CHALLENGES TO DESIGN A DISTRIBUTED DATABASE USING UML AND MERISE

Kasereka Masivi Samuel[1], and Ravi Shankar Shukla[2]

[1]*Computer science, Invertis University Bareilly, India*

[2] *Computer science, Saudi Electronic University, KSA*

ABSTRACT : *Up-front handling of the prior and specific factors in the design of distributed database is an important factor that can influence the effectiveness of its implementation. In this paper, we aim at highlighting the major and specific constraints of a distributed database in the approaches of the MERISE and UML methods that can have a negative impact on the implementation of their results and thus deserve the attention of a researcher to create a new and appropriate method.*

KEYWORDS: *Challenges, Design, Distributed Database, UML, MERISE.*

1. INTRODUCTION

Today, the analysis and design of information systems has most of the time vocation to allow the creation of databases, which must represent as closely as possible the reality of the field studied thus requiring the use of a design method (Diviné, 2008) [1]. Note that the databases designed using these methods are either local or distributed. Thescenario for designing can be presented like this:

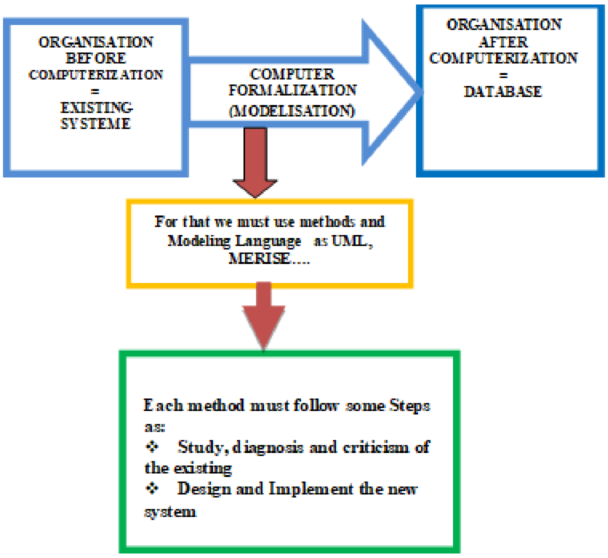

ORGANISATION BEFORE COMPUTERIZATION = EXISTING SYSTEME
COMPUTER FORMALIZATION (MODELISATION)
ORGANISATION AFTER COMPUTERIZATION = DATABASE
For that we must use methods and Modeling Language as UML, MERISE....
Each method must follow some Steps as:
❖ Study, diagnosis and criticism of the existing
❖ Design and Implement the new system

Fig.1 Designing Database procedure

Indeed, as confirmed by Pierre-Alain MULLER and Nathalie GAERTNER (2000) [2], "computer has imperceptibly crept into all of our daily practical activities. Thus, the method must be the most appropriate tool in all areas of microeconomic and macroeconomic life, where information needs to be dealt with reliably and efficiently". Thus, designing of distributed database needs, like any other database, methodology; however, we found that, the design technologies are more oriented towards the design of local databases than to distributed databases and this causes a methodological problem for distributed databases. All conception methods aim at (Rumbaugh, James at al., 2007) [3]:

- Knowing the needs of the users;
- Analyzing these needs;
- Then conceiving models and interfaces for a development in a programming language and Database Management System.

Note that each method has its approaches to achieve these objectives. Presently, the world is dominated by the use of MERISE and UML in the design and realization of programs as method and design language (Nanci, Dominique and

Bernard Espinasse, 2014) [4] and several authors are unanimous on the acceptance of MERISE as a method and UML as a technique or language. It is therefore not surprising that the combination of these tools is effective for software design by naturally developing a unifying and coherent approach between MERISE-UML, referring to their common foundations, their divergences and certain theories about design of distributed Database.

In this work, we want to show the main methodological challenges regarding designing distributed databases in UML and MERISE.

2. RELATED WORKS

Several authors have spoken about Distributed database, but not in the sense of design; rather, in the sense of their creation in a database management system.

Coulouris, George at al. (2012) [5] have developed some principles and practice of distributed computer systems and interesting developments, using a number of cases of modern studies to illustrate their design and development. These techniques have allowed readers to evaluate existing distributed systems and design new systems.

Liu and Obispo (2006) [6] provided some basic concepts and principles of distributed programming techniques. With a "hands-on" approach, the book covers programming paradigms, protocols and application program interfaces (APIs). The presentation uses a narrative, code examples and diagrams designed to explain "subjects in a clear and concise manner". These notions allow performing basic distributed programming techniques used to create network services and network applications, including internet applications.

Mattson, Sanders and Massingill (2006) [7] provide proven solutions to the challenges encountered by parallel programmers and pragmatic advice for using current real world parallel APIs. In their work they met these challenges:

- Finding the competition in a software design problem and break it down into simultaneous tasks;
- Managing the use of data between tasks;
- Creating an algorithm structure that efficiently exploits the competition you have identified;
- Connecting your algorithmic structures to the APIs required for their implementation;
- Specifying software concepts for the implementation of parallel programs.

Dessai (2006) [8] has more talked about how to create distributed databases, its advantages and disadvantages, and other types of database. Joseph Gabay, David Gabay (2008) [9] they find that UML is only a modeling language and have described in their work that the unified process 7 (UP7, Unified Process 7) must be associated with UML for design approach. Christian Soutou (2015) [10], explained in this book how to migrate to the class diagram of UML 2 and vice versa.

Stefan Bergström-Lotta Råberg (2007) [11], provide advice and a real-world case study teaching the reader how to successfully implement the Rational Unified Process (RUP). The reader finds a description of the management of the RUP and details on the effects that the RUP has on the projects. In exploring actual implementation, the authors present a definition of objectives and measures, advice on selecting strategies and steps towards the development of an implementation plan. A discussion of best implementation practices and proven strategies is included. The book ends with descriptions of actual implementations, and a comparison of the book's content with what can be found about this topic in the RUP itself.

Fien Van der Heyde, Laurent Debrauwer (2016) [12] introduce the different diagrams of UML 2.5 from the description of the requirements by the use cases to the profile diagram through the diagrams of interaction, classes, composite structure, transitions states, Activities and components. Then how the interaction diagrams can be used to discover the objects making up the system. This new edition of the book enriches the description of many diagrams and introduces in particular the classes and packages template as well as the relation of fusion of the packages. Laurent Audibert (2014) [13] presents in a pedagogical and rigorous way the bases of the language UML, an overview on the language of constraint OCL, an overview on the design patterns and the implementation of UML by directed works, Complete answers And detailed of all the exercises, numerous examples, Numerous illustration.

Antoine Clave (2016) [14] describes the really useful features of UML (version 2.5), and describes its implementation, step by step, within a "red thread" project. It proposes the use of this tool in several contexts: project management, evaluation of loads, tests and recipes application, writing specifications. Pascal Roques (2015) [15] in his book, he describe UML leaflet contains the essence of UML technology in the form of a 14-page leaflet presenting a standardization of the unified modeling language. Alexandre Guide (2013) [16] presents a description of the UML and C ++ languages, techniques for switching from a design made using UML to C ++ code are presented, for each UML diagram. The most important design patterns are then described and implemented in C ++. Mokrane Bouzeghoub (2006) [17], in his work, anticipates some techniques and global approach of design of information systems of the OOM method from Merise.

Christian Soutou-Frédéric Brouard (2002) [18], describe the construction of a conceptual model using validation and normalization rules. All the mechanisms of derivation of a conceptual model in a relational scheme are clearly discussed with concrete examples. The logic model can then be optimized before writing SQL scripts. Business rules are implemented by SQL constraints, triggers, or in transaction code. The final step is to define views for outside access. The book concludes with a comparative study of the main modeling tools on the market. Roger Mounyol (1995) [19], provides concrete solutions to cases with varied complexities; it opens spaces for reflection both for the modeling and for the conduct of a computerization project; Presents classical and extensive theories are presented and illustrated.

Jean-Luc Baptiste (2012) [20] in his book on the MERISE method, gives a simple, direct and practical information on the MERISE method and the SQL language. How to realize the different models (conceptual models, logical models, physical models) but also the models specific to treatments (conceptual models of treatments, organizational models of treatments ...); Model with MERISE / 2 extensions; Compare some MERISE models to some UML diagrams, the SQL language is presented in a progressive way and is illustrated by numerous examples to learn how to manipulate, filter, sort, group data; Create, modify, delete tables; Assign or remove rights to certain users.

Joseph Gabay, David Gabay (2006) [9], address a pedagogical approach to the normative aspect of UML 2 and an approach to the development of diagrams covering the analysis and design of information systems with progressive learning based on numerous examples, exercises Corrected and real case studies approaching real business projects to allow all professionals, designers and developers to better master UML 2 and acquire a practical approach to implementation. Also their present thirteen UML 2 diagrams by reconciling the strict compliance of the norm with an application centered on the IS of the companies by relying on examples and exercises adapted to the professional context Joseph Gabay, David Gabay (2008) [9]. Russ Miles, Kim Hamilton (2006) [21] provide the minimum knowledge to implement UML 2 to your projects, showing how to use it in the integration of the needs expressed by the client in your model so as not to forget any element of the specifications; How to model the different components of a system and their behaviors how to model the interactions between the different parts of a system and how to deploy a system.

3. MERISE APPROACH AND PRINCIPLES

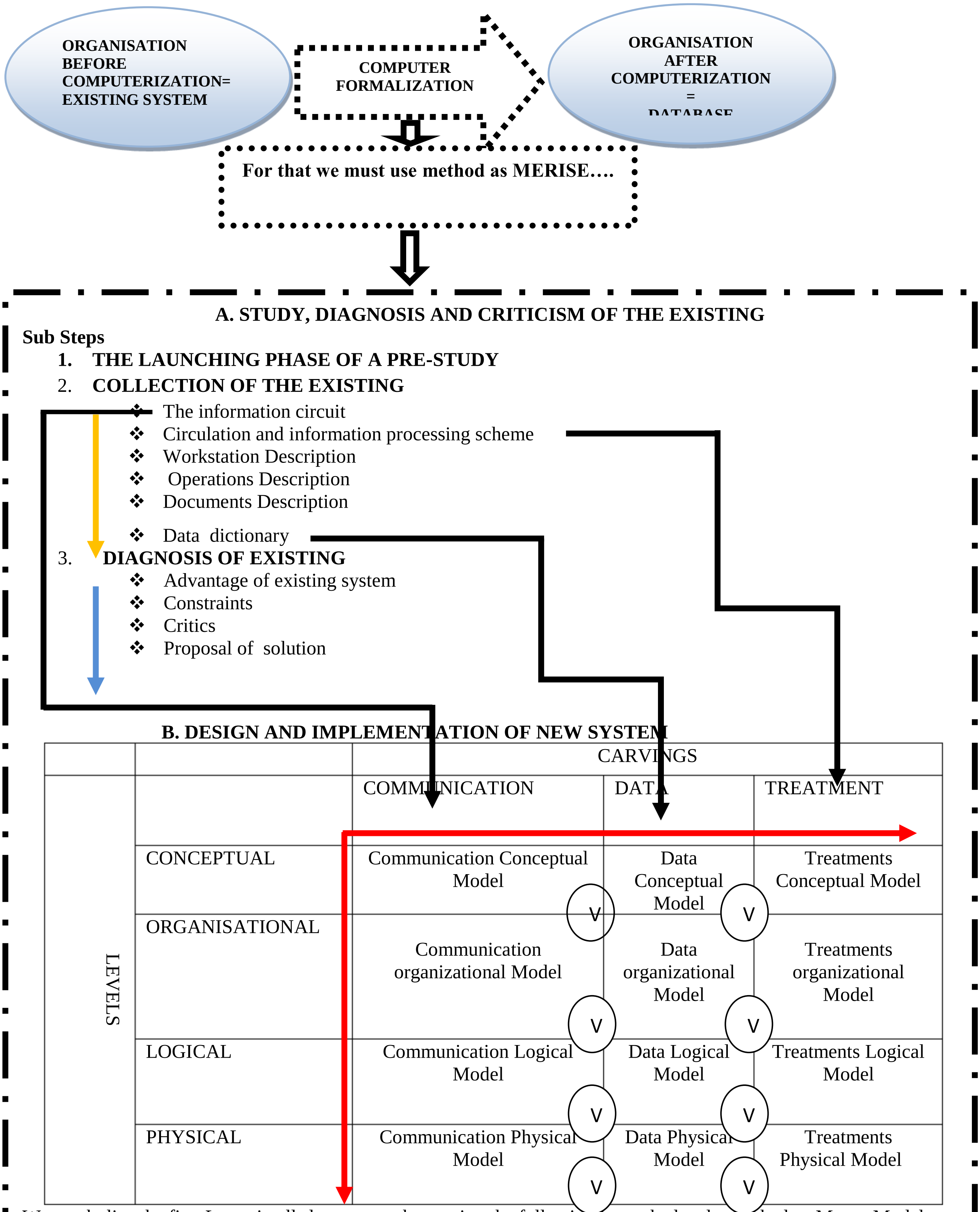
This subsection describes the approach and principles used in MERISE so as to elucide the general overview and specify the deisgn distributed database models.

3.1. MERISE Approach

MERISE an acronym meaning Method of Study and Computerization for Business Systems; it refers to a method of analysis, design and development of information systems. In other words, MERISE proposes a methodology for the realization and management of IT projects; and proceeds as follows in figure 2.

Array comments

For this array, it should be noted that MERISE does not propose principles for elaborating models of the physical design level, which can lead to skidding in the implementation of the new system, especially when the programmer is not the designer of the system. Nevertheless, MERISE offers to the programmer and networker (for the telematics part) the objectives for each model of this physical level.



ORGANISATION BEFORE COMPUTERIZATION= EXISTING SYSTEM

COMPUTER FORMALIZATION

ORGANISATION AFTER COMPUTERIZATION = DATABASE

For that we must use method as MERISE….

A. STUDY, DIAGNOSIS AND CRITICISM OF THE EXISTING

Sub Steps

1. **THE LAUNCHING PHASE OF A PRE-STUDY**
2. **COLLECTION OF THE EXISTING**
   - The information circuit
   - Circulation and information processing scheme
   - Workstation Description
   - Operations Description
   - Documents Description
   - Data dictionary
3. **DIAGNOSIS OF EXISTING**
   - Advantage of existing system
   - Constraints
   - Critics
   - Proposal of solution

B. DESIGN AND IMPLEMENTATION OF NEW SYSTEM

		CARVINGS		
LEVELS		COMMUNICATION	DATA	TREATMENT
	CONCEPTUAL	Communication Conceptual Model V	Data Conceptual Model V	Treatments Conceptual Model
	ORGANISATIONAL	Communication organizational Model V	Data organizational Model V	Treatments organizational Model
	LOGICAL	Communication Logical Model V	Data Logical Model V	Treatments Logical Model
	PHYSICAL	Communication Physical Model V	Data Physical Model V	Treatments Physical Model

We symbolize the first Letter in all slots means the carving the following letter the level, and the last Means Model. CCM = Communication Conceptual Model DCM = Data Conceptual Model TCM = Treatments of Conceptual Model,... The Letter V represents the Validation

Fig.2 Schema and elaboration for cohesion Principles of MERISE Models

3.3. Summary array of the MERISE approach and their Objectives

Stages	**Sub Stages or Models**	**Goals**	**Global Goal**	**Global Result or consequence**	**Observation**
A. The systematic study of the existing	Advisability and feasibility study	Study the utility (opportunity) and feasibility of the project	Quantify the heaviness of the need and understand the functioning of the existing	To design an exact but also useful database for the company	**POSITIVE**
	Collection of the existing	Detect and detail the actual improvement needs to realize the project being timely and feasible			
B. Diagnosis and criticism of the existing	Advantage and Constraints of existing system	Summarize the strengths and weaknesses of the old system that can influence the realization of the new project.	Determine new needs following the strengths and weaknesses of the old system and make a prospective study to consider a new system	Abandon the project or ask for further study either Launch the realization in proposing concrete solutions based on real and accurate facts	**POSITIVE**
	Criticism and Proposition of the solution	Criticize and propose new solutions			
Stages	Sub Stages or Models	Goals	Global Goal	Global Result or consequence	Obs.
C. Design and setting up of the new system	**Conceptual Level**		Define and represent the functions or activities of the company	**Elaborate models representing the reality of the enterprise leading to the new system**	**POSITIVE** But for the Physical Level there is Risk of
	Communication Conceptual Model	Formalize the exchange of information between functional systems and identifying "memory" systems.			

	Data Conceptual Model	represent the enterprise activity in the way it perceives and memorizes the it	regardless of its organization. He answers the question WHAT DOES THE ORGANIZATION DO	**that can meet the concerns identified in the study of the existing. The models of the organizational level allow to:** ❖ obtain the network topologies ❖ Easy implementation of distributed databases	forgetting certain aspects of design in execution by programmer's non-designers.
	Treatments Conceptual Model	Formalize reactions and treatments performed by a functional system, how to react to an information reception			
	Organizational Level (Distributed Database models)				
	Communication organizational Model	Define what is done by each workstation, that is answer the question who does what?			
	Data organizational Model	Add the geographical concept to the data. Each organizational model is a subset of the conceptual model adapted to a data site and quantify the data in terms of the material needed for their storage	Facilitate the design and implementation of network topologies for distributed databases. Integrate the organization on the different models of the previous level, that is, distribute the data back to the workstations and treatment sites. The models obtained at this level allow the service organization of data and Facilitate the design and implementation of network topologies for distributed databases.		
	Treatments organizational Model	Represent the communications between processing sites, attached to a workstation, and a Data site.			
	Logical Level				
	Communication Logical Model	List and fix messages that can be exchanged between site and database.	Define the computer resources to be put to the provision of workstations (users)organize.		
	Data Logical Model	Transformer le Data organizational Model à des enregistrements logiques. Transform the Data Organizational Model into Logical Records.			

	Treatments Logical Model	Describe the computer applications (architecture) being implemented, exchanged messages, and common databases; so this step can detect the interface tools and the inclusion of the existing.			
	Physical Level		Designing models and algorithms for specialist programmers		
	Communication Physical Model	Establish the telematic configuration between computing sites, the techniques of data transmission between applications.			
	Data Physical Model	Represent the future database, the container information in the form of tables, relations and constraints of integrities			
	Treatments Physical Model	Elaborate computer programs their operating environment, real-time monitors, batch and processing, time sharing...			

Fig.3 Summary array of the MERISE approach and their Objectives

4. UML APPROACH AND ELABORATION PRINCIPLES OF UML DIAGRAMS

This subsection describes the approach and principles used in UML so as to elucidate challenges of the design distributed database models.

4.1. Approach, Principles and UML Diagrams

Today, UML 2 describes the concepts and formalism of these thirteen diagrams but does not propose a construction approach covering the analysis and design of a system. This results, for example, in not having a vision of the interactions between the diagrams.

In order to ensure a good level of coherence and homogeneity on all the models, UML proposes, on the one hand, a certain number of rules of writing or standard graphical representations and, on the other hand, common mechanisms or concepts (classes, attributes, operations, packages ...) applicable to all diagrams (Joseph Gabay, David Gabay 2008) [9]; all of these rules are in themselves a modeling language and not a method for lack of specific steps in the elaboration of diagrams.

4.2. UML diagrams

UML groups diagrams either structuro-behavioral.

- Structural diagrams :
  - Class diagram
  - Component diagram
  - Deployment diagram
  - Package diagram
  - Composite Structure Diagram
- Behavioral diagrams
  - Use Case Diagram
  - State-transition diagram
  - Diagram of activities
  - Sequence diagram
  - Communication Diagram:
  - Global interaction diagram

4.3. UP7 Approach

UML is just a modeling language. We do not have today in the standard, unified approach to build models and lead a project implementing UML. However the authors of UML, described in a book [Jacobson2000a] the unified process (UP, Unified Process) (Jacobson, Rumbaugh and Booch, 2008) [22] which must be associated with UML (Joseph Gabay, David Gabay, 2008) [9], however UP had a successive revolution from UP to RUP and from RUP to UP7 which is the latest version of UML process.

Thus we will no longer present UP7 of Joseph Gabay and David Gabay, in their book "UML 2 Analysis and Design". UP7, a UML application approach that is based on UP and is based on the authors' vision and experience of the development process, drawn from the realization of projects in companies with UML. The approach they propose is articulated along two axes: the four phases that correspond to those of UP and seven activities.

According to Joseph and David Gabay, all UML activities begin in the first phase, so from the launching of the project, even the design, implementation, testing, whatever its activity intensity.

4.4. APPROACH ELABORATION AND COHESION SCHEMA PRINCIPLES OF UML DIAGRAMS

4.4.1. Elaboration UML diagrams

The UP7 approach proposes this process in UML design by recommending a sequence of activities and sub-activities that do not necessarily correspond to a UML diagram, but which may correspond to a support schema for the system development outside UML diagrams system (Joseph Gabay, David Gabay, 2008) [9].

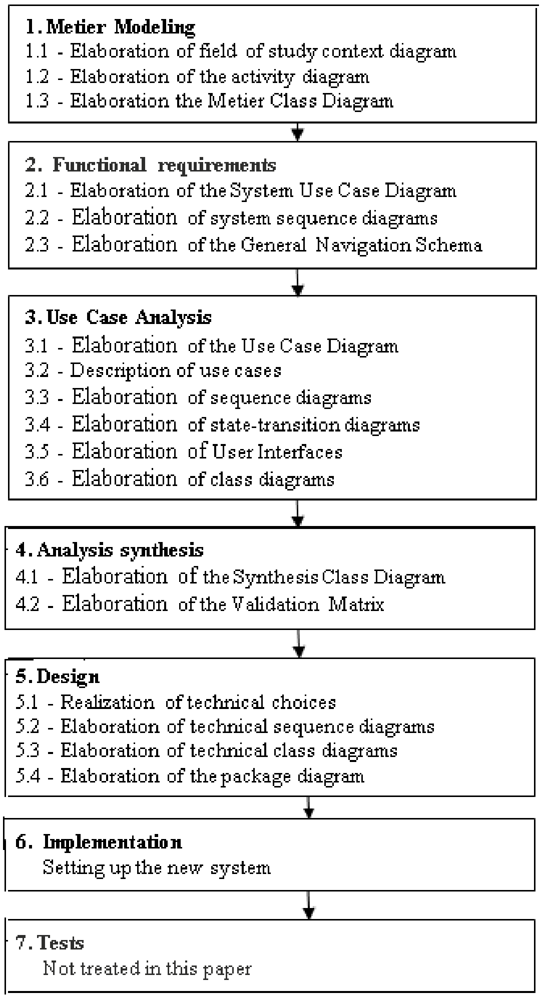

1. Metier Modeling
1.1 - Elaboration of field of study context diagram
1.2 - Elaboration of the activity diagram
1.3 - Elaboration the Metier Class Diagram
2. Functional requirements
2.1 - Elaboration of the System Use Case Diagram
2.2 - Elaboration of system sequence diagrams
2.3 - Elaboration of the General Navigation Schema
3. Use Case Analysis
3.1 - Elaboration of the Use Case Diagram
3.2 - Description of use cases
3.3 - Elaboration of sequence diagrams
3.4 - Elaboration of state-transition diagrams
3.5 - Elaboration of User Interfaces
3.6 - Elaboration of class diagrams
4. Analysis synthesis
4.1 - Elaboration of the Synthesis Class Diagram
4.2 - Elaboration of the Validation Matrix
5. Design
5.1 - Realization of technical choices
5.2 - Elaboration of technical sequence diagrams
5.3 - Elaboration of technical class diagrams
5.4 - Elaboration of the package diagram
6. Implementation
Setting up the new system
7. Tests
Not treated in this paper

Fig. 4 Elaboration UML diagrams procedure

In addition, here under are details of different activities and sub activities with their goals:

Stages/ACTIVITIES	Sub-Stages or Diagrams	Goals	Global Goal	Global Result or consequence
A. Modeling Metier				
	Elaboration Diagram of the System Use Case	Position the system to be studied in all the processes of the company and define the functional limits of the system in general.	Better know and understand processes in which will integrate the future computer system	At the end of this activity, the perimeter of the system to be studied is defined.
	Diagram of Activity	Define the metier processes involved in the system to be developed and to the identification of the actors		
	Diagram of Metier Class	Define domain metier concepts as a class, i.e., information created, transformed or manipulated by experts in the field according to his profession vocabulary		
B. Functional requirements				
	Diagram of System Use Case	Define metier use cases and their general description.	Understand what users do	At the end of these first two activities, the expression of needs is covered.
	Diagram of System Sequence	Scenarios of metier use cases		
	General Navigation Schema	Specify the general navigation of the system to be studied, that is to say, the human-machine interface		
C. Use case analysis				
	Use Case Description	Define all use cases (metiers + IT) and their detailed description		Interfaces and
	Diagram of sequence	Identify scenarios for each use case		

	Diagram of state-transition	List the different states of the objects studied. This Part of the activity is optional because it applies according to the systems studied.	Provide a computer view of use cases system	analysis of each use case is produced but not Still definitively consolidated or validated.
	Diagram of interfaces	Establish user interfaces for each use case		
	Diagram of class	Elaborate classes for each use case		
D. Analysis synthesis				
	Diagram of synthesis class	Group all classes into one diagram	Consolidate and validate all the analysis use cases	Closing the system analysis
	Validation Matrix	Validate the analysis of each use case		
E. Designing				
	Retained technical choices	List techniques to be used	Define and implement the technical architecture choices, and complete the description of the system from the technical point of view.	Cover of the preliminary design.
	Diagrams of Technical Sequence	Elaborate the technical scenarios by use case		
	Diagram of technical class	Determine the technical classes by use case		
	Diagram of package	Package as a bundle of all the technical classes into a single diagram		

Fig. 5 Details of different activities and sub activities with their goals

We notice that in the UP7 approach there are no specifications of the existing study diagrams and the lack of diagrams specifying the distribution of the new system on network.

4.2. Elaboration of UML diagrams cohesion Principles

The following approach of the UP7 does not clearly define the principles of elaboration especially of the passage between certain diagrams such as the obtaining of the classes diagram starting from the existing one because the latter is not even clearly done.

5. CONCLUSION

Currently, research in object-oriented and relational programming, design analysts focus more on the approach they can unanimously adopt in their analysis and design.

Joseph Gabay and David Gabay (2008) [9] confirmed that UML did not have a practical approach and principles which allows the cohesion between its different diagrams for this, proposing an analysis and design approach called UP7, on which we conducted a research that sets the design challenges of distributed databases using UML compared to MERISE.

At the end of our article we summarize the following observations and facts of these two methods:

5.1. MERISE

As far as MERISE is concerned, the latter does not propose principles for elaborating the models of the physical level of implementation, which can lead to skidding in the implementation of the new system, especially when the programmer is not the designer of the system. Nevertheless MERISE offers to the programmer and networker (for the telematics part) the objectives for each model of this physical level.

5.2. UML

- ❖ In the UP7 approach, there is no specification of existing study diagrams but goes directly to the design diagrams.
- ❖ The mixture of the Existing Study diagrams and the conceptual diagrams causes fundamental problems such as not considering of certain aspects:
  - Study of the usefulness of the new system, i.e., compare the expressed claim of the customers to their desired solutions or to propose by the designers.
  - Evaluate the cumbersome activities per workstation or even in a global way to know what the real problem or need is.
  - Do not allow the designer to focus on what type of solutions to adopt in the design itself because not knowing the diagnosis of the existing setting out the real problems to solve.
  - The lack of diagrams specifying the design and implementation of the distribution of the new networked system.
- ❖ The UML in its approach UP7 does not allow the cohesion between the various diagrams (Joseph Gabay, David Gabay, 2008) [9] following the non-routing of ideas in the specific objectives of the diagrams such as the logical principles of passage between the classes diagrams, diagrams of the sequences, interfaces users, use cases ... what persists in this approach.

Comment

The subsidized facts prove sufficiently that there are major and enormous methodological problems to be defied in UML not only in the analysis and design of databases of distributed data but also other methodological problems as indicated above especially that of a study of the existing and network diagrams. So, being a researcher, we have a challenge to make further research in this area and address these multiple challenges and make this famous modeling language UML and method MERISE trustworthy methods in the analysis and design of IT projects for information systems. We find that the database design risk higher in UML than in MERIS so we will no longer look more towards UML in developing our new approach.

5.3. Challenges synthesis array

	UML			MERISE		
	State	Diagrams	Observation and challenges	State	Models	Appreciations
Existing study	Non specified		Specification of existing study is needed because it allows to avoid the problems mentioned above which can be related to the non respect of this stage	Existing	*Page 5*	POSITIVE
Conceptual stage	Existing	*Pages 10-12*	Mixture of the existing and conceptual study.	Existing	*Page 6*	Positive, nevertheless with a problem at the physical Level, if at least programming is done by non-designers
Specific network models	Non specified		To be specified because it facilitates the implementation of a distributed database over a network.	Existant	*Page 7*	POSITIVE

Fig. 6 Challenges synthesis array

5.4. Graphic of challenges RiskSo for their understanding and graphical representation of these challenges, we have proposed ratings for the quantification of their level of risk in database design, as follows:

Level Quantification of implementation of a step in an approach	
No foreseeable risk because this step very well made clearly	2
Minimal risk because this step made but under expected in other steps or not very well done	4
Medium risk because this step is done but no purpose and principles not very well defined or appropriate	6
High risk because this step made with very inappropriate goals	8
Very high risk because the unplanned step	10

So we are going to match assessment values as noted above to the different challenges in both approaches.

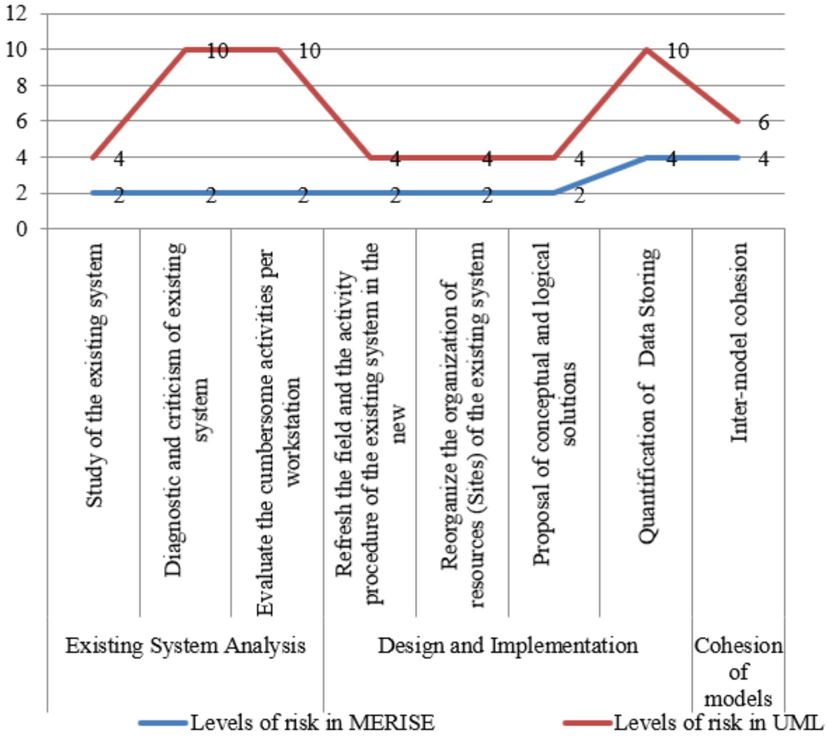


Fig.10 Graphic of Challenges Risk in Database design

5.2. SUGGESTIONS

The major constant is that the disregard of the study, diagnostic and critical of the existing system is at the root of some other challenges in UML design.

So to avoid its consequence as the probable uselessness of the database and obtaining a computerized system not representative of the reality of the information system of the organization studied and not solves the organization needs we suggest:

- The integration of some new schemas describing the existing system allowing the diagnosis of the existing one and / or making some modifications to some UML diagrams to try to reduce the risk in the design in UML.
- Improve the level of design principles and cohesion in the design of UML diagrams.

REFERENCES

1. *Diviné M, Parlez-vous MERISE (Éditions du Phénomène, 2008).*
2. *Muller P, Nathalie G, Modélisation objet avec UML (Second Printing, Paris: Editions Eyrolles, 2000).*
3. *Rumbaugh J. at al., Object-Oriented Modeling and Design (New York: General Electric Research and Development Center Schenectady, 2007).*
4. *Nanci D, Bernard E, Ingénierie des Systèmes d'Information : Merise deuxième Génération (Paris: Vuibert éditions, 2014).*
5. *Coulouris G at al., Distributed Systems: Concepts and Design (5th Edition, Gordon Blair: Lancaster University, 2012).*
6. *Liu M.L., San Luis O, Distributed Computing: Principles and Applications. (Pearson, 2006).*
7. *Mattson Timothy G, Beverly S, Berna M, Patterns for Parallel Programming (Addison-Wesley Professional, 2006).*
8. *Dessai Bipin C, An Introduction to Database Systems (Galgotia Publication, 2006).*
9. *Gabay J, David G, UML 2 - Analyse et conception Mise en œuvre guidée avec études de cas (Dunod-InfoPro - Etudes, développement et intégration, 2008).*
10. *Soutou C, Frédéric B, Modélisation de bases de données UML et les modèles entité-association - Avec 30 exercices corrigés inspirés de cas réels (Eyrolles-Noire, 2015).*
11. *Bergström S, Lotta R, Adopting the Rational Unified Process: Success with the RUP (Addison-Wesley Professional, 2007).*
12. *Heyde Fien V, Laurent D, UML 2.5 Initiation, exemples et exercices corrigés (Eni- Ressources informatiques, 2016).*
13. *Audibert L, UML 2 - De l'apprentissage à la pratique, Cours et exercices (Ellipses-Info+, 2014).*
14. *Clave A., UML Guide pratique au service de l'analyse des métiers (Eni-DataPro, 2016).*
15. *Roques P, Mémento UML 2.5 (Eyrolles-Mémento, 2015).*
16. *Guide A, D'UML à C++, Passage d'une conception UML à une implémentation C++ (Ellipses-Références sciences, 2013).*
17. *Bouzeghoub M., La conception objet des systèmes d'information (Hermès - Lavoisier, 2006).*
18. *Soutou C, UML2 pour les bases de données(2002).*
19. *Mounyol R, MERISE étendue Cas professionnels de synthèse (Ellipses, 1995).*
20. *Baptiste J.L, Merise Guide pratique - Modélisation des données et des traitements manipulations avec le langage SQLEni (Ressources informatiques, 2012).*
21. *Miles R, Kim H, Introduction à UML 2 Une introduction pragmatique à UML (O'Reilly,2006).*
22. *Jacobson R, Booch, Unified Modeling Language Reference Manual, (The 2nd Edition 2008).*