ISSN(e): 2321-7545 Website: www.ijsae.in
International Journal of Scientific Research And Education
Volume||06||Issue||12||December-2018||Pages-8062-8083||ISSN(e):2321-7545
Website: http://jsae.in
Index Copernicus Value- 76.10 DOI: http://dx.doi.org/10.18535/ijsre/v6i12.02

An approach of analysis, criticism and diagnostic of an existing system in Database Design by UML

Authors
Kasereka Masivi Samuel[1], Ravi Shankar Shukla[2]
[1]Inverstis University Bareilly India,
[2]Saudi Electronic University, Saudi Arabia

ABSTRACT
The analysis, criticism and diagnostic of an existing system in the design of database is an important factor that influences its exactitude, usage, and representativity of the reality for the information system of the organization Studied in that database. Thus, the lack of these design faintness, the organization needs could not be solved. This paper is a proposal of integrating the analysis, criticism and diagnostic tools of an existing system in UML language process for database design.
Key words: *Analysis, criticism, diagnostic, existing system, Database Design, UML.*

INTRODUCTION
Alan Dennis, Barbara Haley Wixom and David Tegarden(2012) [1] propose that the analysis phase must include a study of the current system (called the as-is system) and its problems, and envisioning ways to design a new system (called the to-be system).
It should be noted that an existing system is generally born from the expression of needs by users and these needs, which may be due either to the difficulties of working in the existing system, or to a fear of the heaviness of a future project or simply to a need for modernity in its activities.
Computerizing based on the expression of need due to difficulties in an existing system mainly manual; the study, diagnostic and critical of this system is imperative. In order to avoid the probable uselessness of the database or obtain a computerized system not representative of the reality implies no solution of the organization needs. According to Michel Diviné(1994) [2] 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.
On the other hand, in the case where computerizing is due to the fear of a heaviness of a future project or simply a need of modernity in its activities. The study, diagnostic and critical of the existing system is not necessary; because the latter system may not exist. It is in this context that most UML authors have developed their approach, in which they start directly with a set of specifications in terms of use cases where they determine and implement what need the user without worrying about the analysis and criticism of the causes of the expressed needs neither worrying about the details of what the user was doing in the existing system.
Nevertheless, Alan Dennis, Barbara Haley Wixom and David Tegarden (2012) [1] have detailed fundamental four phase model (planning, analysis, design, and implementation) common to all information systems development projects, where he even suggests the study of existing system in a step that he call an analysis strategy. However, they did not give enough practical tools.
This paper suggests integrating practical tools of analysis, diagnostic and critical of the existing system in UML because of its importance and repercussions in the design of a database. So this new approach which is

proposed here, is oriented for computerizing of companies that exist from now on, especially with a manual system, where the design of a database is considered as a solution to expressed user needs.

MATERIALS AND METHODS

To achieve our objectives, we used several methods and techniques including: The Unified Modeling Language that allowed us to do the modeling; the literature review identified the principles and clarify the concepts, while observation and oral interview helped us to come into direct contact with reality to deal with.

RELATED WORKS

Several authors have spoken about the design steps of the UML Object Oriented language without putting more emphasis on the study, diagnosis and criticism of an existing system.

Muhindo Masivi Osée(2018) [3] proposed a methodology adapted the use of the UML modeling language to focus more on the analysis and design of object-oriented systems. Alan Dennis, Barbara Haley Wixom and David Tegarden (2015) [1] showed the dynamic aspects of a system study showing that a systems projects must move through the four phases of planning, analysis, design, and implementation; that mean projects require analysts to gather requirements, model the business needs, and create blueprints for how the system should be built; and all projects require an understanding of organizational behavior concepts like change management and team building. And they showed that today, the cost of developing modern software is composed primarily of the cost associated with the developers themselves and not the computers. As such, object-oriented approaches to developing information systems hold much promise in controlling these costs.

Heyde Fien V, Laurent D (2016) [4] 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 can diagrams be used to discover the objects making up the system. In their 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. Gomaa (2009) [5] has presented here technology and case studies to show how to harness the promise of software product lines and the practicality of the UML to take software design, quality, and efficiency to the next level. An enhanced online index allows readers to quickly and easily search the entire text for specific topics. In their view, Rumbaugh, Jacobson & Booch (2008) [6] –the creators of UML –clearly describe UML concepts, including major revisions of sequence diagrams, activity models, state machines, components, internal structure of classes and components, and profiles.

Audibert (2014) [7] UML 2 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. Clave A. (2016) [8] 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. Roques P. (2008) [9] proposes in his book the implementation of the UML syntax adapted to the modeling of online applications, and declines the analysis performed in UML on three technical architectures: .NET, J2EE, and scripting languages (PHP type).

Guide A. (2013) [10] 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 ++.

Soutou C, Frédéric B (2015) [11] 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.

Debrauwer (2012) [12] shows the design and realization of a rich web application in the field of semantics, exposing the iterative approach of software design according to the iterative approach of the Unified Process divided into four major phases: Inception, elaboration, construction and transition. Its design is detailed by many UML diagrams refined progressively before the realization in Java of the various objects forming the application.

Debrauwer and Karam (2009) [13] offer many practical exercises of varying difficulty in mastering the different aspects of the main diagrams (modeling of a scientific conference, an XML document path, ticket system Computer scientists close to modeling and those close to design and development who wish to master UML 2. More specifically, it presents the use of UML in modeling and design (realization of a state-transitions diagram) and a complete case study (on-line library) explains how to implement UML in an ecommerce framework. In the end their book, they introduces a new chapter dedicated in particular to the diagram of composite structure and the composition of patterns.

J. Gabay and D. Gabay (2008) [14] UML 2 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. They present the 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.

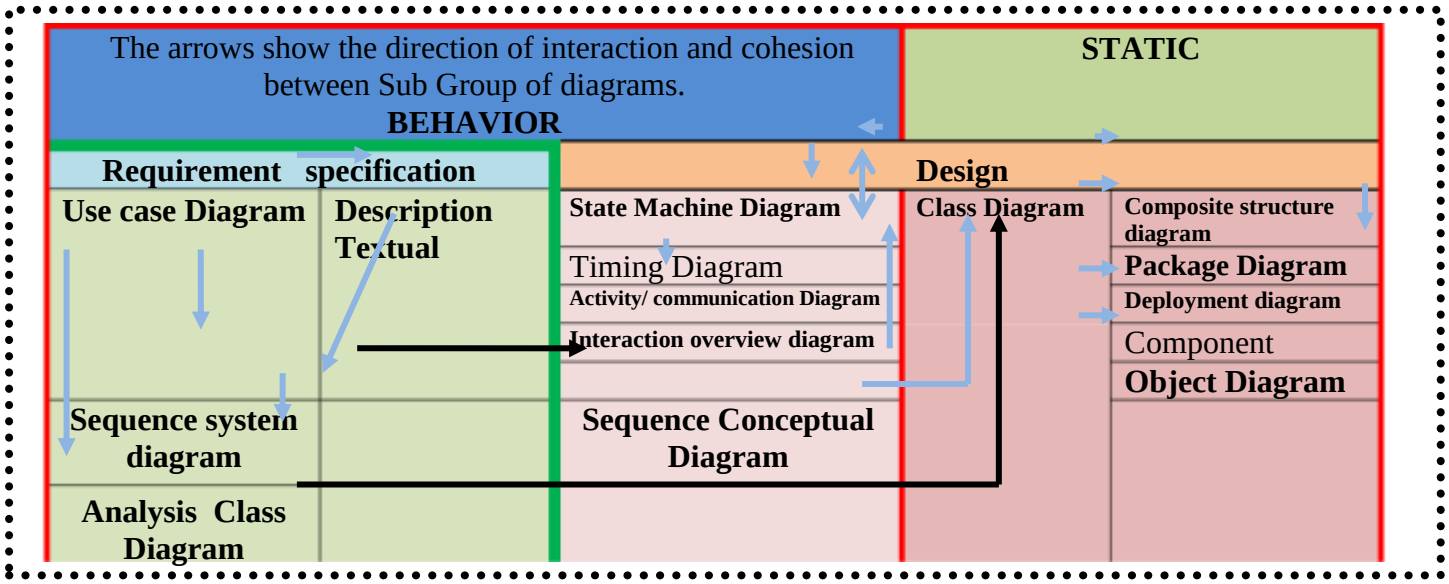
Miles and Hamilton (2006) [15] provide the minimum knowledge to implement UML 2 to your projects, showing how to use it in the integration of the needs expressed by the client in your model so as not to forget any element of the specifications; how to model the different components of a system and their behaviors how; to model the interactions between the different parts of a system and how to deploy a system.

Wazlawick (2014) [16] shows that many people think that UML is a method that is true and that UML is not method but rather Unified Model Language. Bersini (2007) [17] has demonstrated practically how to implement the algorithms of shared applications in object oriented languages like Java. Boucheny (2016) [18] in a first part presents in detail the concepts of OOP, with simple and detailed examples, the theory of this programming paradigm become clearly accessible. Then, after a few tips to install a powerful work environment (plug-in management utility, language documentation, test tools ...), the author proposes the implementation with Python code. Each concept of OOP is explained in detail and illustrated by relevant code. Again, complete and commented examples make it possible to fully understand the power of language. Bouzeghoub M. (2006) [19] anticipates some techniques and global approach of design of information systems of the OOM method from MERISE.

Gilles Roy (2009) [20], gives a predominant importance to the conceptual model of data analysis in the design by proposing rules, techniques, tips and warnings illustrated by numerous examples and case studies which adopt the notation UML by integrating the techniques of modeling and derivation rules within a logical and real-world continuum by providing the necessary support for the success of a database design project and the consistency of models [8]. Craig Larman(2015) [21] wanted update his book to fully reflect the new UML 2 standard in the art of object design, and to promote high-impact, iterative, and skillful agile modeling practices and allow developers and students to learn object-oriented analysis and design (OOA/D) through iterations of cohesive, start-to-finish case studies. Leszek Maciaszek(2001) [22] in their book describes the methods and techniques used for analysis and design, with implementation issues addressed to the extent to which they must be considered in the design with UML.

ABOUT UML POCEDURE

We have summarized as following from above related works, and especially those of Muhindo Masivi Osée, Alan Dennis, Barbara Haley Wixom and David Tegarden, J. Gabay and D. Gabay, and for Pascal Roques:



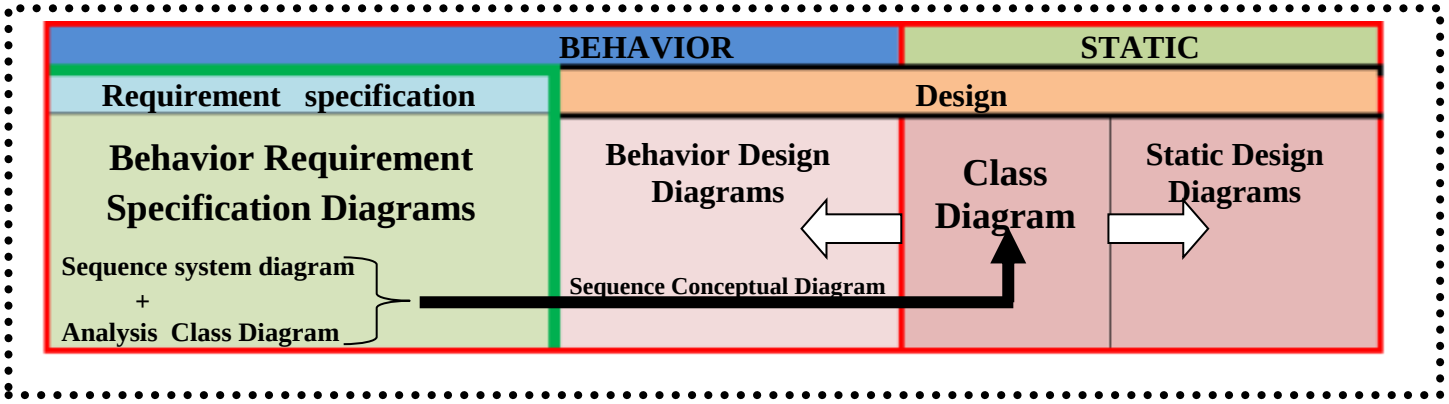
We observe on this table that there are behavioral diagrams in requirements specification and behavioral diagrams in design. In the same way, the transition between the requirements and design specifications diagrams is ensured by the sequence system diagram supported by the analysis classes and these links are represented by black arrows. In addition, we find that the class diagram provides a bridge between the behavioral and static design diagrams. We have arrived at the subgroups of the following diagrams:

Behavior Diagrams

Behavior Requirement Specification Diagrams

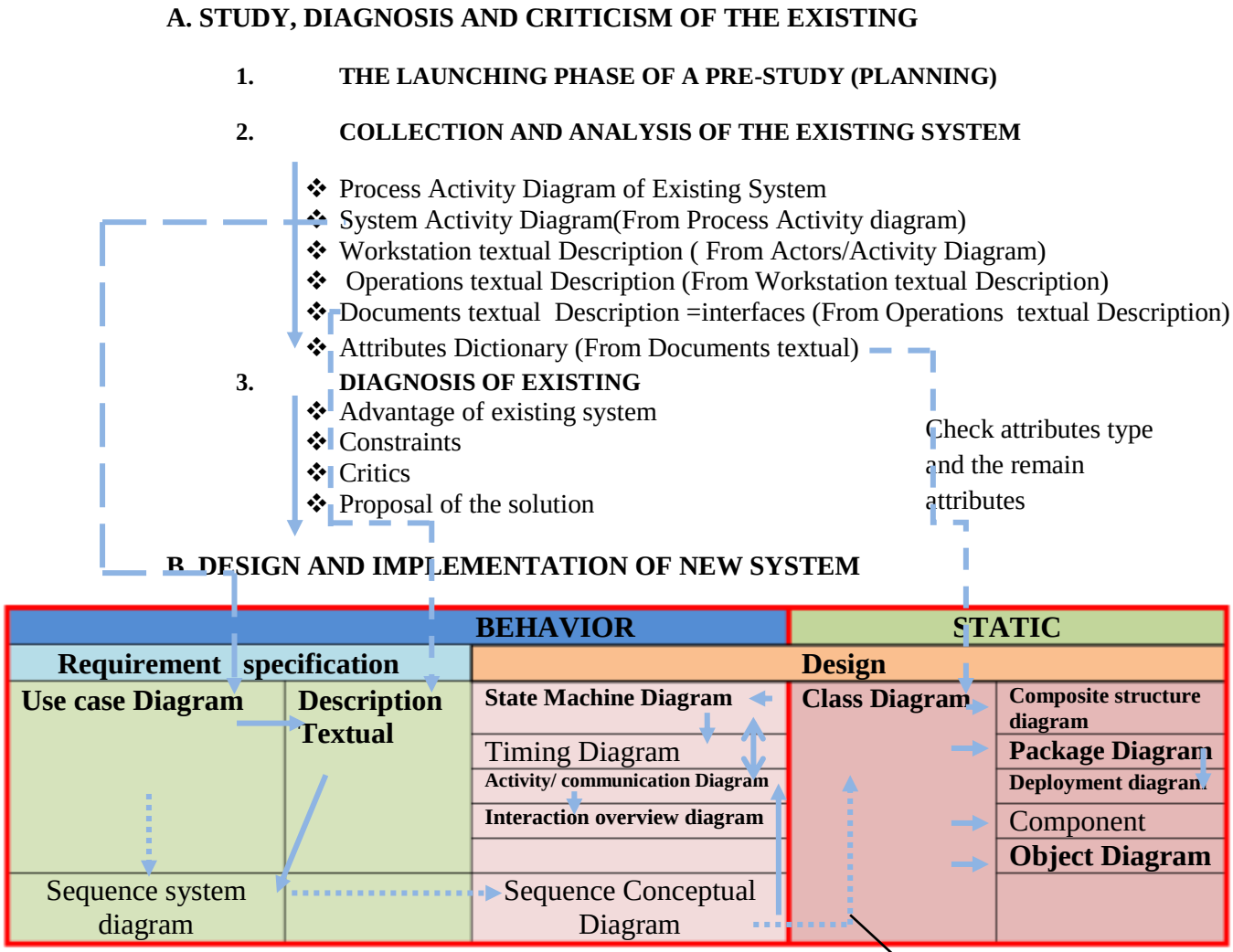
Behavior Design Diagrams

Static Design Diagrams



UML NEW PROCEDURE PROPOSAL

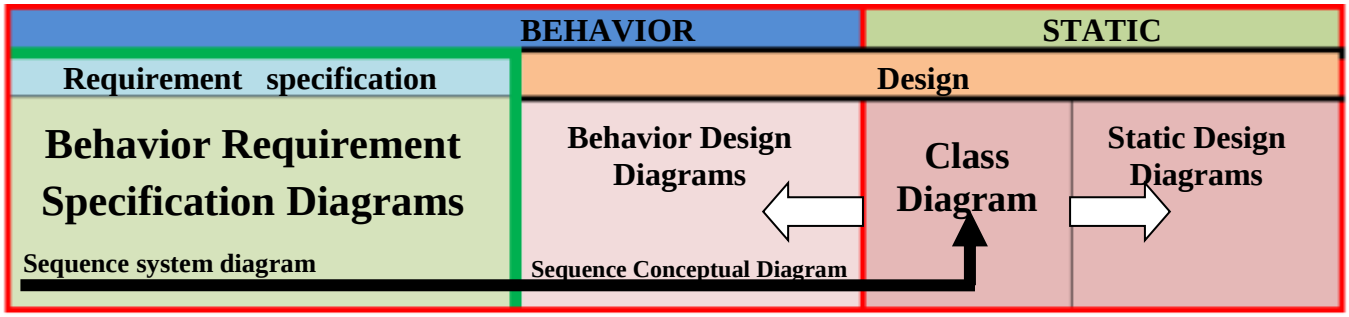
On the graph below we present a diagram of a UML procedural proposal integrating the analysis, the criticism and the diagnosis of the existing system.



The dashed arrows show the direction of data collection from the study of the existing system towards the diagrams in category and specific new design phase.

1.New links of cohesions between subgroups of diagrams

In this new approach, the transition between the requirements and design specifications diagrams is done only by the Conceptual Sequence Diagram based on the sequence system diagram and these links are represented by dotted arrows. The synthesis between subgroups of diagrams after study of the existing becomes:



The arrows show the direction of interaction and cohesion between Sub Group of diagrams.

2. Contributions and Interpretation of the approach

In this part we present a brief explanation of our new approach and specifying the contribution of some diagrams, added models, modified or even some cohesive links between diagrams modified.

A. Diagrams and models added

According to the need and reason mentioned below we considered adding these tools in UML:

Process Activity Diagram of Existing System (From interview and observations)

This step, made during interviews of workstations, helps us improve knowledge about the field to analyze, especially on process, documents, and technical vocabulary in the organization; in addition it gives an accurate picture of the information process related to the case studied.

System Activity Diagram(From Process Activity diagram)

Apart the above diagram this step help us also to improve knowledge of the observer about the field to analyze, specially on: information systems (information circuit), documents, rules observed and technical vocabulary in the organization, formalizing activities done by different actors that mean mains functionalities of the existing system means the Use Case Diagram of existing System.

Workstation textual Description (From Actors/Activity Diagram)

In this step we evaluate the heaviness of workstation, then helps us to know really which workstation or what must be computerize, if we not have, then also is not necessary to computerize or we computerize for technologies purposes but not for solves heaviness in activities. Note that a workstation is defined as a structure composed of human, hardware and software resources to perform a specific job within an organization.

Operations textual Description (From Workstation textual Description)

This step describes the operations which are in the precedent steps showing input documents, output documents and the processing between the input and the output of each operation. It prepares the material of the queries or views in DBMS and in oriented object modeling details system operations for their elaboration and implementation.

Documents textual Description =interfaces (From Operations textual Description)

In the following arrays, we recover all rubrics that are on each document from the preview step, specifying their type (date, text, picture…), obtaining mode (memorized, calculated), formulas if they are calculated. This step helps collects rubrics and by his calculated rubrics formulas details system operations on class diagram for their elaboration and implementation.

Attributes Dictionary (From Documents textual)

This table synthesizes and refines the rubrics for the different diagrams of the design by eliminating some anomalies like redundancy, synonymous and polysemy.

B. Diagrams and modified steps

Similarly according to the need and reason mentioned below we have considered adding these tools in UML:

Textual description

Retrieve the items to use on Use Cases not on interfaces but rather documents because an existing system is composed by documents in the form of register and reports.

Activity diagram

This diagram represents the chronology of the activities performed by the different actors, nevertheless the present diagram makes the activities carried out by only the old actors as if we are still in the old system by ignoring that it is the new system that realizes many activities. So we have to propose the following:

1. Add the system as an actor with its operations that it realizes
2. Make him follow workstation textual description of the operations performed by the other actors finally to reassess the heaviness charged in their workstation.

Sequence diagram

In addition to the role of sequence diagram to assign responsibilities in terms of operations on the classes, the latter will even allow us to assign the rubrics across its operations class systems (can therefore change the state of an object in a class) generally derived from more basic or principals Uses Cases. Note that we call basic use cases those that allow the system to memorize the basic data to the process and the rest of the cases of complementary uses because they help to memorize the additional data what the main ones enter.

Class diagram

The class diagram is the foundation of a database and does not have explicit principles for its development so we proposed to reinforce these principles as follows:

1. Resume the analysis class diagram developed previously without rubrics
2. Add Class System Operations (Can Change the State of an Object in a Class) by Design Sequence Diagram
3. Add rubrics from the system operations classed by design sequence diagrams in the concerned classes.
4. The rest of the rubrics that are found on our collector Attributes that have not yet been assigned to classes by the previous system operations classes can be added to the respective classes.
5. The rest of the topics that are found on our Attributes Dictionary not yet assigned to classes by the previous system operations classes can be added in the respective classes.
6. Add some operations like modify, delete ... to the respective classes according to the need not being able to be found on the documents in the existing system, operations which can only be found on software interfaces.
7. Accompanying the class diagram by quantifying the probable storage space of class instances.

ABOUT ANALYSIS, CRITICISM AND DIAGNOSTIC OF AN EXISTING SYSTEM

In UML the phase that would correspond to the study of the existing one is titled specification of the requirements, where represents and to specify the needs of the user in term of new System considering the system like a black box in its own right, therefore not saying in detail what the system will do. Nevertheless, in the case where there was existing system, the user will also have to specify his needs to see even in detail in this step how he did it (manually or semi-manually), to inspire the designer in his conception and representation of the user needs; that is why we will insert the study, diagnostic and critical of the existing system in the UML approach to integrate this aspect of things.

It should be noted that the specification of the requirements does not replace the study of the existing system because the specification of the requirements is a description of what the new system will do for the user, that is to say a specification contract between these two contracting parts; while the study of the existing system describes what the user was doing in his system without considering what the new system will do.

Note that in the stage of the study of the existing system, one determines the structure of the information system existing by a precise study by determining the real situation of the organization; thus we elucidate the new needs to be satisfied and make a prospective study to consider the evolution of the system over time. The Study, diagnosis and criticism of the existing System has as purposes:

1. To study the advisability of the project (Is it consistent with the company's objectives?) and feasibility (Is the project feasible from the constraints of realization as planning, finance ...?);
2. To improve knowledge of the observer about the field to analyze, on information systems (information circuit), documents, observed rules and technical vocabulary in the organization;
3. To know really which workstation or what must be computerized;
4. To detail all treatment that will be implemented in the queries or views in DBMS;
5. To provide also to the next steps, the "material", i.e., an inventory of documents and information;

6. To help the company's executives choose to launch the realization, ask for further study, or abandon the project.

Generally this step is subdivided by three sub steps: the launching phase of a pre-study, collection of the existing, diagnosis of existing and each of them has can have sub steps according his goal.

1. The launching phase of a pre-study

Called preliminary analysis, this step aims to study the advisability of the project (is it useful compared to the real business goals?) And feasibility (Will the project is feasible from the constraints of realization as planning, finance ...?)

On what, is based a study of computerization?

A preliminary study is available on the known elements from:

• A master plan that, provided by the company,

• On the expression of a need by a user.

In both cases, a prior interview is required, to properly know the real needs and strategic constraints.

What that the business manager expect this phase

The preliminary study results in a draft after which the company's executives chose:

1. Launch the realization,
2. Ask for further study, or
3. to abandon the project

Pre-study: Advisability and feasibility study

Studying the opportunity of the project refers to examining its usefulness in relation to the real objectives of the company and the feasibility of a project refers to answering the question if the project is feasible in relation to the constraints of realization. Here we must be sure that opportunity and feasibility are confirmed then next step for collection of the existing can be done [18].

2. Collection of the existing

2.1. Introduction

This phase aims to identify and detail the actual needs for improvement to realize the project when it is timely and feasible.

2.2. Information Collection

The collection used to define and describe the existing system, and leads to an understanding of the current functioning of the information system concerned by the project.

Principles

This phase aims to improve knowledge of the observer about the field to analyze, on: internal and external actors, information systems (information circuit), documents, rules observed by the players in the management and technical vocabulary in the organization.

It provides also to the next steps of UML language the "material", i.e. an inventory of documents and information. We have proposed to evidence the result of this phase using 6 tools:

1. Process Activity Diagram of Existing System
2. System Activity Diagram(From Process Activity diagram)
3. Workstation textual Description (From Actors/Activity Diagram)
4. Operations textual Description (From Workstation textual Description)
5. Documents textual Description =interfaces (From Operations textual Description)
6. Attributes Dictionary (From Documents textual)

3. Diagnosis of existing

This phase is based on the observed operations, to identify the real needs to be met. The remarks expressed by the users can find their causes in the malfunctions of the details of internal administrative and IT procedures. In this phase the study details, refines, and can sometimes call into question the objectives established in the IT master plan. During the collection phase, the analyst notes the constraints on actors and processes.

Criticism of the existing, is a work that should be done with all the necessary hindsight and at the most opportune moment, because it consists of identifying strengths and weaknesses in the organization of the company as well as the processes that take place easily, those that present difficulties of functioning and thus be able to validate the field of the study as it was initially defined where to reformulate the problem to be solved by taking into account the observations that it found in the business. Ultimately the critic will objectively present the processes for which automation will bring a significant improvement.

In general, in the criticism of the existing one will have to take into account the flexibility of the men and the system, the criticality of the processes, or even take into account the interference with existing systems and all other constraints whose impact on the project is not negligible. To do this, we must ask ourselves the following questions:

- Are the documents sufficient to effectively manage the system?
- Do the documents have all the necessary information or rubrics?
- Do not the types of treatment entail errors and delays in carrying out operations?
- Are the workstations not overloaded or underemployed given the working hours required per day, the operations required and to be performed by each workstation?
- Are the allocated resources (human, hardware, software) at each workstation sufficient or are they not too much?
- Are the algorithms (descriptions and treatments) efficient in terms of time and space?
- Are the data storage systems (local DBMS or Server ...) appropriate to better manage the storage of data?
- The system architecture (client-server, distributed, centralized ...) and platforms (Web, Windows, Linux ...) are they appropriate for the management of the information system?

For our case we will focus to demonstrate by an example the capital importance of the first phase that is study, diagnosis and criticism of the existing system in UML approach.

CASE STUDY: COMPUTER CENTRE (REPUBLIC DEMOCRATIC OF CONGO)

All documents, procedures those we will use in this work are taken from AGAPE/MAHAGI one of Congolese's computer centre.

1. The Launching Phase Of A Pre-Study

Based on the expression of the need for computerization by users and forecasts in the master plan that provided by the company we will elaborate his preliminary study. Then, we will decide Launch the realization, ask for further study, or to abandon the project.

As said above the opportunity the computerization it was mentioned in the master plan and a need expressed by users. So about the feasibility and opportunity, company is supplied with well-educated personnel in management and will need to be strengthened in terms of IT tools. Second, depending on the need and opportunity of the project, the company is ready to mobilize the necessary means for its realization. Even though, after the analysis we find that our project is feasible and timely because the company has more assets than constraints mastered through retraining of the IT personnel.

2. Collection Of The Existing

As the project it is timely and feasible then we can realize this phase identifying and detailing the actual needs for improvement to realize using the 6 tools cited above: Process Activity Diagram of Existing System, System Activity Diagram, workstation textual Description, Operations textual Description, Documents textual Description = interfaces and Attributes Dictionary .

The collection used to define and describe the existing, and leads to an understanding of the current functioning of the information system concerned by the project. To improve knowledge of the observer about the field to analyze, on actors, information systems (information circuit), documents, rules observed by the players in the management and technical vocabulary in the organization we can elaborate our tools one by one.

Before these sub stage implicated actors must be defined, for our reception, charged of office, customer, Bookkeeper, material Provider, Executer Solicited Service, IT Chief Service.

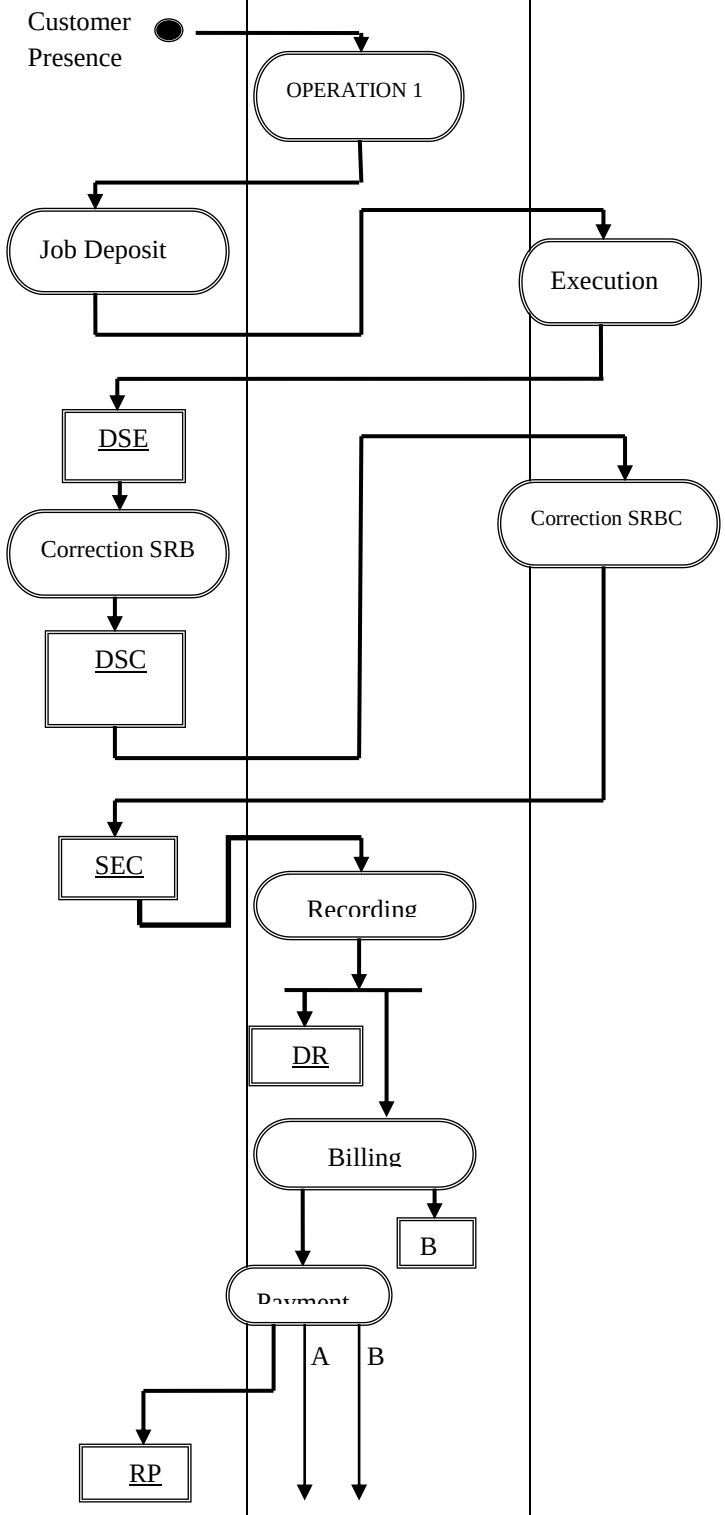
A) Process Activity Diagram of Existing System

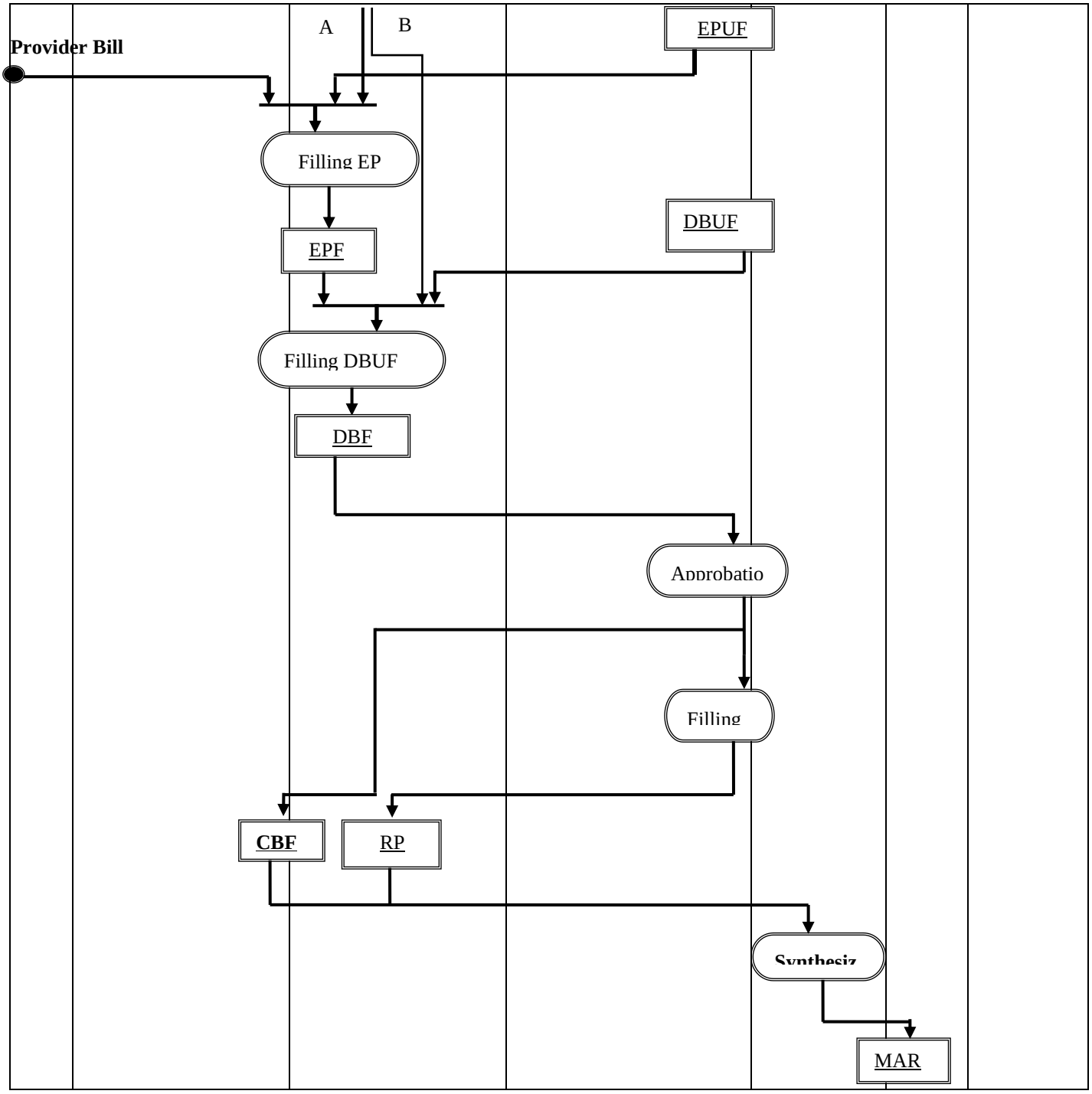
This step helps us improve knowledge of the observer about the field to analyze, especially on process, documents, and technical vocabulary in the organization. This diagram in study of existing system is a tool that aims at giving an accurate picture of the information process related to the case studied. It should rather be a graphic representation made during interviews of workstations. In this tool, we have the knowledge about the different actors, the documents exchanged between them.

We have elaborated this diagram as UML symbols of activity diagram. On this diagram we call actors who are directly manipulating operations and documents in computer centre. So this tool shows different operations executed by each actor one document received.

This is the representation of the operations performed by the actors in the scheme of circulation and processing of information; it is in a table, the actors in the header, and carrying documents and Operations connected with arrows.

An operation and document are represented as follows:

Provider	Customer	Receptionist	Executer Solicited Service	Bookkeeper	Charged of office	IT Chief Service
						



LEGEND

DSE: Draft Service Executed
DSC: Draft Service Corrected.
SEC: Service Executed Corrected
DBUF: Daily Balance Unfilled
BJR: Daily Balance Filled
RP: Receipt Paper
EP: Expense Paper
EPUF: Expense Paper Unfilled
CDBFA: Copy of Daily Balance Filled Approved
MAR: Monthly Activities Report.
DR: Daily Register
BR: Bill Register

B) System Activity Diagram (Process Activity Diagram)

Apart the above diagram this step help us also to improve knowledge of the observer about the field to analyze, specially on: information systems (information circuit), documents, rules observed and technical vocabulary in the organization, formalizing activities done by different actors that mean mains functionalities of the existing system means the Use Case Diagram of existing System.

The principles of elaboration and cohesion are:

- Determine stakeholders on the information circuit and represent them in circles and this in a large rectangle.

At the conceptual level, the stakeholders are the main internal activities for achieving the overall goal; these stakeholders correspond to the internal actors of the above diagram expressed by verbs and the operations found on the different workstation descriptions in our case we have: Reception, Charged of office. Note that our main goal is billing service in Congolese computer centre structure.

- Determine the partners

A partner is a stakeholder outside the company or outside system to be computerized. It can be perceived in a functional way and described by a verb, physical. Inspiring once again from our information circuit, our external actors correspond to the partners who are in a physical way: Customer, Bookkeeper, material Provider, Executer Solicited Service, IT Chief Service.

- Determine exchanged messages

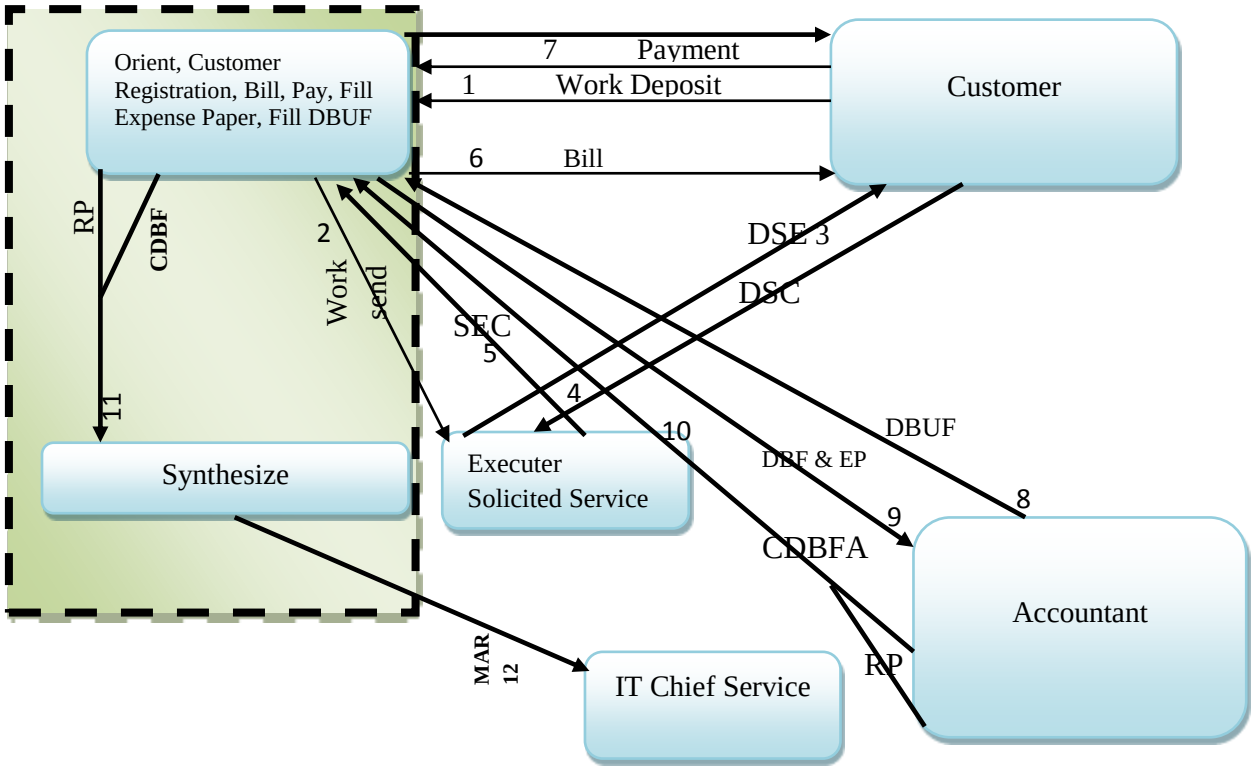
A message is information or a flow of information exchanged between stakeholders or between stakeholders and partners specifying the requests or replies exchanged necessary for the functioning of the system and generally these messages are written on documents. The messages coming from our Process Activity Diagram of Existing System are carried on the following documents:

- Draft Service Executed
- Draft Service Corrected.
- Service Executed Corrected
- Daily Balance Unfilled
- Daily Balance Filled
- Receipt Paper
- Expense Paper
- Expense Paper Unfilled
- Copy of Daily Balance Filled Approved
- Monthly Activities Report.
- Daily Register
- Bill Register

- Represent partners by circles but outside a large rectangle of stakeholders and represent the different messages between partners and stakeholders by arrows

After applying the above principles, we found the following communication conceptual model:

Internal actors	Stakeholders (activities)
Reception	Orient, Customer Registration, Bill, Pay, Fill Expense Paper, Fill DBUF
Charged of office	Synthesize



LEGEND

DSE: Draft Service Executed
DSC: Draft Service Corrected.
SEC: Service Executed Corrected
DBUF: Daily Balance Unfilled
DBF : Daily Balance Filled
RP : Receipt Paper
EP : Expense Paper
EPUF : Expense Paper Unfilled
CDBFA: Copy of Daily Balance Filled Approved
MAR : Monthly Activities Report.

C. Workstation textual Description (From Actors/Activity Diagram)

A workstation can be defined by a structure composed of human, hardware and software resources to perform a specific job within an organization. Thus, our internal actors with specific resources and jobs become work stations so each workstation is equal to the workstation.

NB. After this step we can already know which workstation to computerize.

WORKSTATION DESCRIPTION
Name : Reception Label : Reception Place : IT Building Timing : $8h^{00}$ AM to $4h^{00}$PM
OPERATIONS

Nº	*Description*	*Type execution*	*Case/Day*	*Unity time/minute*	*Time/Day*
1	Orientation	Oral	20	3’	60’
2	Customer recording	Manual	20	5’	100’
3	Billing	Manual	20	5’	100’
4	Payment	Manual	20	5’	100’
5	Filling Expense Paper	Manual	5	5’	100’
6	Filling Daily Balance	Manual	1	30’	30’

RESSOURCES Total 490’= 8H 10’

Human : Bachelor

Hardware: Pen, Register, stump…

Software: non

OBSERVATION

If the recording of receipts, expenses and Expense Paper, has been carried out in a database, production of daily Balance would be automatic and time-saving especially for receptionist who works more than 8 hours of Work at work without break.

WORKSTATION DESCRIPTION

Name : Charged of office

Label : Charged of office

Place : IT Building

Timing : 8h00 AM to 4h00PM

OPERATIONS

Nº	*Description*	*Type execution*	*Case/Day*	*Unity time/minute*	*Time/Day*
1	Documents Synthesis	Manuel	20	25’	500’=8H 33’

MOYENS

Human : Bachelor

Hardware : Paper, pen, calculator

Software : non

Observation

If the daily reports were automated, the monthly report, the annuals are automatically constituted and the charged office will not work hard enough.

D. Operations textual Description (From Workstation textual Description)

Here we describe the internal actors operations of Internal WORKSTATIONS.

OPERATION DESCRIPTION

Operation : n°1

Label : orientation

Treatment Type: oral

Periodicity : $8h^{00}$ AM à $4h^{00}$ PM

INPUT/OUTPUT

Input :Customer Arrival

Output :Get Service

TRAITEMENT DESCRIPTION: The customer arrive, explain his demand and is oriented towards the service requested

OPERATION DESCRIPTION
Operation $n^0 6$ Label : recording Treatment Type: manual Periodicity : $8h^{00}$ AM à $4h^{00}$ PM
INPUT/OUTPUT Input: Customer Arrival Output : DR, Billing
TREATEMENT DESCRIPTION: Registration: customer number, customer name, customer name, service requested, number of services requested, unit price and total price.

OPERATION DESCRIPTION
Operation : nº7 Label : Billing Treatment Type: manual Periodicity : $8h^{00}$ AM à $4h^{00}$ PM
INPUT/OUTPUT Input: number of services requested, Descriptions, unit price Output: Total price of each service requested and total
TREATMENT DESCRIPTION calculate: - The total price of each service requested = number of service * Unit price - General total (or payable) = Total price1 + Total price2 + ... + Total price$_n$

OPERATION DESCRIPTION
Operation : nº8 Label : Payment Treatment Type: manual Periodicity : $8h^{00}$ AM à $4h^{00}$ PM
INPUT/OUTPUT Input : Recipe Output : Receipt
TREATMENT DESCRIPTION : When the customer is billed, he pays (the service rendered billed) and is given a receipt.

OPERATION DESCRIPTION
Operation : nº9 Label : Filling Expense Paper Treatment Type: Manual

Periodicity : $8h^{00}$ AM à $4h^{00}$PM
INPUT/OUTPUT INPUT : Amount, motive, title service, Description, Beneficiary OUTPUT : EXPENSE PAPER
TREATMENT DESCRIPTION : The name, post-name of the beneficiary, amount paid, reason, service title and Description are filled in.

OPERATION DESCRIPTION
Operation : nº10 Label : Filling of Daily Balance Treatment Type: Manual Periodicity: At the end of the day (A few minutes before 4h00 PM)
INPUT/OUTPUT INPUT : Recipe and Expenses OUTPUT : Remainder and Daily Balance Filled
TREATMENT DESCRIPTION : DB : Calculate the total Recipe, Expenses of each Description Remainder: Recipe – Expenses
OPERATION DESCRIPTION
Operation : nº13 Label : Synthesis Treatment Type: Manual Periodicity : At the end of the month
INPUT/OUTPUT INPUT : Activities of 1^e Week, A 2^e W, A 3^e W, A 4^e W OUTPUT : Weekly Activities Report
TREATMENT DESCRIPTION : Week Revenue=Sum (Sum Rev Day/Service) Week Expenses = Sum (Sum Exp Day/serv) Remainder Week = (Week Rev – Week Expe)

NB. This step prepares the material of the queries or views in DBMS.

A description is required for each operation, So many operations, and so many descriptions.

E. Documents textual Description = Interfaces (From Operations textual Description)

In this array we recover all rubrics those are on each document and specify their type (Date, picture…) Formulas is they are calculated, Obtaining Mode (Memorize taped on Keyboard, Calculated).

DOCUMENT DESCRIPTION

Document Name: DAILY REGISTER

Nº	Rubric	Type	Formula or constraint	Obtaining Mode
1	Date Recipe or Expense	Date	–	M
2	Record	Text	–	M
3	Customer Name	Text	–	M
4	Customer Post-Name	Text	–	M
5	Service title	Num.	–	M

6	Service solicited number	N	–	M
7	Bill Unit Price Service	N	SerSo numb*	Calcul
8	Bill Total Price service	N	U.P	M
9	Customer Id	N	–	M
	Service Id		–	

DOCUMENT DESCRIPTION

Document Name: BILL REGISTER

Nº	Rubric	Type	Formula or constraint	Obtaining Mode
1	Bill Date	Date	–	Memorize
2	Number Bill	N	–	M
3	Customer Name	Text	–	M
4	Customer Post name	Text	–	M
5	Service solicited	N	–	M
6	Number	Text	–	M
7	Service Title	N	–	M
8	Service Solicited U.P	N	BillServ Numb	Calcul
9	Bill Service T. P	Yes/No	*U.P	M
10	Bill Observation Paym	N	–	M
12	Customer Id	N	–	M
	Service Id		–	

DOCUMENT DESCRIPTION

Document Name: DAILY BALANCE

Nº	Rubric	Type	Formula or constraint	Obtaining Mode
1	Service Title	Text	–	M
2	Recipe	N	Sum (ReceiptAmount)	Calcul
3	Expense	N	Sum	Calcul
4	Remainder	N	(ExpenseAmount)	Calcul
5	Service Id	N	Recipe-Expense	M
6	Day operation	Date	–	M
			--	

DOCUMENT DESCRIPTION

Document Name: RECEIPT

Nº	Rubric	Type	Formula or constraint	Obtaining Mode
1	ReceiptNumber	N	–	M
2	ReceiptDate	Date	–	M
3	Customer Name	Text	–	M
4	Custom.PosaNa	Text	–	M
5	m	N	–	M
6	ReceiptAmount	Text	–	M
7	ReceiptDescript.	N	–	M
	Customer Id			

DOCUMENT DESCRIPTION **Document Name: Expense Paper**				
Nº	Rubric	Type	Formula or constraint	Obtaining Mode
1	Customer Id	Text	–	M
2	Expense Desc	Text	--	M
3	ExpenseAmo	N	--	M
4	u Expense Id	N	--	M

DOCUMENT DESCRIPTION **Document Name: MOTHLY ACTIVITIES REPPORT**				
Nº	Rubric	Type	Formula or constraint	Mode d'obten
1	Service Title	AN	–	M
2	Week	Date	–	Deducte
3	SumRemainder	N	Sum(Remain) per week per service	d Calcul
5	LineTot	N	Sum (Remainder)per	Calcul
6	ColumTot	N	Service	Calcul
7	CodeService	N	Sum (Expense)per week –	Calcul M

F. Attributes Dictionary (From Documents textual Description)

On the basis of the various descriptions of the documents, a table is constituted from the rubrics on these documents and is called the Data Dictionary, note this Data Dictionary constitute the foundation of entity-Association Model.

This table contains the following columns: the number, mnemonic code, description, type of item and even the origin document.

The mnemonic code is the coding of each item on the various documents; it must be significant, short, without space and without special characters (!@#$%^&*()?><?).

The detail of mnemonic code is made in Description columns. We have some restrictions that the data dictionary should not have non calculated rubrics, non-redundancy, respect for semantic unity (one mnemonic = one rubric), non-synonymous (customer name and client name), non- polysemy (CustName PostName= CustName and custPostName) .

Nº	Mnemonic Code	Description	Type Rub	Obtain. M	Origin. Document
	CustName	Customer Name	AN	M	DR+RP+BR
2	CustPostName	Customer Post name	AN	M	DR+RP+BR
3	ServDescrip	Service Description	AN	M	DR +BR+DB+MAR
4	NumbServBill	Number service Billed	N	M	DR +BR

5	UnPriServBill	Unitaire Prix	N	M	DR +BR
6	TotPriServBill	Service Billed Total Price service	N	Calc	DR +BR
7	BillObsPaym	Billed Bill Observation	O/N	M	BR
8	BillDate	payment	Dat	M	
9	BillNum	Bill Date	e	M	BR
10	ReceiptNum	Bill Number	N	M	
11	RecDate	Receipt Number	N	M	BR
12	ReceitAmount	Receipt Date	Dat	M	RP
13	ReceiptDescript	Receipt Amount	e	M	RP
14	Recipe	Receipt Description	N	Calc	RP
15	Expense	Recipe	AN	Calc	RP
16	Remainder	Expense	N	Calc	DB
17	ExpenPapNum	Remainder	N	M	DB
		Expense paper	N		DB
18	ExpenPapNamBen	Number	N	M	EP
	ef	Expense Paper			EP
19		Beneficiary Name	AN	M	
	ExpenPapPostNa	Expense Paper			EP
	mBen.	Beneficiary Post	AN		
20		Name		M	
	ExpenPapAmount	Expense Paper			EP
21		Amount	N	M	
22	ExpenPapDesc	Expense Paper		M	EP
	RecorExpenDate	Recording Expense	AN		EP
23		Date	Dat	Induit	
24	Week	Week	e	Calcu	MAR
25	SumRec	Sum Recipe		Calcu	MAR
26	LinTot	line Total	Dat	Calcu	MAR
27	ColTot	column Total	e	M	MAR
28	CustomCode	Customer Code	N	M	DR+RP+BR
	ServCode	service Code	N		DR
			N		+BR+DB+M
			N		AR
			N		

LEGEND
DR. = DAILY REGISTER
BR=BILL REGISTER
RP =RECEIPT
EP =Expense Paper
DB=DAILY BALANCE
MAR=MOTHLY ACTIVITIES REPPORT

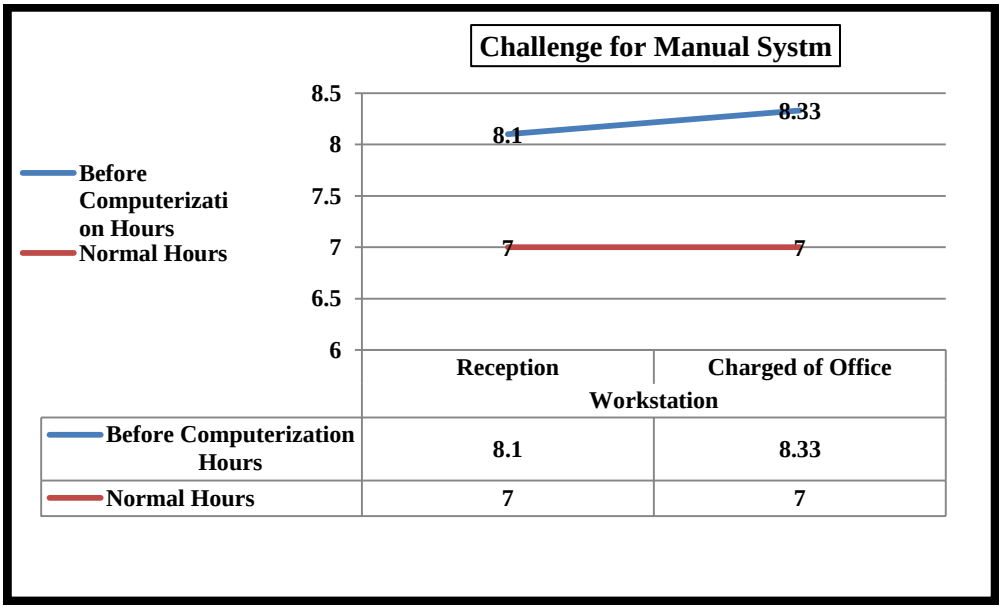
CONCLUSION AND DIAGNOSIS OF EXISTING SYSTEM

In this paper we proposed an approach that integrate analysis, criticism and diagnostic tools of an existing system for a database design in UML approach studying upstream the feasibility, the opportunity, the delimitation of the need expressed by an organization, see even the true's causes of the needs as expressed by users.

In this study, the computerizing it was mentioned in the master plan and a need expressed by users which means that the opportunity is confirmed. In addition company is supplied with well-educated personnel in management and will need to be strengthened in terms of IT tools. Depending on the need and opportunity of the project, the company is ready to mobilize the necessary means for its realization. Even though, after the analysis we find that our project is feasible and timely because the company has more assets than constraints mastered through retraining of the IT personnel.

Likewise, we found in the main workstations that the workers spend more than 8 hours at work without rest, whereas if the recording of receipts and expenses would be done in a database, the production of the Daily Reports would be automatic and would save more time and especially that the continuation of work (the various reports) would be automatic and thus the head of the service would be cleared.

Similarly, the institution has several human and material assets to materialize the design and implementation of this database project. It is only after the studies of existing system that one can think about the database design and its implementation.



This graphic show the real need expressed by users that is true, the works ours is the main challenge to be done in the design step.

REFERENCES

1. Dennis A., Haley Wixom B. at. Al., System analysis anddesign. , Fifth Edition, John Wiley & Sons, Inc., 2012.
2. Michel *Diviné*, Parlez-vous MERISE? , ed. Eyolles, 1994.
3. Muhindo Masivi Osée, Analyse et Conception des Systèmes Orientés Objets Avec UML, Editions Universitaires Européennes, 2018.
4. Heyde Fien V, Laurent D, UML 2.5 Initiation, exemples et exercices corrigés. Eni- Ressources informatiques, 2016.
5. Gomaa, 2009.
6. Rumbaugh, Jacobson & Booch, Unified Modeling Language Reference Manual. (Paperback), the 2nd Edition, 2008.
7. Audibert, UML 2 - De l'apprentissage à la pratique, Cours et exercices. Ellipses-Info+, 2014.

8. Clave A. UML Guide pratique au service de l'analyse des métiers. Eni-DataPro, 2016.
9. Roques P., les Cahiers du Programmeur UML2, Eyrolles, Paris, 2008.
10. Guide A D'UML à C++, Passage d'une conception UML à une implémentation C++. Ellipses-Références sciences, 2013.
11. 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
12. Debrauwer L., UML et Java - Conception et réalisation d'une application Web de Laurent DEBRAUWER, Broché, 2012.
13. Debrauwer and Karam, UML 2 - Entraînez-vous à la modélisation, ENI, Second Edition, 2009.
14. Gabay J, David G., UML 2 - Analyse et conception Mise en œuvre guidée avec études de cas. Dunod-InfoPro - Etudes, développement et intégration, 2008.
15. Miles R, Kim H., Introduction à UML 2 Une introduction pragmatique à *UML*. O'Reilly
16. Sidnei Wazlawick R., Object-Oriented Analysis and Design for Information Systems Modeling with UML, OCL, and IFML, Elsevier, USA; 2014.
17. Bersini H., L'Orienté Objet, Eyrolles, Paris, Third Edition, 2007.
18. Boucheny (2016) [19]
19. Bouzeghoub M., La conception objet des systèmes d'information. Hermès - Lavoisier, 2006.
20. Gilles Roy, Conception de bases de données avec U M L, Presses de L'Université du Québec, 2009.
21. Craig Larman, Applying UML Patterns: An Introduction to Object -Oriented Analysis, Design and Iterative Development, Broché, 2015.
22. Leszek Maciaszek, Requirements Analysis and System Design: Developing Information Systems with UML, Pearson, 2001.