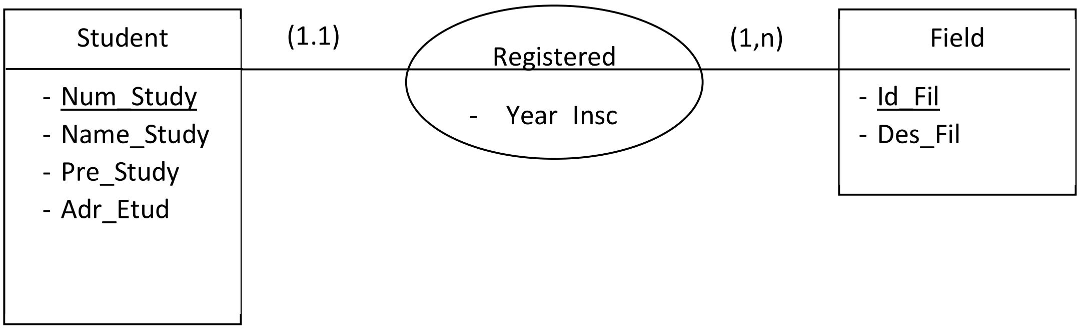
Wire (0,1) or (1,1)

Transforms like this:

- The association disappears
- The father entity becomes the father relationship,
- The child entity becomes the child relationship,
- The identifier of the parent entity becomes an attribute of the child relationship and is called a foreign key,

- The properties of the association become attributes in the child relationship.

Example:



The resulting LDM is as follows:

Student (Num-Etud, Name_Etud, Pre_Etud, Adr_Etud, Id_Fil*, Anne_Isc)

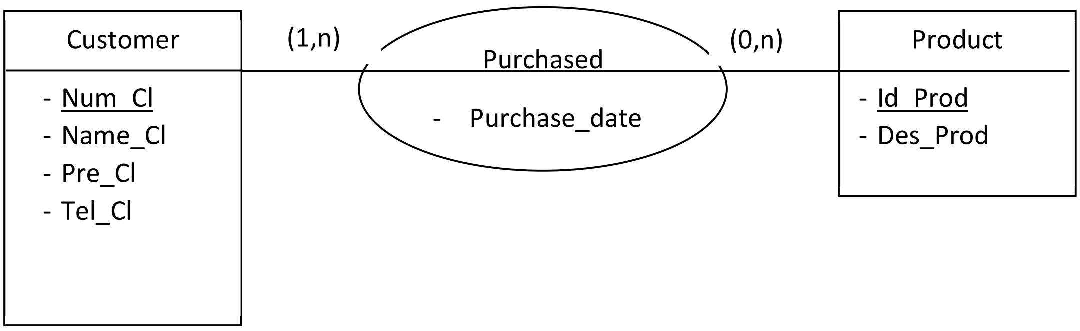
Field (Id_Fil, Des_Fil)

- ***Non-father-son binary association***

  An association having as cardinality entities participating in the association (0, N) or (1, N) is transformed as follows:

- Entities always transform into relationships,
- The entity identifier becomes the primary key of the relationship,
- The association becomes a relationship with its own attributes if there are any.
- The primary key of the association is the concatenation of the primary keys of the entities constituting this association.

Example:



The resulting MLD is as follows:

Customer (Num_Cl, Name_Cl, Pre_Cl, Tel_Cl)

Product (Id_Prod, Des_Prod)

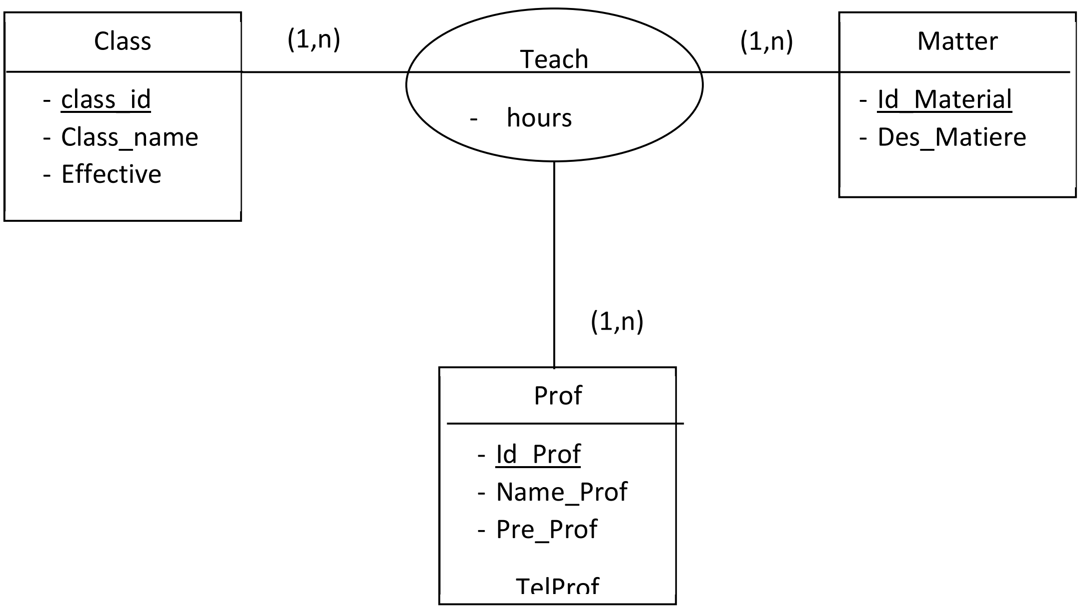
Purchased (Num_Cl, Id_Prod, Date_purchase)

- ***N-ary associations***

  Includes all associations of dimension 3 or more. They transform like this:

- Entities always transform into relationships,
- The entity identifier becomes the primary key of the relationship,
- The association becomes a relationship with its own attributes if there are any.
- The primary key of the association is the concatenation of the primary keys of the entities constituting this association

Example:



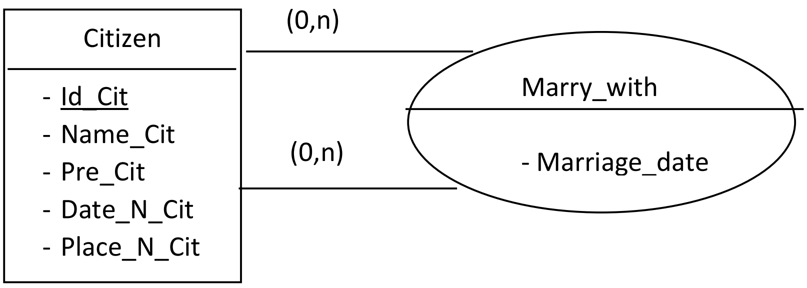
Class (class_id, class_name, staff)

Matter (Id_Matiere, Des_Matiere)

Prof (Id_Prof, Name_Prof, Pre_Prof, Tel_Prof)

Teach (**<u>Id_class,Id_Material,Id_Prof,</u>** hours)

- ***Reflexive association***



Citizen (Id_Cit, Name_Cit, Pre_Cit, Date_N_Cit, Location_N_Cit)

Marry_with (Id_Cit_source, Id_Cit_target, Date_marriage)

7.2.3. Normalization of the relational model

Normalization is the process of organizing data in a database. This process ensures:

- Non-redundancy of data
- Data integrity
- Ease of updating

When normalizing a data model, we must have at least the 3rd normal form.

A. 1st normal form

A relation is in First Normal Form (1FN) if and only if it contains only simple and elementary or atomic values (not structured or repetitive).

- If a table contains groups of repetitive data, you must output these groups of data and create other entities which will contain these repetitive groups.
- If a column of a table contains structured data, this column must be defragmented into several columns which will contain this structured data.

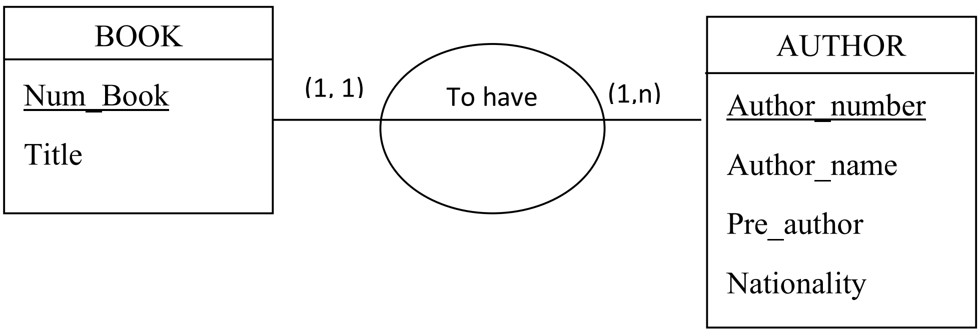
Example

Or the relation BOOKS (Book_Num, title, author, author_nationality)

Book
Book_number
title
author
nationality_author

Book_number	Title	Author	nationality_author
01	EZ-ZILZEL (THE EARTHQUAKE)	TaharOuettar	Algerian
02	THE MARTYRS RETURN THIS WEEK	TaharOuettar	Algerian
03	THE ACE	TaharOuettar	Algerian
04	WAR AND PEACE	Leo Tolstoy	Russian
05	THE DEVIL	Leo Tolstoy	Russian

This table is not in 1FN, therefore it must be broken down as follows:



Book (Book_Num, Title, Author_Num*)

Author (Author_number, Author_name, Pre_author, Nationality)

Num_Book	Title	Author_number
1	EZ-ZILZEL (THE EARTHQUAKE)	1
2	THE MARTYRS RETURN THIS WEEK	1
3	THE ACE	1
4	WAR AND PEACE	2
5	THE DEVIL	2

Author_number	Author_name	Pre_author	Nationality
1	Ouettar	Tahar	Algerian
2	Tolstoy	Leon	Russian

B. 2nd normal form

An entity (Table) or a relation is in second normal form 2FN, if it is:

1- In 1FN
2- All attributes of the relationship or entity depend on the entire key (concept of compound primary key) and not on part of the key.

The second normal form only applies to tables with a compound primary key.

Example

Or the relationOPERATION (Num_Account*, CodeOpe*,DateOpe*,Name,Firstname,LibelOpe,Sum)

Here is an extract from the extension of this table

Account_number	CodeOpe	DateOpe	Name	First name	LibOpe	Sum
125978	30	05/12/2020	Ali	Ahmed	Speed	15000
125978	40	09/15/2020	Ali	Ahmed	Credit	45000
125978	30	05/11/2021	Ali	Ahmed	Speed	30000
125978	40	07/09/2021	Ali	Ahmed	Credit	25000
151936	30	05/10/2020	Tarek	Salim	Speed	35000
151936	40	12/11/2021	Tarek	Salim	Credit	40000

Note that: Name and First name functionally depend on Account_Num Operation label functionally depends on Operation code

Correction: to correct this situation we must subdivide this relationship into three relationships: ACCOUNT(Account_Number, LastName, FirstName) LABEL(OpeCode,OpeLibel) OPERATION(AccountN°*,OpeDate*,OpeCode*,sum)

Which gives us the following tables:

Account_number	Name	First name
125978	Ali	Ahmed
151936	Tarek	Salim

CodeOpe	LibOpe
30	Speed
40	Credit

Account_number	CodeOpe	DateOpe	Sum
125978	30	05/12/2020	15000
125978	40	09/15/2020	45000
125978	30	05/11/2021	30000
125978	40	07/09/2021	25000
151936	30	05/10/2020	35000
151936	40	12/11/2021	40000

C. 3rd normal form

A relation is in third normal form (3FN) if it is:

1- In second normal form
2- Additionally, any non-key attribute is not functionally dependent on any other non-key attribute.

<u>Example</u>

Either the relationship
Flight (Flight_Num, Departure_City, Arrival_City, Flight_Date, Flight_Time, Plane_Name, Manufacturer, Number of Seats, Speed)

We note that the manufacturer, the number of seats and the speed functionally depend on the Plane_Name. Therefore, the correction of this relationship allows us to generate the following two relationships:

Flight (Flight_Number, Departure_City, Arrival_City, Flight_Date, Flight_Time, Plane_Name)

Airplane (Airplane_Name, Manufacturer, No. of Seats, Speed)

<u>Noticed</u>: If we rigorously apply the method of developing a Conceptual Data Model (CDM), the resulting Relational Model will automatically be in 3FN

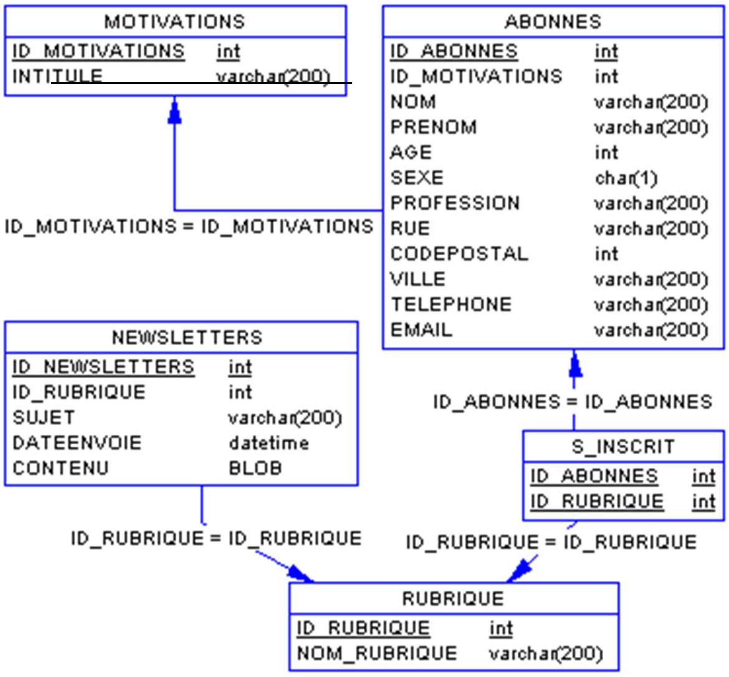
7.3. Physical or operational level

7.3.1. Data: Physical Data Model (PDM)

The physical data model, MPD, makes it possible to construct the final structure of the database with the different links between the elements that compose it. It is a representation of the organization of data taking into account a selected data management system (DBMS). Generally, we opt for relational database management systems (RDBMS). The result is therefore a set of tables containing columns, precisely data types of each column according to the DBMS used and according to what we have declared in our data dictionary.

The language used for this type of operation is SQL. You can also use an AGL (PowerAMC, WinDesign, etc.) which allows you to automatically generate the database.

Example



7.3.2. Treatments: Physical Model of Treatment (MPT)

The Physical Model of Processing (MOT), also called Operational Model of Processing (MOpT) consists of writing the program. This can be generated as part of a “software engineering workshop”. The purpose of methods such as MERISE is the production of automatic "code" from the design. The MPT is interested in the structureinternal of all project applications. Its objective is preparation for development: i.e., breaking down each operation into technical modules:

- Define the internal data of the technical module;
- Define the module's processing (procedure or function);
- Define algorithms or pseudo-codes;
- Presentation of technical processing;
- Input information;
- Release information
- Results.