Numéro d'ordre : 17/11

Thèse de doctorat

Pour obtenir le grade de Docteur de l'Université de

VALENCIENNES ET DU HAINAUT-CAMBRESIS

Et l'Ecole Nationale d'Ingénieurs de Sfax

Sciences et Technologie, Mention : Informatique

Présentée et soutenue par Emna Ben Ayed

Le 27/03/2017, à Sfax

Ecole doctorale :

Sciences Pour l'Ingénieur (SPI)

Equipe de recherche, Laboratoire :

Département Informatique, Thème : INTERaction et Agents (INTERA)

Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines (LAMIH UMR CNRS 8201)

Une approche pour l'évaluation des systèmes d'aide à la décision mobiles basés sur le processus d'extraction des connaissances à partir des données : Application dans le domaine médical

JURY

Président du jury
- BELGUITH, Lamia. Professeur des universités. Université de sfax.

Rapporteurs
- CALVARY, Gaëlle. Professeur des universités. INP de Grenoble.
- MAHJOUB, Mohamed Ali. Maître de Conférences (HDR). Université de Sousse.

Examinateur
- MOUSSA, Faouzi. Professeur des universités. Université de Tunis.

Directeurs de thèse
- EZZEDINE, Houcine. Professeur des Universités. Université de Valenciennes et du Hainaut-Cambrésis
- GARGOURI, Faiez. Professeur des Universités. Université de Sfax.

Co-encadrant de thèse :
- KOLSKI, Christophe. Professeur des Universités. Université de Valenciennes et du Hainaut-Cambrésis.

ACKNOWLEDGMENT

Firstly, I would like to express my sincere gratitude to my supervisors Prof. Houcine EZZEDINE and Prof. Faiez GARGOURI for the continuous support of my Ph.D study and related research, for their patience and motivation. Their guidance helped me in all the time of research and writing of this thesis.
My sincere thanks also goes to Prof. Christophe KOLSKI. Without his precious support and immense knowledge it would not be possible to conduct this research.
Besides, I would like to thank the rest of my thesis committee: Prof. Lamia BELGHITH, Prof. Gaëlle CALVARY and Dr. Mohamed Ali MAHJOUB,and Prof. Faouzi MOUSSA for their insightful comments and encouragement, but also for their questions which incented me to widen my research from various perspectives.

I equally want to extend sincere appreciation to Dr. Mounir BEN AYED for his encouragement and support throughout the elaboration of this project. Also, I would like to express my gratitude to Mrs. Leila MAHFOUDHI.
To all my family members, prayerfully supporting me in this work, I am greatly indebted to you all. I really love and appreciate you.

Contents

List of Figures

List of Tables

INTRODUCTION

Even highest towers begin at the ground.
(Chinese proverb)

General context

The domain of the decision-making support is particularly growing. In the middle of the 1980s, Decision Support Systems (DSS) appeared. These systems allow the easy access to the data and supply one or several decision-makers with the required indicators and analysis in order to support them in making the appropriate decision(s).

The research in the field of DSS had, as consequence, the appearance of new technologies and concepts concerning the storage, the treatment and the analysis of data as well as information necessary for the decision-making support. As a consequence, the technology of extraction of knowledge from data occupies a more important square. In this context, our work is interested in the DSSs that are based on the most known process of discovery of knowledge, which is the Knowledge Discovery from Data (KDD) [71]. A

DSS based on KDD process (DSS/KDD) is a system which allows the resolution of a problem of decision-making through a data mining technique. In this process (KDD), there are several important stages such as the needs analysis of the decision-makers, the preparation and the manipulation of the relevant data, as well as the integration of knowledge for the decision-making support.

The progress of mobile technology and the wide availability of the personal mobile devices create a new class of DSS known under the name of mobile DSS (MDSS), which offers the users the possibility of making appropriate decisions at any time via their mobile devices, regardless of the location. Developers have tried to develop MDSS in several fields of application. These systems allow the users to easily manage the knowledge base and allow fast and effective decision-making. As presented in Figure 1, this work is interested in MDSS that are based on the KDD process (MDSS/KDD).

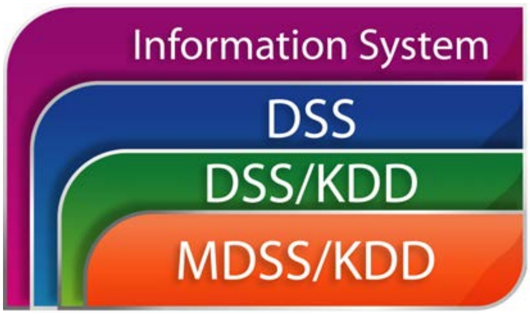

Figure 1: The context of our research work

Our work deals not only with the evaluation of these systems, but also to the evaluation in the KDD process itself.

Motivations

Evaluating systems based on KDD is presently bounded in the KDD process [74] [71]. This process adopts a centralized evaluation module localized after the Data Mining which is the focal module that generates patterns from large data bases. This evaluation module is provided to verify whether the patterns generated from the DM module are interesting.

Although these last years have seen an increased interest within the research community in the evaluation of interactive systems, MDSS have not had a sufficient focus. Few researchers have underlined this gap and defined the criteria which must be measured and optimized to obtain a better quality of DSS. Nevertheless, MDSS has always been considered as either a DSS or an interactive system. Besides, although previous research works pertaining to the KDD process have clearly shown that each module in KDD should be designed, implemented and assessed [32], their proposed evaluation is, from our point of view, incomplete as it neglects several quality factors such as quality in use, quality of data, etc. These works remain limited to the DSS that are not mobile and still concentrate only on the evaluation of the data mining stage.

Problem statement

In this context, it is important to propose an approach that allows an enhanced evaluation in the KDD process and that takes into account the mobility aspect that characterizes an MDSS. Therefore, the problem tackled in our thesis is the following:

How should we enhance the KDD support to a better evaluation of an MDSS/KDD, while taking into account its mobility aspect?

Objectives

The KDD process, which was proposed by Fayyad in [71], is the most known and most used process for DSS development [32]. We are only interested in a particular technology used for the decision-making: MDSS/KDD. The main concern of this piece of research is the evaluation of these systems.

The approach we propose appends an evaluation support module for each module composing the KDD process based on quality models.

The proposed evaluation support modules allow to evaluate not only the quality in use of each module composing the KDD process, but also other criteria that reflect the objective(s) of each KDD module. Our main goal is to help evaluators to detect defects as early as possible in order to enhance the quality of all the modules that constitute an MDSS/KDD.

The outline

This thesis is composed of five chapters.

The first chapter aims at presenting the key notions. So, we begin by presenting and defining DSS as well as its composition and evolution. The interest afterward will only be in the MDSS, a new generation of DSS. Then, we present three examples of Knowledge Discovery from Data processes with a focus on a particular process that represents a link between decision support systems and systems of knowledge discovery, which is the KDD process.

In the second chapter, we introduce a general overview on the evaluation of interactive systems, as DSSs are often highly interactive. We begin with the discussion of the evaluation in development processes in software engineering as well as in the field of human-computer interaction. Afterward, we present the basics, some theory and standards regarding the quality measurement. Finally, we present a state of the art about the evaluation of decision support systems. We finish this chapter by a synthesis and a conclusion.

In the third chapter, we present our contribution regarding the KDD process. This proposal allows a more enhanced evaluation of all the KDD processes in order to evaluate all its modules (Data acquisition and storage, Data Mining, and knowledge management). This contribution is detailed in the first part of this chapter. In the second one, we present a context-based method that takes into account the change of context of use due to mobility. In the third section, we propose an evaluation support system that monitors and measures all criteria detailed in the first part of this chapter.

In the fourth chapter, we present the implementation of the proposed approach. We present all the realized developments to put into practice the proposed approach. These developments concern mainly the tool of evaluation called: Contextbased EVAluation support System for Mobile decision support systems based on KDD process (CEVASM). It contributes to the existing tools by offering not only remote support but also detailed and summarized synthesis of the obtained measures of evaluation. This chapter contains three sections. We begin with the presentation of the developed system and its main objectives. Then, we describe our evaluation process adopted from the standard ISO/IEC 25040 as well as the developments allowing the creation of CEVASM. Finally, we conclude this chapter by a conclusion concerning the realized work.

In the fifth chapter, the approach we propose is applied for the evaluation of the Modules of an MDSS/KDD for the fight against nosocomial infections, representing one of the major problems in the intensive care unit of Habib Bourguiba hospital o Sfax, Tunisia. For every module in KDD, we are interested in the phases of evaluation. We follow the evaluation process, defined in Chapter 4 and based on the standard ISO/IEC 25040. The objective of this chapter is to be able to validate, *a priori*, the realized evaluation tool (CEVASM), and consequently the proposed approach. This chapter is structured as follows. In the first section, we present the general context of the work by introducing the nosocomial infections and the previous proposed systems dealing with this problem. In the second section, we present the MDSSM/KDD to be evaluated using our evaluation support system. Then, we present a discussion which concerns the risks of validity of our led work. This discussion opens several perspectives of research with the aim of the improvement and the extensibility of the proposed approach. These perspectives are drawn in the last part of this chapter.

1. STATE OF THE ART: MOBILE DECISION SUPPORT SYSTEMS AND KNOWLEDGE DISCOVERY FROM DATA PROCESS

1.1 Introduction

The field of decision support is an important area of the information systems (IS) discipline [14]. This chapter aims to present decision support systems (DSS).

The first part of this chapter pertains to the definition of decision-making support as well as DSS and their composition. The focus will afterwards be only on the Mobile Decision Support Systems (MDSS) as it is an emerging field of research. The fourth part of this chapter will present three examples of Knowledge Discovery from Data processes. This research work will concentrate on a particular process that represents a link between decision support systems and systems of discovery of knowledge.

1.2 Decision support

To *decide* does not correspond to a precise, clearly recognizable phase [9]. A decision is a choice among alternatives based on estimated values for these alternatives [179]. Supporting a decision means helping people to work alone or in groups to gather information, generate alternatives and make decisions. The decision-making is a process, which involves the estimation, evaluation and comparison of alternatives. The objective of this process is to define a space of solution answering a given problem, a need to satisfy or a wish of improvement, change or adaptation by taking into account diverse constraints [9, 140]. Thus, the field of decision support is intended to assist the decision maker to understand the situation by proposing choices to be made [9, 214, 57].

Since Scott Morton's works [193], the field of decision support did not stop evolving. DSSs have been developed to support and improve decision making. Numerous definitions of the DSS exist in the literature. These diverse definitions concern either the type of problem of decision [180] or the functions of the system [63], or its constitution [50], or the process of development [32].

In the next subsection, a detailed presentation of this concept is given.

1.2.1 Decision support systems

The concept of decision support system is extremely vast and its definitions depend on the point of view of each author. The DSS can be characterized as interactive computer-based systems that help decision makers to use data and models and solve problems [202]. DSSs have also been developed to improve decision making for complex structured [1] , ill-structured [2] and sometimes unstructured [3] decisions [93].
DSS are defined as follows:

1. According to Turban [211] a DSS is *"an interactive, flexible, adaptable information system and specifically developed to help in resolving a problem by improving the decision-making. It uses data, supplies a simple user interface and allows the user to develop his/her own ideas or points of view. It can use models to bear the various phases of the decision-making and include a knowledge base"*.

2. Frawley *et al.* have also defined a DSS as *"an interactive system that should help a decision maker throughout the decision-making process through appropriate interactions [74]. It consists of tools for measurement, analysis and comparison. It should assist in the evaluation of alternatives."*

[1]Structured Decision Making is an organized approach to understanding complex problems, developing and evaluating creative alternatives, and making defensible choices. It is founded on the idea that good decisions require a rigorous treatment of both facts and values [80].

[2]In this case, different actors in a system tend to perceive the same issue in very different terms. When these different views are conjoined together, a set of inconsistent or contradictory conclusions often follows [150].

[3]each problem is new to the decision maker and has characteristics that are not previously experienced [151].

According to Arnott and Pervan [14], DSS includes personal [4] DSS, group [5] DSS, intelligent [6] DSS, executive information systems [7], data warehousing, and knowledge management-based [8] DSS.

The interactions between the user, the DSS and all tools allow the user to make a decision. For this reason, the interactions between the DSS and the user should be included in the decision-making process [127].

1.2.2 Decision support systems composition

A DSS generally consists of a human-computer interface (also called user interface), a data base, a knowledge base and a model base (see Figure 1.1) [212]. We present hereafter each component:

1. **The user interface:** It allows the establishment of collaboration between the decision-maker and the machine. It is in the center of the DSS and its realization is essential. In fact, a study has shown that at least 50 % of the code of the interactive applications corresponds to the HCI and 50 % of the time of the development is spent on its setting-up [153]. Through the user interfaces, the decision-maker reaches the data and the functions of calculation of DSS. Once the manipulations required by the decision-maker are made, the system sends him/her back the results via the user interface [134]. A Human-machine interaction has to allow the presentation of the information under various forms (2D or 3D graphs, texts, video or other). It also has to supply

[4] small-scale information systems that are normally developed for one manager, or a small number of managers for an important decision task [13].

[5] dedicated for groups of people where each user delegates to an agent that represents his/her preferences and argues with other agents to obtain the best alternative for the whole group [179].

[6] such as: text analytics and mining-based DSSs; ambient intelligence and the internet of thing-based DSSs; biometric-based DSSs; recommender, advisory and expert systems; data mining, data analytics, neural networks... [110].

[7] systems used by the organization for decision making by executive managers [138].

[8] systems that support decision-making in aiding knowledge storage, retrieval, transfer and application, by means of supporting individual and organisational memory and inter-group knowledge access [12].

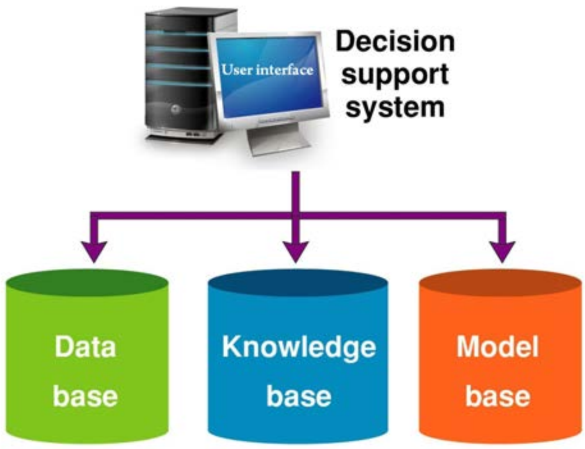

Figure 1.1: Decision support system composition

a help to the user to have a successful conclusion. Its task is also to guide the user by means of precise and flexible examples to adapt itself to the needs of a large number of users.

2. **The data base:** It has the function of memory. It does not only store the data, in a permanent or temporary way, but also manages the recording of volatile data as well as the disappearance of the same data according to the user's wish. These data correspond to the results obtained during data processing. The data we consider are the statistical information or other data which describe the current and past situations. Among these data, it can also build estimations concerning the evolution of certain environmental parameters [32].

3. **The model base:** It consists of a set of models and its management system. The models can be: tools of operational research, statistical models or other. To have more flexibility, a DSS has to possess several models. In this regard, the DSS organizes the passage of parameters between the various models [128].

4. **The knowledge base:** It is a computer-processable collection of knowledge [178]. It includes a set of knowledge on the domain of the problem, on the models and on the strategies of constructions of the models. It helps in the resolution of the problem of decision during all the phases of the process. It introduces the notion of learning into the

DSS. The knowledge base can also play in certain cases the basic role of models.

1.3 Mobile Decision Support Systems

Over the four decades of its history, the DSS field has evolved from personal DSS to Group decision support system (GDSS) [77]. It supports group or collective decision-making by combining communication techniques, computer techniques and decision support techniques, AI and reasoning techniques, and structuralization group decision methods [230]. Then, a new comprehensive decision system was appeared, which is oriented toward decision-makers and the decision-making process, i.e., intelligent DSS that helps in solving the hard problems using artificial intelligence methods and tools. In 2000, the concept of Mobile DSS (MDSS) was introduced in the *ICA3PP* conference [225]. Furthermore, the first MDSS was developed in September 2001 [223]. MDSS can be used through different mobile devices, such as smartphones, PDAs, laptops, tablets and others, whose characteristics are presented in the next section.

1.3.1 Mobile devices

A mobile device, in general, can be defined as a small, lightweight, portable and convenient electronic device with a screen as a display and an input tool such as a keyboard and/or touch screen [133]. Mobile devices need to be small, lightweight and portable [47] to be easily carried and moved by the users. According to Junglas *et al.* [107], the main strengths of mobile devices relate to their mobility and portability.

Modern mobile devices are also expected to provide sufficient wireless Internet coverage [107]. Moreover, they are usually able not only to connect to the Internet using wireless capability, but also to provide and query information using a standard protocol [77]. That is why, wireless communication networks have become a fundamental part of modern mobile devices. Pu-

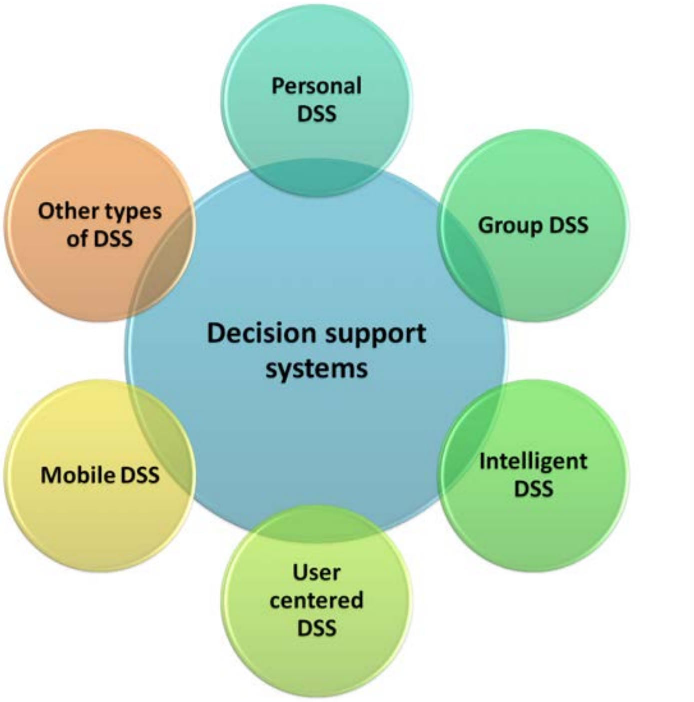

Figure 1.2: Diffrent types of Decision support systems

uronen and Savolainen have also observed that proper mobile devices need to have a certain communication standards [175]. For that reason, data from one mobile device need to be transferable and readable by another mobile device using a standard protocol, and vice versa. With standard wireless Internet connection, mobile device users have ubiquitous access to information, services and the exchange of information.

Mobile devices need to have input and output devices. With the former, users can give commands and communicate with the devices. Traditional mobile phones usually have a small keyboard as their input parameter. With the emergence of touch-screen technology, modern mobile devices do not need a keyboard and users manipulate the screen via touch features on the most advanced input devices [52]. As regards the output devices, the screen is the most common for mobile devices [77].

Although mobile devices have superior mobility and multi-functional capabilities, there are a number of associated inferior aspects, including relatively small screens, less computational power and shorter battery life [108]. It is notable that there is a continuous spread of new mobile devices, as well as increases in computing power, longer battery life, improved screen resolution and other quality enhancements. Yet, the particular characteristics of mobile devices should always be taken into consideration when designing, implementing and evaluating applications for mobile devices.

In fact, all mobile devices contain some key attributes that offer the opportunity for the development of new applications that are possible only in the mobile environment.

1.3.2 Mobile devices and mobile decision support systems

Mobile devices can take different forms. Mobile Decision Support Systems (MDSS) can have any form of mobile device if it meets the definition of mobile device and supports or improves the user's decision-making [77]. Thanks to the following reasons, it would be considerable to adopt an MDSS in an organization/Company [218]:

- The growth in the number of mobile subscribers is expected to surpass the number of fixed subscribers at some point in the near future.
- Globalization and information technology have altered business management and competitive styles.
- Nowadays, many companies need to manage and control their organization in a global marketplace via the Internet, since most businesses face global competition.
- Managers sometimes hold meetings to communicate or give management instructions to their subordinates. If there is nothing new to report, all data related to working progress or accomplished performance can actually be found in a database and accessed through an information system.

Based on the analysis of MDSS publications in nine prominent information systems journals which are the A* journals identified by the Australian

Council of Professors and Heads of Information Systems (ACPHIS), the search term '*mobile*' extracted 335 papers which represent only 3.12% of the total journal article population [77]. By filtering the articles relevant to mobile decision support content, only 32 MDSS papers were identified, which constitutes 0.30% of the published articles (see Figure 1.3).

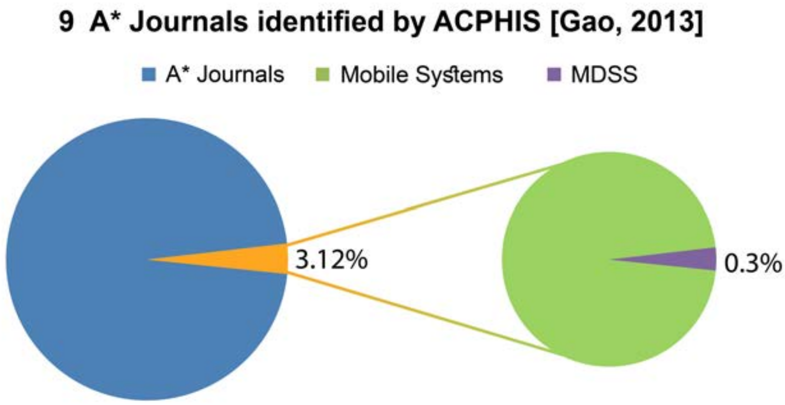

Figure 1.3: MDSS reasearch

This can be further interpreted by the fact that although many people have chosen to incorporate mobile information systems computers into their daily life and/or work routine, very limited IS research [162] has been conducted on tablet computers, probably due to the short time frame since their rise in popularity. According to Gao, MDSS research is dominated by personal DSS (65.625%) [77]. Intuitively, as mobile devices are very personal devices, the DSS designed for mobile devices are for individual users, so this result should come as no surprise. Moreover, the popularity of personal DSS is quite consistent over the years in MDSS research.

1.3.3 Possible architectures of mobile decision support systems

The improvement in computational device miniaturization and in wireless communication has moved forward relevant advances in mobile systems development. Such advanced systems offer new functions to support users'

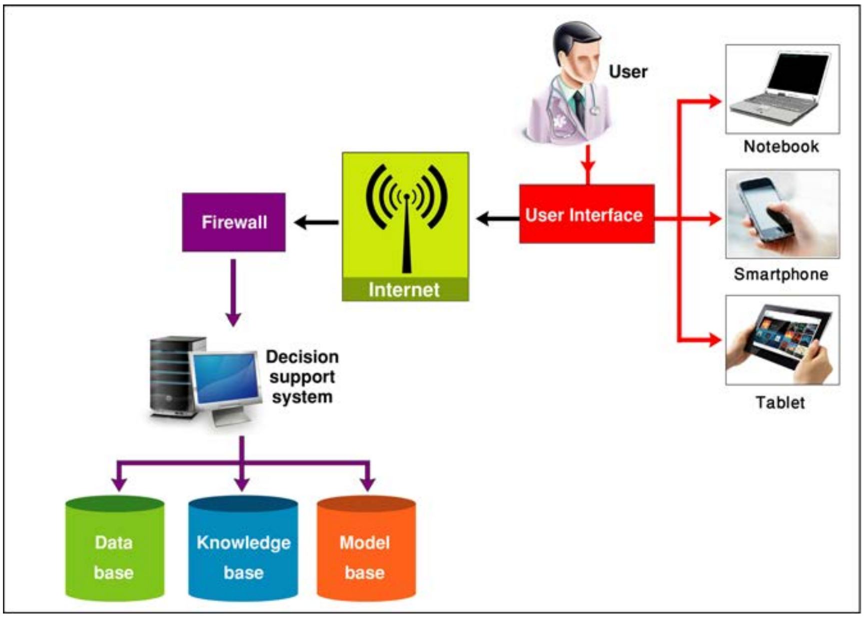

Figure 1.4: Operation client/server structure of prototype of MDSS

daily activities and can be everywhere around them [42]. This support should be executed without users that need to be aware of their interaction with various technologies.
Several architectures have been proposed in the literature to allow developers to implement MDSS. In the next sub-sections, we present the most substantial ones.

1. **Client/server architecture:**

   Through this architecture, the end user can receive and/or send information thanks to Internet technologies (see Figure 1.4). The server is the most important part of the DSS [168], as it controls the database that stores all the data of a problem, recommendations, parameters, etc. The client device in the client-server architecture depends on the server for processing activities. There is always communication between the user and the remote server. This architecture is considered to be used on several mobile devices, such as GSM [9] or UMTS[10] mo-

[9] Global System for Mobile Communications
[10] Universal Mobile Telecommunications System

bile phones or PDA[11] devices that use mobile network infrastructures and mobile messaging services. Clients (such personal computers, or mobile devices) and servers (powerful personal computers) are both connected by a network enabling servers to provide different services for the clients. Indeed, when a client sends a request to a server, this server that contains the database, processes the request and sends a response back to client [168]. To establish a secured communication, the client as well as the server should authenticate each other [181]. There are lots of issues in a client-server system, some of which include:

- A client server network is quite difficult to set up, so it requires lots of servers so as not to render the application useless.
- Setting up a client server network is so complex, so it requires skilled technicians and maintenance engineers to handle it.
- The client operating system is easily accessed by servers (security issue).

2. **Cloud computing**:

It has recently appeared as a paradigm that can offer support with technological benefits for end users. Cloud computing is definitely at the top of the technology trend. This trend is enforced by providers such as Amazon, Google, SalesForce, IBM, Microsoft, and Sun Microsystems, which have begun to establish new data centers for hosting Cloud computing applications such as social networking (e.g. Facebook), gaming portals (e.g. BigPoint), business applications (e.g., SalesForce.com), media content delivery, and scientific workflows [46]. The advantage of cloud computing is that it is capable of offering a cloud-based DSS service to meet the emergency users' decision needs [205]. In addition, it can allow access and service flexibility both for service users and service providers. Numerous providers offer designers the possibility to make web-based mobile applications more easily and effectively [70]. For instance, Google App Engine offers a complete development stack that uses familiar technologies to build and host web applications [46, 104]. Some service providers treat requests from

[11] Personal Digital Assistant

their customers with priority while others apply *first-come, first-served* policy to all requests [103]. Several researchers, such as Miah [147], have proposed DSS application by the provision of cloud computing. Miah's approach, for example, provides a domain-specific decision support to the decision makers through cloud-based functionalities on an 'anywhere-anytime' basis. Figure 1.5 illustrates his approach.

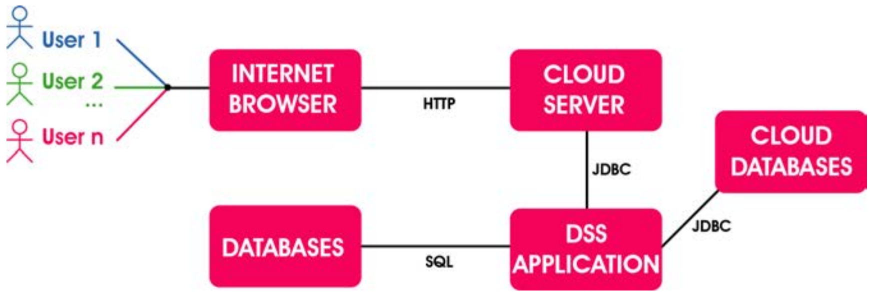

Figure 1.5: System units for a cloud DSS [147]

1.4 Knowledge Discovery from Data processes

Due to the growth of the data sources number as well as the quantity of data in those sources, it becomes necessary to develop systems that are able to extract, automatically or semi-automatically, knowledge hidden by the complexity of the data. The main causes of this complexity are the heterogeneity, diversity, dispersion of the huge number of data. According to Frawley [74], the main purpose of the Knowledge Discovery from Databases (KDD) is to find Knowledge in the data flood.

Any discussion of a KDD process must be preceded by defining the terms to be used: data, information and knowledge. It would be worthy to mention the definitions that have been proposed in the literature especially in the cognitive sciences. We can refer only to those accepted generally in the computer science field and presented by Habert as follows [81]:

- Data are the results of observation.

- Information are the results of interpreting those data, answers to "who", "what", "where", and "when" questions.

- Knowledge defines how to use the Data and the Information, answers "how" question.

Knowledge Discovery from Data (KDD) refers to a set of activities designed to extract new knowledge from complex datasets. The KDD process is often interdisciplinary and spans computer science, statistics, visualization, and domain expertise. In recent years, large quantities of data have become increasingly available at significant volumes. Such data have many sources including online activities (social networking, social media), telecommunications (mobile computing, call statistics), scientific activities (simulations, experiments, environmental sensors), and the collation of traditional sources (forms, surveys). Consequently KDD has become strategically important for large business enterprises, government organizations, and research institutions.

However, effectively producing knowledge from datasets remains challenging, especially for large enterprise organizations composed of multiple sub-organizations (each of which may have its own internal processes, formats, etc.).

Effective KDD, therefore, requires effective organizational and technological practices to be in place. Specifically, knowledge discovery processes are composed of:

- Data collection, storage and organization practices;

- Understanding and effective application of the modern data analytic methods (including tools);

- Understanding of the problem domain and the nature, structure and meaning of the underlying data.

1.4.1 CRISP-DM

Osei-Bryson and Kweku-Muata explain that "CRISP-DM (cross-industry standard procedure for data mining) was developed by multi-industry collective of practitioners after the practitioner community became aware of the need for formal data mining process models that prescribe the journey from data to discovering knowledge" [161].

The CRISP-DM process model includes six steps (1.6): business understanding, data understanding, data preparation, modeling, evaluation and deployment [170]. One of the main limitations of the CRISP-DM life-cycle representation is that it is essentially sequential and linear. This sequential nature of the representation suggests an ordering of the knowledge space, and its exploration, which does not appropriately characterize the hierarchical and the interactive network features of enterprise knowledge space, or the dynamics of knowledge discovery.

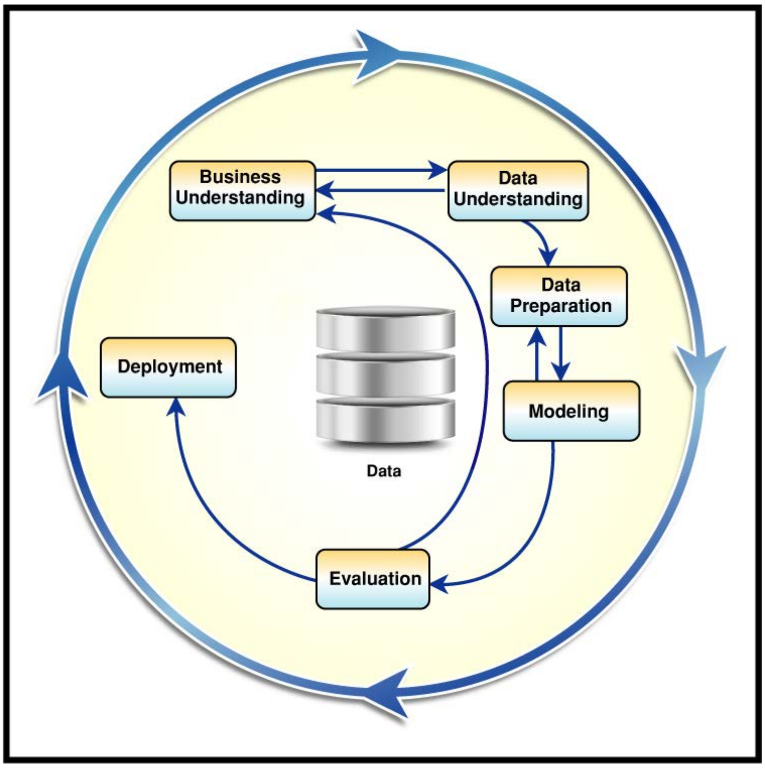

Figure 1.6: The CRISP-DM

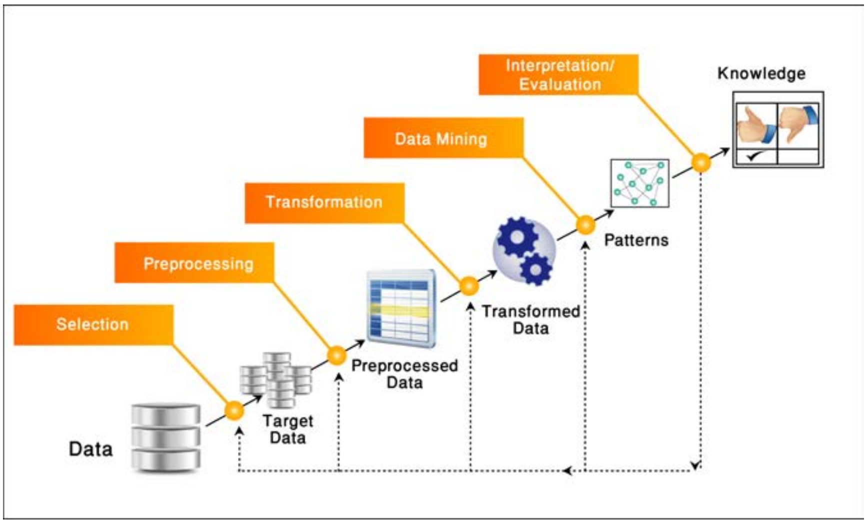

Figure 1.7: KDD process according to Fayyad [73]

1.4.2 Fayyad's KDD process

According to Benjamins, Information is not knowledge [34]. Knowledge can be extracted using a decision-making tool based on the Knowledge Discovery from Databases (KDD) process [73]. The process of KDD could be defined by a sequence of process operations and data analysis (see Figure 1.7).

The aim of KDD is to retrieve knowledge. As a definition, it is also described as *"extraction of new knowledge, useful, valid from a mass of data"* [73].

Historically, the notion of finding useful patterns in data has been given a variety of names, including data mining, knowledge extraction... [72]. The term data mining gained popularity and has mostly been used by statisticians. The concept of knowledge discovery in databases was coined at the first KDD workshop in 1989 [172] to emphasize that knowledge is the end product of a data-driven discovery. It has been popularized in the Artificial Intelligence and machine learning fields.

In line with Figure 1.7, the KDD process follows these steps face to an already-known problem:

- (0) Identifying objectives, setting targets and checking requirements, Research data (identifying information and the sources),
- (1) Selection of data relevant to the analysis requested in the database,
- (2) Cleaning data to correct inaccuracies or data errors,
- (3) Transformation of data in a format that prepares them for the Mining (convert dates in duration, ratios, etc.).
- (4) Data mining, application of one intelligent methods or more, such as neural network, Bayesian networks, decision trees, etc., to extract interesting patterns,
- (5) Evaluation of the results to estimate the quality of the model discovered,
- (6) Integration of knowledge by implementing the model or its results in the computer system of the company.

These modules are related since they compose the KDD process. However, each module has its individual objectives. Thus, the design and creation of each module can be done in parallel or in overlap with the other modules of the KDD-based DSS.

1.4.3 The derived Fayyad's process

Ben Ayed *et al.* [32] and Ltifi *et al.* [136] consider that there are four main phases that form the KDD process, as shown in Figure 1.8. In line with Figure 1.7, the KDD process follows the following steps:

- (1) The data acquisition and storage step which consists in selecting, cleaning and transforming data in a compatible format for the next phase;
- (2) Data mining at the focal step of the KDD process: it allows the extraction of relevant and interesting patterns (non-trivial, implicit,

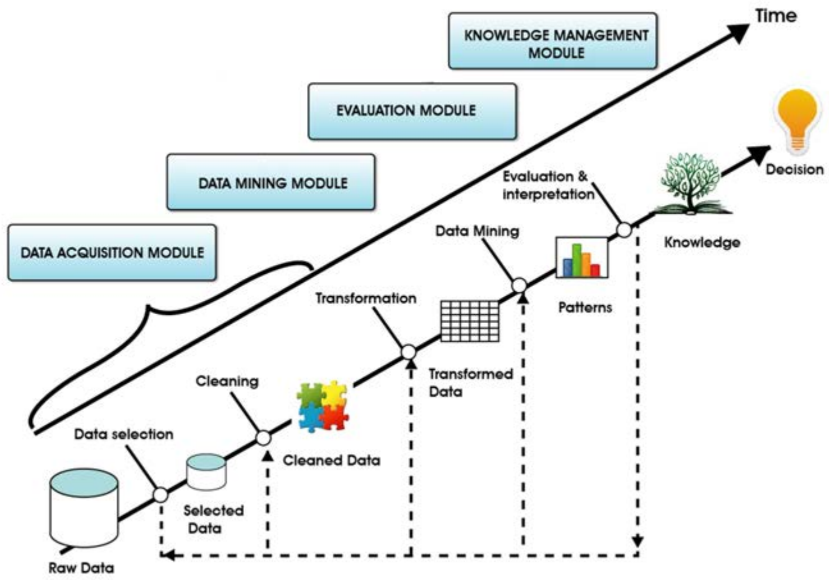

Figure 1.8: The derived KDD process [32, 136]

previously unknown and potentially useful) from large quantities of data by applying intelligent methods;

- (3) A post-processing module is important to interpret and evaluate the patterns provided to get Knowledge Units (KU). This module is called "evaluation and interpretation", during which the generated patterns are interpreted and evaluated for the knowledge integration in the decision making stage.
- (4) These KU are to be modeled, stored and shared to help the decision makers get the best action to do. Besides, these KU are fruitful on the next iterations of the KDD process.

Data acquisition and storage step

As documented in [143] and [122], this module encloses several stages.

- 1) Understanding of the problem domain: it includes not only the problem definition from the user's viewpoint, but also the translation of the problem into data mining goals to select the data mining algorithm(s) to be used later in the process [137].

- 2) Creation of a target data set: this stage concerns collecting sample data and deciding which data will be useful in the data mining method.
- 3) Preparation of the data: It includes data cleaning through basic operations (check the completeness of data records, remove or correct noise and missing values, etc.). It also includes data transformation to reduce dimensionality (data discretization and granularization) [137].

Data mining step

Data Mining (DM) is a step in the KDD process that consists of applying data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns (or models) over the data such as classification rules, association rules, or summaries. According to Fayyad [73], a pattern is an expression describing a subset of the data. While this step is presently attracting the attention of researchers, the next step "*evaluation*" is often neglected [77]

Evaluation step

According to Fayyad [71], evaluation is considered as a centralized module, through which only the extracted pattern is evaluated and interpreted. Based on data mining algorithms [222], the patterns that should be evaluated are extracted, and from these patterns, the potential concepts are extracted. Additional important concepts can be proposed by the expert either from his/her own knowledge and experience or from interpretations to the yielded patterns helping him/her to discover the hidden knowledge from the huge number of data and variables. So, the process of knowledge discovery yields to get two knowledge sources: the Data Mining engine and the expert.

Patterns mined from the data can be represented in different forms, such as classification rules, association rules, clusters, sequential patterns, etc.

Data Mining is known for its capabilities of offering systematic ways for giving useful rules and patterns from large amounts of data. However, there is a continuous flow of data, and therefore patterns [17], some of which are not considered as interesting for the application in use. In fact, patterns should have some degree of certainty. So, interestingness measures are important in the context of DM, regardless of the pattern's form. These measures are intended to select and rank patterns according to their potential interest to the user [79].

Association rule algorithms, for instance, usually generate too many rules. So, many researchers, such as Silberschatz *et al.* [199] and Baena-Garcia and Morales-Bueno [17], have focused on finding interesting recommendations for users.
An association rule has two parts, an antecedent (if) and a consequent (then). An antecedent A is an item found in the data. A consequent B is an item that is found in combination with the antecedent. Let(r: A→B) an association rule extracted from a database. A measure of interest is a function that associates a real number characterising the interest of this rule with an association rule.

Objective measures of interest are values which are determined by the contingency table of r. In fact, Figure 1.9 shows such a contingency table, in which we note P(x), the frequency of the pattern X.

	B	$\bar{B}$	
A	P_{ab}	$P_{a\bar{b}}$	P_a
$\bar{A}$	$P_{\bar{a}b}$	$P_{\bar{a}\bar{b}}$	$P_{\bar{a}}$
	P_b	$P_{\bar{b}}$	1

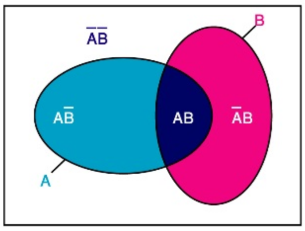

Figure 1.9: Contingency table

There are many substantial previous research works, surveyed in [79] and [17], in which several criteria were proposed to verify the strength of the generated patterns before presenting them to the user (i.e. decision-maker). Support and confidence [6] are the most used criteria. Therefore, the support and confidence should be augmented with a pattern evaluation measure,

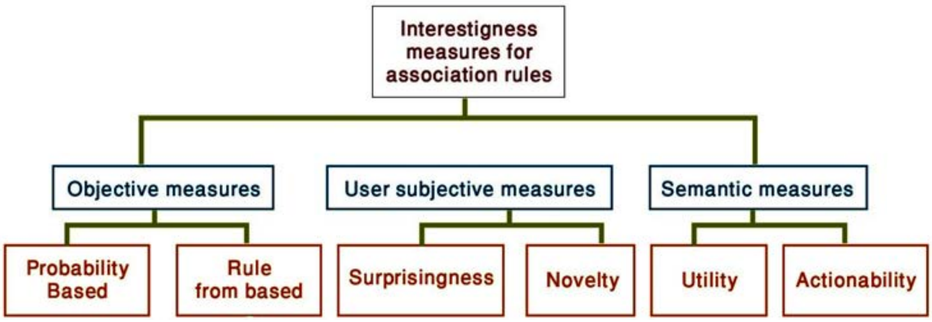

Figure 1.10: Categorization of Interestingness measures for Association Rules according to [79]

which promotes the mining of interesting rules. Fayyad *et al.* (1996), in [73], defined a set of criteria for interestingness. Later, Geng and Hamilton, in [79], classified those criteria in three classes, as shown in Figure 1.10. Objective measures are based only on the raw data while user-subjective interestingness measures are based on both data and the knowledge of the expert using these data. **Semantic interestingness measures** emphasise the semantics and explanations of the patterns [79].

Objective interestingness measures are based on probability theory, statistics and information theory. These measures take into account neither the context of the domain of application nor the goals and the background knowledge of the user. However, subjective and semantics-based measures incorporate the user's background knowledge and goals, respectively. They are both suitable for more experienced users and interactive data mining. Recently, Ltifi *et al.*, in [136] distinguished two types of evaluation (see Figure 1.11):

- Evaluation of patterns as it was proposed by Fayyad [71], and
- Evaluation of software in terms of the functional and HCI requirements as shown in Figure 1.11.

This proposition is not based on well-known quality models from the literature. From our point of view, ISO quality models, Nielsen's model [157] and Geng's model [79] can be used to establish more efficient evaluation. In

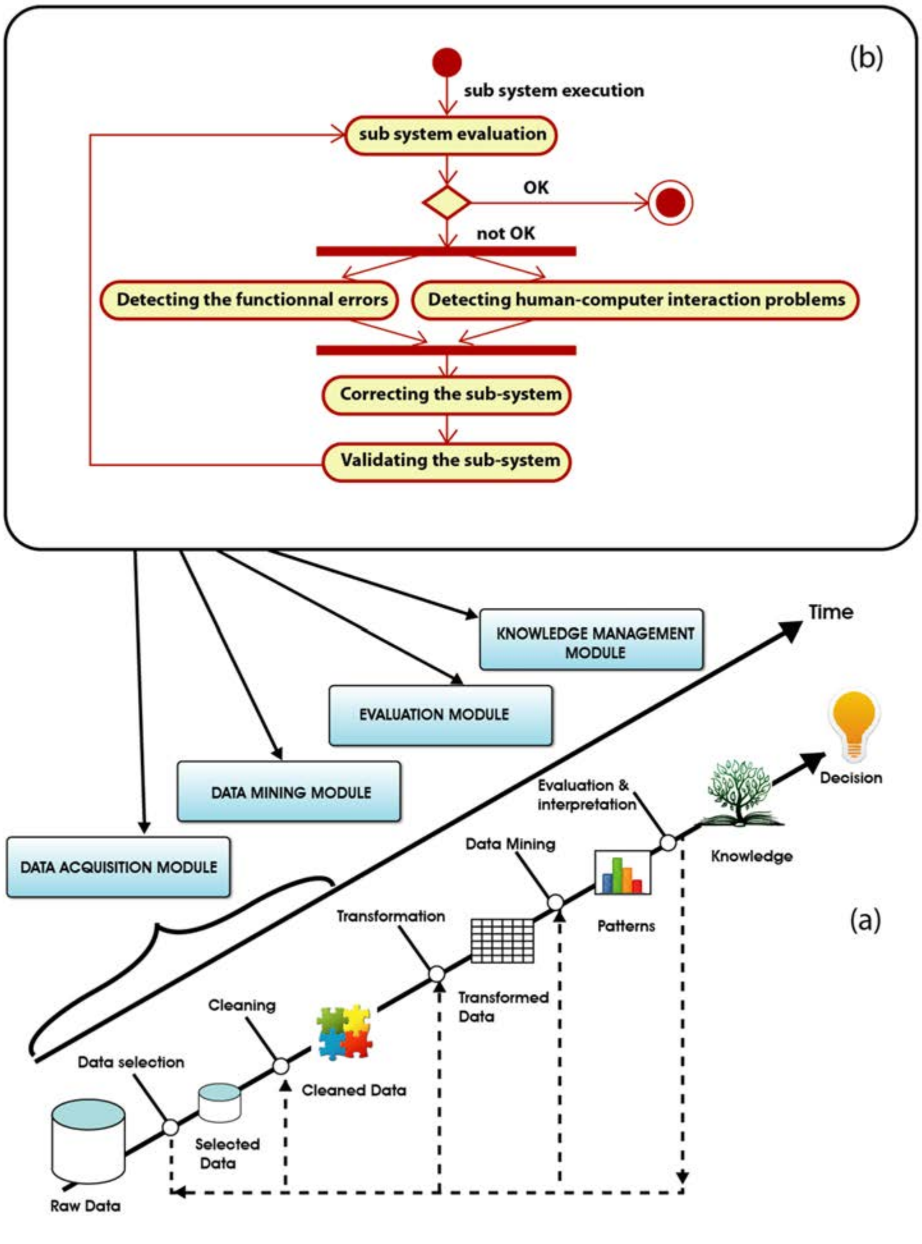

Figure 1.11: Evaluation in KDD process as proposed by Ltifi *et al.* in [136]

fact, these evaluation methods ensure better utility of systems, and guarantee not only the satisfaction of the user, but also the quality of patterns mined. Although, we agree with [136] that evaluation is a main objective that should be achieved in all the steps of the KDD process, we consider that evaluation of Decision support systems based on KDD process is an intention that should be treated by taking into account two main elements:
(1) The specifications of each step constituting the KDD process,
(2) The quality in use of each step.

In chapter 3, we will detail our proposition to ensure an efficient evaluation of all KDD steps.

Knowledge management step

Knowledge Management (KM) is a multidisciplinary subject with contributions from different disciplines such as Information Technology, Information Systems, Strategic Management, Human-resource Management, Cognitive Science, Artificial Intelligence, etc. Although, to the best of our knowledge, there is no universal definition of Knowledge Management in the literature, it is largely regarded as a process involving various activities. Benbya *et al.* [33] define KM as the systematic way to manage knowledge in the organizationally specified process of acquiring, organizing and communicating knowledge, in order to enable organizations reach their performance and goals. Several different approaches dealing with KM process were proposed. In fact, it was regarded as the process of creating (developing new understandings from patterns and relationships between data, information, and prior knowledge units), collecting (acquiring and recording knowledge), organizing (establishing relationships and context to facilitate the access to the collected knowledge), delivering (searching for and sharing knowledge), and using (bearing knowledge on a task) knowledge [176]. According to Miled *et al.* [148], it consists in acting on the discovered knowledge using the knowledge directly, incorporating the knowledge into another system for further action, or simply documenting it and reporting it to interested parts.

Several researchers such as Alavi [145] and Zaim [226] distinguish four main operations in the KM process which are:

- The Identification: allows the selection not only of the valuable knowledge from provided patterns, but also of the model representing the decision made by the expert.
- Preserving: integrates the new learned knowledge in the context (semantic context for example).
- Valorizing: achieves the classification according to learned knowledge and interprets the existing knowledge for future reuse.
- Updating: is the step that allows to add, delete, and modifiy knowledge by either expert or analyst orders in case of redundancy, contradiction, amelioration.

In fact, to add value with knowledge management, there is a need for Knowledge Management Systems (KMS), which are systems that facilitate at least one operation among the list (identification, preserving, valorizing, updating) [224] [145].

It is patently shown that the previous works focused on the identification and valorization of knowledge. However, to the best of our knowledge, no approach currently exists to help in the KMS evaluation, except the work of Ngai [156] that is concerned with providing such solution by giving a quantitative evaluation of a set of quality factors (cost, functionality and vendors). Nevertheless, their work needs to be complemented by taking into account other features such as quality in the use of a KMS. Moreover, knowledge can be stored in different sources [20], so, access control needs to be added to evaluate if end users can exploit their stored knowledge and if they can exchange it.

When we evaluate a KMS, we should keep in mind that rapidly changing data may make previously discovered patterns invalid [54] [197]. So, a continuous evaluation method for updating the patterns and treating changes as an opportunity for knowledge discovery is well needed.

For the reasons cited above, we propose a novel approach that ensures an enhanced evaluation in KDD process and concerns all the KDD steps.

1.4.4 Decision Support Systems based on KDD process

In literature, the purpose of DSS is mainly assisting decision makers to resolve complex problems. Within the general framework of DSS, the KDD has become a research topic that has already amply demonstrated its scientific and economic importance and appears now as a strategic area. Both KDD process and DSS are highly interactive [135]. It is therefore important to understand the user and join the human creativity, flexibility, and knowledge with the huge storage capacity and computing power of computers in the Decision Support Systems based on KDD process. The development of the latter requires a real knowledge on the application domain, which refers to the approach of Knowledge Discovery in Databases (KDD). A link between the decision support systems and knowledge discovery systems can be established (KDD based DSS). This kind of systems allows the user to explore a large amount of data to discover new usefull patterns for decision-making.

1.5 The context of use

We need first of all to define the paradigm of the context of use and its relation to the concept of mobility. Then, we present the approach that can be used to collect contextual data.

1.5.1 State of the art

According to Kakihara and Sorensen [109], the contexts in which people reside continuously frame their interaction with others, including their cultural background, situation or mood, and degree of mutual recognition. In the studies on mobility in work contexts, workers were considered to be mobile. Actually, wandering, traveling, and visiting strongly affect workers, whose mobility is enabled and facilitated by new technologies [141]. This means that the mobility is strongly linked to the context concept.

The context is defined as any information that can be used to characterize the situation of the entities (a person, a place or an object) which are considered as relevant for the interaction between a user and an application, including the user and the application themselves [61].

Abowd defines context-Awareness as the ability of the system to use context to provide relevant services to user, where relevancy depends on the user's task [4]. When a system uses context information to provide relevant services to the user, this system is called context aware. Sottet *et al.* [201] defines the context as a triplet $\prec User, Platform, Environment \succ where:$

- The *User* covers a set of persons (people) that have roughly the same characteristics such as age, knowledge, experience, etc.
- The *Platform* denotes the set of variables that characterize the computational device(s) used for interacting with the system. Typically, memory size, network bandwidth, screen size, etc., are determining factors.
- The *Environment* covers the set of entities (e.g. objects, and events) that are peripheral to the current task(s) but that may impact the system and/or the user's behavior. These include surrounding noise, lighting conditions, user's and objects location, social ambiance.

In order to consolidate the main concepts and proposals related to the state of the art, we propose the notion of context in Table 1.1, which summarizes the main concepts proposed in the state of the art. The table consists of 6 elements. It presents a synthesis of the research works based on the proposal of Sottet and Calvary [201], which defines the context according to the platform, the environment and the user:

- Ref: resumes citation associated
- Author: lists the author or authors

- Date: is the date of the work (publications)
- Platform: specifies which platform the authors have used. In this table, we present only the works that have used mobile platforms
- Environment: defines the environment from the point of view of the authors
- User: indicates the type of the given users

All the definitions cited in Table 1.1 reference the location and the physical environment, the user and a specific platform to define a context of use. However, the authors brought some changes that helped clarify the context with more precision by including other specifications such as the time or the state. We can also notice that recent research (beyond 2010) have not changed the previous proposed definitions.

1.5.2 Contextual data collection

There are two approaches that allow the collection of the information related to the context, which are defined as follows and are further compared in Table 1.2:

- **Direct sensing:** This is often used in applications with in-built local sensors. The client software gathers the desired information directly from these sensors, without an additional layer for gaining and processing data. Drivers for the sensors are hardwired into the application [53].
- **Server of context:** Multiple clients have permitted access to remote data sources. This is a distributed approach that extends the middleware based architecture by introducing an access management component with sensor data gathering function moved to the so-called context server to facilitate concurrent multiple access [152].

Table 1.1: Summary and comparison of the notion of context

Ref	Author	Year	Platform	Environment	User
[191]	Schilit *et al.*	1994	Mobile devices	Where? with whom?	Persons in mobile situation
[43]	Brown	1996	PDA, Computer	Objects in the environment, localisation	Persons
[44]	Brown *et al.*	1997	PDA	Where? with whom?	Persons
[44]	Ryan *et al.*	1997	PDA	Localisation	Persons
[166]	Pascoe	1998	Mobile device	Localisation, time	Persons
[62]	Dey and Mankoff	2005	Mobile device, mobile computer	Environment	Mobile users
[165]	Pascoe *et al.*	2007	Mobile device	Localisation	User
[76]	Ganneau *et al.*	2007	PC, PDA, Smartphone	Environment	Mobile users
[141]	Mallat *et al.*	2009	Mobile device	places and times	people
[187]	Santos *et al.*	2010	Mobile device	Sound, Light, GPS, Temperature, Time, Movement, Humidity	profile
[45]	Burns *et al.*	2011	Mobile device	Location, social environment	User's activity and his/her internal states[a].
[19]	Baltrunas *et al.*	2012	-	-	User preferences
[231]	Zhu *et al.*	2015	Mobile device	Localisation, time	User preferences
[66]	Espada *et al.*	2015	Mobile platform	Physical world and bluetooth	User
[229]	Zheng *et al.*	2016	Android platform	Internet network	User profile and category

[a] his/her mood

Table 1.2: Summary of differences between the approaches allowing the collection of information related to the context

Direct detection	**Context server**
Local approach	Distributed approach
Local sensors	Sensors moved towards the server
Functionnalities	
Contextual data collection	Contextual data collection
No multiple access	Multi-access management

1.6 Conclusion

A wide range of applications, including decision support systems, being supported on mobile devices and users' expectations are progressively improving. Nowadays, MDSS are often used in changing environments, yet do not adapt to those changes very well. Although moving away from the desktop model brings a variety of new situations in which an application may be used, computing devices are rarely aware of their surrounding environments. Thus, information in the physical and operational environments of mobile devices creates a context for the interaction between users and devices. The future of mobility is inherent in the concept of Context. MDSS would rather provide support based on the context of use. Mobility offers the opportunity to gain awareness of the individual and their interactions with their ever changing surroundings. So, a model of raising awareness to the context is necessary to define and store contextual information to be readable by the machine.

Although the existent architectures are providing benefits to their users, some gaps impede the growth and implementation of such technology in real environments. The need for high quality mobile systems becomes more required. So, further research and development projects may cover these gaps and allow the implementation of more pervasive systems. The evaluation of those systems seems to be highly needed.

The next chapter presents previous works in the field of evaluation of interactive systems and discusses methods available in the literature.

2. EVALUATION OF INTERACTIVE SYSTEMS

2

2.1 Introduction

After more than about twenty years of research and practice, it is undeniable that the evaluation of the interactive systems is an essential activity to produce high quality systems [157] [51]. Evaluation is the process that consists in estimating or justifying the value of the evaluated systems [200]. It presents one of the biggest interests of the community of Human-Computer Interaction (HCI). Therefore, a high quality interactive systems offers not only the success in the industry but also the satisfaction of the end user.

Many concepts, methods and evaluation tools were proposed by the community of HCI in order to validate a system already built or under design or development and improve its effectiveness.

There is an increasing number and many types of interactive systems that are being developed for end-users. Among these systems, we find Information Systems (IS). The incorporation of users into IS evaluation has been identified as important concern for IS researchers [115]. However, the evaluation has always been an issue as regards decision support systems which was introduced into the computing and information systems literature [193]. According to [77], there are no precedents to follow from DSS evaluation for evaluating MDSS. We can assert that up to now, this statement is right.

In this work, we are interested only in a particular technology which is Mobile Decision Support Systems (MDSS) used and implemented in several fields. Their potential importance for supporting timely access to critical information involves reducing errors and improving access to all information that was previously centralized [163]. However, some gaps impede the growth and implementation of such technology in real environment. In fact, many requirements such as mobility and context-awareness have to be complied. So, further research and development projects may cover these gaps and allow the implementation of more pervasive systems.

Through this chapter, we introduce the most known works in the field of the evaluation of interactive systems. In the first section, we introduce the most known development processes in software engineering with an emphasis on the evaluation in these processes. In the second section, we establish a state of the art about the evaluation in the field of human-computer interaction including the existing evaluation methods, techniques and processes. In the third section, we focus on the measurement of quality through criteria by presenting the most known theories. Finally, previous research works dealing with the evaluation of decision support systems are given, followed by our point of view regarding this issue.

2.2 Evaluation in development processes in software engineering

The processes or the cycles of software development of software are often general. They can be considered as a way of composing the production of software on a set of phases, describing the process, as well as indicating the logic or the temporal order in which these phases occur. In the following subsection, we present the most known development processes in software engineering.

2.2.1 Development processes in Software engineering

The waterfall model [182] is one of the first models that were proposed to satisfy the industrial needs in terms of software quality and productivity. One of the problems with this model is that it is recommended for use only in projects which are relatively stable and where user's needs can be clearly identified at an early stage. This model was also criticized by Kolski, in [119], as it does not incite to the consideration of the user interface even when the system is highly interactive. Due to the appearance of V model [146], the evaluation has been integrated through an ascending process having for object the validation and the tests. However, it implicitly promotes writing test scripts in advance, rather than exploratory testing; it encourages testers to look for what they expect to find, rather than discover what is truly there.

The agile models for software development appeared in the early 1990s. They include the Rapid application development (RAD) which is a response to the processes developed in the 1970s and 1980s. It is especially well suited (although not limited to) developing software that is driven by user interface requirements. RAD first appears with the publication of James Martin [144] in which he defines the key objectives of RAD as: high quality systems, fast development and delivery, and low costs.
Examples of the best known agile methods are Scrum [192] and XP [22]. Scrum aims at providing an agile approach for managing software projects, while increasing the probability of successful development of software [186],

whereas XP focuses more on the project level activities of implementing and testing software. Both approaches, however, embody the central principles of agile software development [186].

Such development models recommend regular meetings with the customer, delivering an initial product as rapidly as possible and adapting to changing customer needs. But these models do not cover all the steps of a process. In addition the agile methodologies deployment often encounters resistance from systems developer [32]. The general tendency in software development is towards iterative processes such as spiral [40] and the Unified Process (UP) which consists of a set of generic principles that can be adapted to specific projects. UP is thus a process pattern that can be adapted to a large category of software systems, various fields of applications, different types of companies, different qualification levels and various project sizes. [100].

Most of the traditional models are too often directed towards the technical aspects of the system (e.g., the code) and not enough towards user needs. The only real exception to this observation is UP. Even users are relatively involved in the analysis and validation stages for a prototype, the models and processes are generally not accompanied by explanations of their involvement [32]. HCI evaluation principles are not part of these generic processes [134]. From the perspective of interactive system development, the user characteristics must be clearly expressed [32]. as the evaluation is a phase pertaining to the process of system development, he/sheit should not be discarded. In fact, the SE evaluation models have always been judged to take the user insufficiently into account.

2.2.2 Quality models in the field of software engineering

The models quoted previously have a common objective that is the production of high quality software. Some researchers, having focused on the evaluation of software, have defined the software quality engineering field as an emerging discipline that is concerned with improving the approach to software quality. This discipline needs a quality model which embraces all

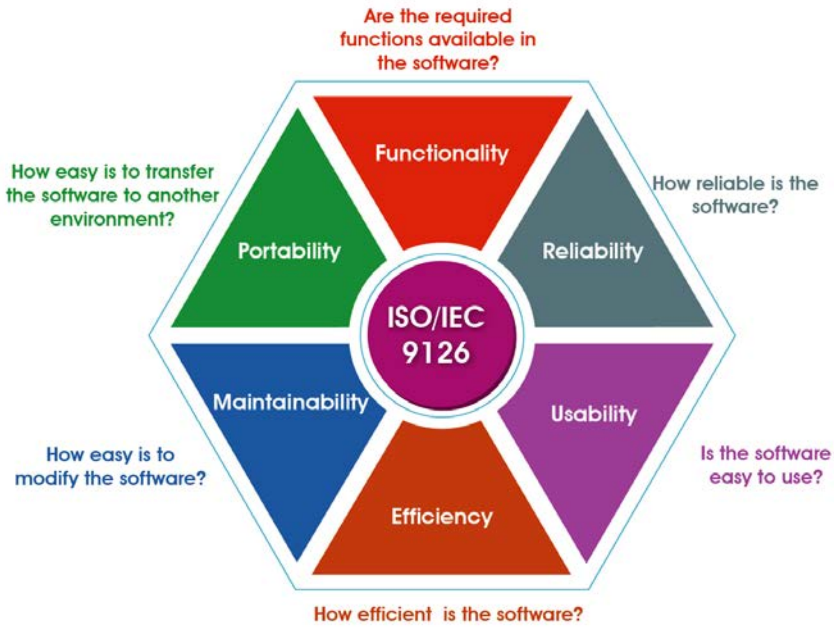

Figure 2.1: ISO 9126 quality factors

the perspectives of quality [59].

Numerous models have been developed in the literature to support software quality. McCall's quality model [58] is the first of the software product quality models. It was followed, in 1978, by Boehm's quality model [41]. Both Models present product quality in a hierarchy with three high level characteristics. Boehm's model has a wider scope with more emphasis on the cost-effectiveness of maintenance [149]. A morerecent work has been conducted to create an international standard for software product quality measurement-ISO 9126 [2]. This standard, presented in Figure 2.2.2, was applied to evaluate numerous prototypes and products in several fields, such as B2B applications [23] and Electronic Books [69]. It is organized in a hierarchy with six characteristics at the top level and 20 sub-characteristics with indicators used to measure the sub-characteristics. In addition to the aspects covered by McCall and Boehm's models, ISO 9126 includes the quality characteristics of functionality [149]. However, ISO 9126 does not clearly state how quality should be measured.

None of these three models presents a rationale behind the selection of characteristics to be included in the quality model and it is not possible to tell if a model presents a complete or consistent definition of quality.
Furthermore, software engineers often believe that software development is negatively affected by measurements without quantitative description. As Pfleeger and Atlee [171] points out, a measure should define where we are and where we would like to go. The problem with all these models is their inability to combine all metrics to provide a global measure that will actually estimate the software quality. Further;ore, they are not user centered. Such features are not easy to evaluate in a subjective manner. Standardized quality models, such as ISO 9126, are only useful as a source of ideas to establish an agreement for a better understanding between customer and developer. The metrics, validated for a correct measurement of each criterion, are not clear. For example, many metrics proposed by McCall *et al.* are obsolete, not validated and simply subjective.

Most of the approaches proposed for modelling software quality are limited in their applicability. They are only useful in the area for which they were designed. Evaluation software is good if it does not add a particular burden to the users [219]. Moreover,, since the quality is improved when the user is involved, user-centered evaluation methods were proposed, which is the subject of the third section of this chapter.

2.3 Evaluation in the field of Human-Computer Interaction

Development models in the field of SE are often directed towards the technical part and not towards the user. Although the users are mentioned for the stages of analysis and validation of prototype, the models and the processes give few explanations relative to the consideration of the users. Besides, the evaluation of the Human-Computer Interaction (HCI) are rarely specified in these processes. So, there are many models that appeared to face this inconvenience.

2.3.1 Development models enriched under the angle of HCI

Under the angle of the development of an interactive system, it is important to take into account the fine aspects of the user. Since the processes of development issued from Software engineering are considered insufficient for the consideration of the user in the design of an interactive system, there was, for more than about twenty years, an enrichment of classic models by trying to integrate the human dimension. We speak about models enriched under the angle of the HCI. As examples, we can cite the Star model [87], the Nabla model [119] [120], the improved V model [18], and the U model [126]. These user centered models show evolutions carried by the HCI domain and used in SE by focusing on essential ideas for the development of interactive systems such as:

- considering the evaluation as the center of the process,
- Fix the activities for the various participants (humans),
- modeling the human activities, the human-machine interfaces and the system,
- further to evaluations, confronting the theoretical tasks planned by the designers with the activities really made by the users.

Nevertheless, these enriched cycles are usable with difficulty because they are not sufficiently complete and show inadequacies such as the iterative development which remains limited (for example in Nabla model). So, these enriched models do not take into account the environments of evolution of the users and technologies supporting these environments characterizing the heterogeneity of the latter.

2.3.2 Quality models in the field of HCI

The development of a wide range of standards related to HCI has been achieved during the last twenty years. As quoted by Bevan [36], the international standards for HCI were developed under the auspices of the ISO

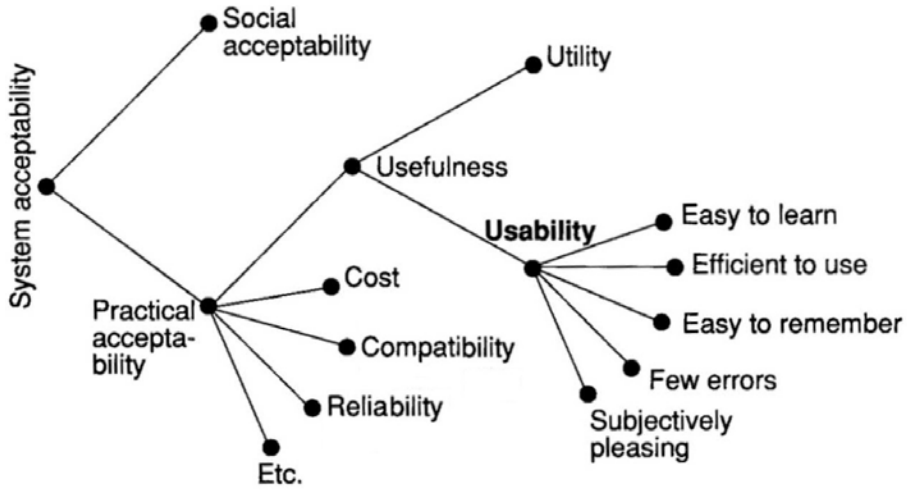

Figure 2.2: Acceptability of systems, as documented in [157]

and the IEC. Most of these standards include general principles from which appropriate user interfaces and procedures can be derived. This assertion yields to upgrading the standards to be more authoritative statements of good professional practice. However, it complicates the determination of whether an interface conforms to the standard or not.
Some researchers have adopted the user viewpoint of quality, recognizing that each person has a different perception of quality. Wong and Jeffery [221], for example, have found that developers and users have different cognitive models for software quality. The evaluation of user interfaces confirms the user's ability to perform his/her task by using the existing communication system.

Through the evaluation of HCI, it would be possible, as mentioned by Nielsen [157], to validate the quality of a system in terms of utility and usability of the systems, as presented in Figure 2.2. Most designers focus on providing the necessary utility, or functionality of the system required for the task, and the social acceptability for users. Nevertheless, the usability, which concerns the quality of the human-machine interaction in terms of ease of learning and use, ensures the adaptation to the user capabilities.

2.3.3 Usability evaluation types

Generally, we distinguish different types of usability evaluation. Indeed, the evaluation of the usability of the user interfaces is often performed to:

- gather information before and/or after the development of a system;
- improve and refine its quality;
- inspect the user judgments or the data that describe the quality of a user interface [174].

In this context, we distinguish four types of assessment that can be built around two categories as follows:

- ***Objective evaluation vs. subjective evaluation:***
  The subjective assessment typically represents the personal judgments of users, expert or not, on the quality of the user interfaces of the system to evaluate [35]. Its focus is on detecting usability problems from a subjective point of view [159], while the objective assessment is usually performed by an evaluation team. Its goal is to achieve results that are independent of personal opinions of users or subjects [35].
- ***Formative evaluation vs. summative evaluation:***
  Formative evaluation (sometimes called internal) is an assessment of the inspection of usability issues that must be resolved during the design phase of the prototype, before the finalization phase [83]. It aims to identify and diagnose problems, then carry out the recommendations and make improvements to the design of the evaluated system. Then, this latter would be assessed again [11] [130].
  As for the summative evaluation, it focuses on the effectiveness evaluation of the final system design. Its purpose is to determine how much a system can meet its objectives [11]. This assessment focuses on the comparison of the level of usability achieved in a design of an interactive system. It can also be used in order to compare alternatives of design in terms of usability [83].

According to Ivory and Hearst [98], the evaluation of usability itself is a process that includes several activities according to the method adopted. In the next section, we will cover the most used and cited evaluation processes in the literature.

2.3.4 From the usability to the quality in use

Abowd *et al.* [3] were the first to propose the categories of quality factors, namely: learnability, interaction flexibility and interaction robustness, which contribute to the usability of a software product. Within those categories, some criteria which are more directly related to the interactive features of a software product are defined. Each category is divided into sub-factors.
Usability is a quality factor that characterizes a software product [105]. What makes a software product usable is the absence of frustration in using it [206].

ISO 9241 [204] describes seven high-level principles for the design of dialogue between human and computer. It defines usability as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use. This definition is close to the one used in HCI field.

To evaluate the quality criteria for the usability factor, these criteria need to be divided into sub-criteria, and then, into usability measures.
As usability problems are harder to specify, evaluate and modularize than certain functionality problems, different usability evaluation techniques have been developed. The existing standards describe the way user-centered design should be practiced. This may have a significant impact on HCI and usability practices [36]. Indeed, considering these recommendations and relating them to usability improvement based on international standards may lead to a new method that includes them in the detailed practices [219].
Bevan [36] states that international standards for usability should be more widely used. According to him, if evaluators have to rely on only one standard, it should be ISO 9241 [204]. This standard provides a high-level framework for usability work.

Table 2.1: Usability criteria as defined by ISO/IEC25010 [95]

Criterion	**Definition**
Effectiveness	Accuracy and completeness with which users achieve specified goals
Efficiency	Resources expended in relation to the accuracy and completeness with which users achieve goals
Satisfaction	Degree to which users' needs are satisfied when a product or system is used in a specified context of use
Freedom from risks	Degree to which a product or system mitigates the potential risk to economic status, human life, health, or the environment
Context coverage	Degree to which a product or system can be used with effectiveness, efficiency, freedom from risk, and satisfaction in both specified contexts of use and in contexts beyond those initially explicitly identified

Later, usability was defined as one of the main product quality attributes for ISO 9126 [2]. It represents the capability of the product, under specific conditions, to be understood, learnt and used [36] [209]. ISO/IEC9126 was renewed into ISO/IEC25010 in 2011, undergoing many changes. Actually, the major ones are that the title of the left side was changed from "internal and external quality" to "system/software product quality" and that all the sub quality characteristics of usability in ISO9241-11 were moved to quality in use. But, it is quite confusing that the sub quality characteristics of usability, i.e. effectiveness, efciency and satisfaction, were all moved to the side of the quality in use even though the usability is still located on the product quality [123].
Moreover, The ISO/IEC 25010 standard, presented in Table 2.1, takes account of both positive and negative outcomes by defining quality in use as a combination of the positive outcomes of usability in the existing ISO 9241-11 combined with freedom from the risk of negative outcomes [37].

So, usability is a qualitative software characteristic associated with most of the requirements concerning the evaluation of user interface. It is defined by quality standards in terms of achieving the quality in use that is perceived by the user during the actual utilization of a product in its *real* context of use [84].
Conceptually, according to Kurosu [123], the quality in use is the relationship that can be described through the quality of the artifact, the user, and the context of use; where the quality of the artifact is the sum of the internal quality and the external quality, and the context includes the environment and the situation.

2.3.5 Evaluation approaches and methods

Several classification methods and evaluation techniques exist in the literature. These methods may be classified into analytical and empirical approaches [196] [102].
The analytical approach involves the usability personnel assessing systems using established theories and methods by estimating several ergonomic criteria. It makes the analyst think deeply about the design and about users, which can yield insights and long-term learning that inform future design decisions [39].
However, the empirical approach is based on the measurement of performance during the experiment in order to test the finished product through a set of data, which is collected during its use by users. The collection is performed through a monitoring or by interviews and questionnaires. The observed data are then analyzed. The analysis usually covers the procedures adopted by users, execution time, the frequency of incidents, etc.

In general, these two approaches complete each other and can be practiced throughout the development process. The iterative practice test-corrections defines the basis of formative evaluation detailed by Hix *et al.* in [87]. According to them, each evaluation provides new lessons. Their integration led to a new version of the product design and/or software.

In the other side, other authors such as Jaspers [101] and Ivory *et al.* [98]

classify the existent methods in five categories:

- ***Expert-based methods:***

  Such methods include guideline review, heuristic evaluation, consistency inspection, usability inspection and walkthroughs [158]. In general, expert-based methods have the aim of uncovering potential usability problems by having evaluators who inspect a user interface with a set of guidelines, heuristics or questions in mind or by performing a step-wise approach, derived from general knowledge about how humans process through tasks.
  With these methods, the measures are easily obtained and can be used to infer problem areas in a system's design. However, they do not give any clear indication why a certain user interface aspect poses a problem to a user or how to improve the interface [101].

- ***Analytic modeling:***

  They are based on predictive models incorporating formal knowledge about the task and grammars or formal models quality. They are envisaged when the user interface is non-existent and/or the user is not available. In this case, the use of abstract representations permits the prediction of the performance which can not be determined with an empirical approach because there has not yet been any experience in using the interactive system.
  GOMS model (Goals, Operators, Methods, and Selection rules) [106] and CTA (Cognitive Task Analysis) [56] are two examples of analytic modeling.

- ***Inquiry:***

  Like the user-based testing methods, these methods require the intervention of the users and are often used during the usability tests. However, the study of the specific tasks or the measurements of performance is not the aim of these methods. Rather, the objective is to collect the preferences or the subjective opinions of the users on diverse aspects of a user interface [88] [194].
  As examples, one can mention the observations [98], the questionnaires [129] [131], and the interviews [98].

Generally, these methods and tools can be used by the evaluators to collect additional evaluation data. They can be used at the beginning of the design process and/or after the realization of the system. These data are useful for the improvement of the user interface to obtain future versions.

- ***Simulation:***

  This category supports in an intrinsic way the automated analysis. It is about programs that simulate the user interacting with the user interface by using models of the user and/or the design of the interface. These programs present the results of this interaction (such as the measures of performance) [98], under various forms: as examples, we can refer to the modeling of the genetic algorithm [114] and the modeling of Petri net [177].

- ***User-based testing methods:***

  They include user performance measurements [157] [83] [195], log file and keystroke analyses [228], cognitive workload assessments, satisfaction questionnaires, interviews and participatory evaluation [139]. Participatory evaluation methods require actual end users to employ a user interface as they work through task scenarios and explain what they are doing, by talking or 'thinking-aloud' [160] or afterward in a retrospective interview. These methods do provide insight into the underlying causes for usability problems encountered by users and participatory evaluation has therefore led to a high level of confidence in the results produced.
  These methods can be applied only if the user interface is ready to be used by the end users (exceptions for interfaces that are in a well advanced phase in their development) [101].
  During the tests, the participants use the system or the prototype to be evaluated to carry out a set of tasks (already determined by the evaluator(s)). A software is used to record the results of the users and generate a set of measures concerning the execution of the achieved tasks, such as the number of errors and the time required for the accomplishment of the tasks.

The common point of all methods for evaluation remains the problem of choosing the most appropriate method(s) for evaluating a system. This choice depends, on the one hand, on the capacity of verification of the evaluation criteria and, on the other on, the existent constraints such as the budget, type of application, time available, etc. Moreover, during the evaluation, the used methods and the collected data are generally numerous and require, sometimes, significant processing time to draw conclusions about the quality of the user interface, especially when we attain a complex system. In this case, the evaluator may be unable to draw conclusions [65].

Usability evaluation methods differ along many dimensions, such as resource requirements, costs, results, and applicability (i.e., at what stages of the interface development process) [98]. There is a wide range of methods that one could employ at all stages of system development, which actually makes the choice of the appropriate method difficult. Usability evaluation methods uncover different types of usability problems; therefore, it is often recommended for evaluators to use multiple assessment methods [157]. For example, during a usability test, participants may also complete questionnaires to provide subjective input, thus, enabling evaluators to gather quantitative and qualitative data.

2.3.6 Evaluation techniques within the user-based testing methods

We present in this section brief descriptions of techniques within the user-based testing methods.

1. **The questionnaire**

   This technique allows obtaining a set of the most subjective judgments [15]. The collected data represent the user's problems in a safe and structured form conducive to the analysis. The questionnaire is a complement to other techniques such as interviews, observations, etc.

   The advantage of using questionnaires is that they provide a fast way of reaching out to many users. Furthermore„ the results obtained from

questionnaires are often quite simple to analyze, and to visualize in tables or diagrams. By simply letting the users rank the quality of that specific feature in the questionnaire, evaluators can easily know what the users think of [184]. So, the questionnaire helps to get a quick overview of what features are good and which ones one needs to be focused on to maintain the quality of the user interface [64].
However, using questionnaires, it can be hard to know exactly what to do with the results. In fact, if the quality of a certain feature was "quite bad", it is not exactly clear what should be done to improve it or what the users think is bad with the feature.
To summarize, the best way to use this technique might be to confirm the results got from using other techniques.

Assila *et al.* have recently established a state of the art in which they have distinguished 24 standardized questionnaires for the usability evaluation of user interfaces [16]. It was found that 71% (17/24) of the questionnaires can be applied to the evaluation of all types of interfaces (such as the WIMP [1], Web and Mobile interfaces). However only seven questionnaires support the evaluation of specific (not for all types of) interfaces. Only one questionnaire deals with the evaluation of **mobile** applications (MPUQ) [183]. This questionnaire includes 72 questions.

Regarding the outputs of the questionnaires, different presentations of the results have been proposed (such as graphic form, number, spreadsheets, CSV files ...). In addition, several ways exist to determine the degree of satisfaction that is sensible to the scales of the used questionnaires. The average computing method is used in general by the famous Likert scale. The latter was adopted by 80% of the questionnaires [16].

Recently, Sauro and Lewis [189] have proposed three types of questionnaires. The first one includes the Post-study questionnaires dedicated to ensure an assessment at the end of a study, especially after complet-

[1] In human–computer interaction, WIMP stands for "Windows, Icons, Menus, Pointer", denoting a style of interaction using these elements of the user interface

ing a series of test scenarios. The second one concerns the post-task surveys that provide a more contextual evaluation. They are used immediately at the end of each task or a scenario in a usability study. The latter category includes the specific questionnaires dedicated to the evaluation of Web applications.

2. **The monitoring**

   In general, the automatic techniques include the automatic capture, the automatic analysis of the captured data, and the automatic criticism [21]. Thanks to these techniques, we reduce the need for experts in evaluation as well as the cost of the evaluation (especially the time).

   The monitoring begun in the 80s with the appearance of the Playback system, which is a simple system allowing the capture of the user's actions by means of a physical device situated between the keyboard and the machine [155]. The central idea of Playback is that, while a user is working with the system, the keyboard activity is timed and recorded by a second computer. This stored log of activity is later played back through the host system for observation and analysis.
   The monitoring, as depicted in Figure 2.3, is an automatic and non-intrusive tool for the record and collection of the user's actions and interactions in real work situations. It can automatically collect objective data to support the evaluation of interfaces. This captured information must be discreetly and transparently done. In fact, the monitoring should not make the user feels embarrassed [68]. This technique may propose analyses of data collected for later treatment. The Figure 2.3 describes the principle of operation of a monitoring. It is composed of three steps [67]:

   (a) The monitoring captures data from the interaction between the user and the system (events). According to Hilbert Redmiles [86], there are six levels of events:

      - The physical events (also called the events of the lowest level, eg, mouse click).
      - The events of input devices (eg, material generated by key interruptions or mouse).

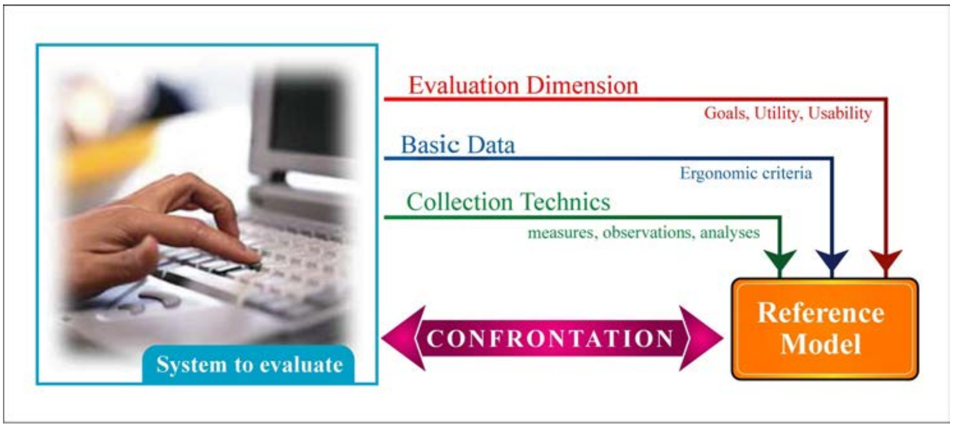

Figure 2.3: The monitoring principle, inspired from [196]

- The User Interface events (such as, changes in the input focus)
- The abstract level of interaction (for example, supply values in the input fields).
- The domain (for example, providing address information).
- The goal / task (for example, place an order)

(b) The captured data is stored in a database and then analyzed to assist the evaluator in achieving his/her activities. Analyses can be achieved after obtaining the result of various calculations or statistics. It can have different forms (text, tables, etc.). Consequently, the evaluator can rebuild models of user's activities. These models are called observed models.

(c) The observed models are compared to the models already specified by the designer (reference model). The result of these comparisons (also called confrontation) may be useful to the designer to improve the interactive system.

The interview techniques and questionnaires can also be used to understand the user‘s activities. However, by capturing events from the user interfaces or interaction devices, the monitoring often allows low-level analyses of these data; for example: search for any sequence of interaction, statistical calculations and visualization of these results. The evaluator would interpret these test results to obtain meaningful conclusions and propose the necessary improvements to the designer. These interpretations differ from one evaluator to another.

Monitoring focuses on the user interface of the interactive system. It does not facilitate the evaluation of many non-functional properties of the system as the response time, reliability, and so on. Nevertheless, these properties are very important to evaluate the functioning of a system. In literature, we distinguish a variety of tools that ensure the evaluation of different types of interactive systems such as MESIA [207], EISEval [210], WebQuilt [92] [216] and Web RemUSINE proposed by Paganelli and Paternò [164]. These tools are able to automatically capture and analyze data. However, they are not able yet to critique and/or suggest interface improvements. Usually, there are two stakeholders in the evaluation process: expert(s) (evaluator(s)), and users.

2.3.7 Evaluation processes

Several evaluation processes have been proposed in the literature. In this subsection, we present firstly some of the well-known and most referenced software product evaluation processes from the standards perspective, and secondly the process of usability evaluation proposed in the context of HCI. Finally, we conclude this section by a discussion.

1) Software product evaluation processes from the perspective of standards

In the literature, various standards (such as ISO 14598-5 (1998) and ISO/IEC 25040 (2011)) have focused on the definition of a general formalism on the standardized evaluation process of software product quality. According to ISO/IEC 25040 (2011), the evaluation process is generally based on the following essential steps:

1. **Establish the evaluation requirements:** The purpose of this step is to establish the evaluation objectives; identify the software product

quality requirements; identify the parts of the product to be included in the assessment; identify stakeholders and define the rigor of the evaluation.

2. **Specify the evaluation:** The purpose of this step is to select quality measures to be adopted depending on the specified requirements of evaluation and the assessment context; establish decision criteria that correspond to the quality measures and then identify the criteria for assessment decisions.

3. **Design the evaluation:** The purpose of this step is to specify an evaluation plan to be described in a specification of the evaluation. Different attributes must be specified such as the objective of the evaluation, environmental assessment, evaluation methods and tools involved, the decision criteria relating to quality measures, etc.

4. **Perform the evaluation:** This step is mainly dependent on the specified evaluation plan. It consists firstly in collecting the values of quality measures and then, applying the decision criteria.

5. **Conclude the evaluation:** In this step, the evaluator(s) has/have to analyze the results of the evaluation and prepare the assessment report that includes a list of criticisms concerning the problems detected to improve software quality.

2) Usability evaluation processes proposed in the context of HCI

In the context of assessing the quality of the user interfaces, there are no specific standards that formalize the process of evaluation. Nevertheless, several assessment processes have been proposed in the literature such as Ivory and Hearst [98] and Mariage [142]. Each process consists of a set of activities that generally include the following ones:

- **The capture:** which is related to the collection of usability data;
- **The analysis:** which is related to the processing and interpretation of usability data to identify the user interface issues;

- **The criticism:** which concerns the suggestion of solutions or improvements to mitigate usability problems.

According to Mariage [142], the evaluation process is generally based on four phases. The capture, analysis and criticism activities are introduced in the second phase *Conduct*. Each phase involves certain goals:

1. **Planning:** The purpose of this phase is to specify:
    - The objectives of the evaluation, taking into account the evaluation stakeholders (designer, evaluator, user, etc.) and the type of the desired results.
    - The context of use of the system to evaluate.
    - The choice of the method(s) to be applied based on the specification of the evaluation targets.
    - The evaluation protocol that includes the points already mentioned followed by the evaluation of scenario (defined, if it exists, by the evaluator).
2. **Conduct:** This phase is based on the objectives of the assessment. It aims to discover the problems with the user interface to evaluate by following the steps of the evaluation scenario. It focuses on the three data collection, analysis and criticism already mentioned above. It is preceeded by a preparation step to develop or configure the tools needed to conduct the evaluation.
3. **Finalization:** The purpose of this phase is to prepare the final assessment report describing the problems detected previously. Subsequently, a communication between the various stakeholders need to be achieved to interpret the results.
4. **Tracking:** Following the communication between stakeholders, different critics concerning the quality of the assessed interfaces can be specified. These can approve the high quality of user interfaces. Otherwise, a new version of the user interface can be realized according to the list of usability problems detected, in order to improve its quality.

Concerning the evaluation of the usability of user interfaces, Hearst and Ivory proposed a process that is based on 12 steps. The capture, analysis and criticism activities are introduced in the steps 8, 9 and 10 [98] detailed in the appendix A.

2.3.8 Usability evaluations of mobile applications

When evaluating the usability of mobile applications, different methods, like heuristic evaluations as rule-based evaluations performed by usability experts and user tests, are applicable. Kjeldskov and Graham [116] surveyed evaluation methods that focus on user-tests with respect to mobile applications. Due to the mobile context that such applications are usually used in, mobile applications can be evaluated in the field. Conducting field evaluations requires a lot of effort. However, the more precise results that can be gained that way do not outweigh higher costs and efforts needed compared to laboratory evaluations [117].

Depending on the type of evaluation (formative or summative) and the current stage of the design phase, results of usability evaluations differ extremely [38]. In laboratory evaluations and evaluation scenarios where data can easily be gathered, quantitative data is collected, analyzed, and related to certain problems [118]. In field studies, often qualitative data and reports from users and evaluators are the only source of information. In long-term studies mobile applications can be used to provide logs for recording interactions of the mobile user to be observed. They can also be applied for remote usability studies of mobile devices [167].

2.3.9 Discussion

In order to evaluate the usability of interactive systems, researchers have to select the aspects of usability to evaluate. At the same time, usability measurement and analysis techniques and methodologies are being developed. Laboratory experiments, field studies, and hands-on measurement (questionnaires) are some of methodologies most often applied by researchers. Every usability evaluation method has its advantages and disadvantages. Some are difficult to apply, and others are dependent on the measurers' opinions or instruments.

Users always tend to choose mobile apps that are easy to learn, take less time to complete a particular task, and appear to be more user-friendly because they are less computer-oriented. However, mobile devices and applications change very quickly, and updated methods of usability evaluation and measurement are required on an ongoing basis. The usability of mobile devices and their applications differ from other computer systems, because their characteristics are different.

2.4 Quality measurement

The area of software metrics has been under research from the early days of software engineering. It describes a way that helps the software engineers to develop high quality of software. As measurement is dealt with, we adopted the term *measure* instead of *metric* in most parts of this report.
In this section, we discuss the basics of software metrics with interesting properties and scales and then focus on the aggregation theory.

2.4.1 Basics of software metrics

Measurement is the mapping from the empirical world to the formal world [112]. In the empirical world, there are entities (things) that have certain attributes that can be expressed with measures from the formal world. Measurement theory is therefore responsible for arguing about the relationship between reality and measures. For example, the table in my office is certainly an entity of reality and has the attribute height. Measurement maps this real attribute of the table to the formal world by stating that the height is 71 cm. Transferred to software, this means that the entity source code has the attribute length which we can measure in lines of code (LOC). This relationship is depicted in Figure 2.4. Also for measurement, there is a well-proven measurement theory that helps us in avoiding mistakes in measuring

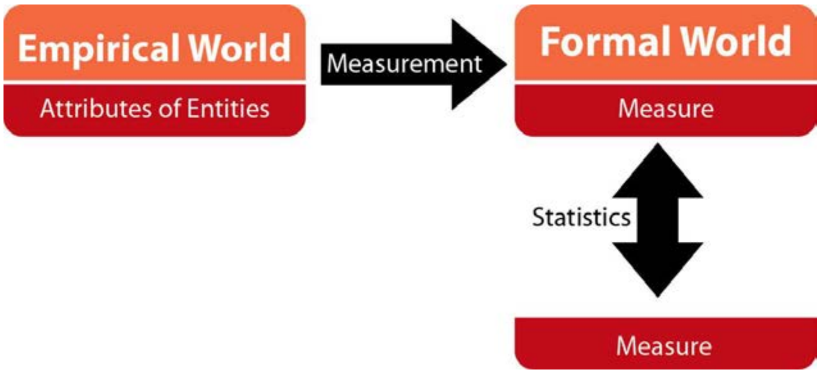

Figure 2.4: The general concepts of measurement and statistics, as presented in [215]

and interpreting measurements. We will, particularly, discuss scales and the important properties of measures. Scales are probably the most important part of measurement theory because they can help in avoiding misinterpretations.

- Scales

  In principle, there are several possible scales for software engineering measures [215]. It usually suffices, however, to understand five basic scales to be able to interpret most measures:

  1. Data that only give names to entities have a nominal scale. Examples are defect types.
  2. If we can put the data in a specific order, it has an ordinal scale. Examples are ratings (*high, medium, low*).
  3. If the interval between the data points in that order is not arbitrary, the scale is interval. An example is temperature in *Celsius.*
  4. If there is a real 0 in the scale, it is a ratio scale. An example are LOC.
  5. If the mapping from the empirical world is unique, i.e. there is no alternative transformation, it is an absolute scale. An example is the *ASCII* characters in a file.

  Therefore, the scales define what is permissible to do with the data

[215]. For example, for measures with a nominal scale[2], we cannot do much statistical analysis but count the numbers of the same values or find the most often occurring value (called *the mode*). For ordinal data[3], we have an order and, therefore, we can find an average value which is the value in the middle (called *the median*). From an interval scale[4], we can calculate a more common average, the mean, summing all the values and dividing them by the number of values. With ratio and absolute scales[5], we can use any mathematical and statistical techniques available.

- Properties of measures

  Apart from the scale of a measure, we should consider further properties to understand the usefulness and trustworthiness of a measure. Very important desired properties of measures are reliability and validity. Reliability means in this context that the measure gives almost the same result every time it is measured. Validity means that its value corresponds correctly to the attribute of the empirical entity. According to Wagner [215], a measure is neither reliable nor valid if it produces a different value every time it is measured.

  In addition, there are further properties of measures which are also desired but not always possible to be achieved. In fact, the reliability of a measurement can be problematic for subjective measures. Therefore, researchers' aim for objectivity in measures meaning that there is no subjective influence in measurement. Next, we want to be able to use the measure in the formal world and compare it to other measures (*comparability*). This requires a suitable scale for the measure (*standardisation*). Moreover, we can measure countless things in a software development project, but the usefulness of those measures should be ensured in the sense that they (those measures) fulfill practical needs.

[2]Nominal scales are used for labeling variables, without any quantitative value. Nominal scales could simply be called *labels*.

[3]With ordinal scales, it is the order of the values, what is important and significant, but the differences between each one is not really known.

[4]Interval scales are numeric scales in which we know not only the order, but also the exact differences between the values.

[5]Ratio scales include weight and height. They give us the ultimate–order, interval values, plus the ability to calculate ratios since a "true zero" can be defined

Finally, *economy* is also a desired property for measures. It is helpful to be able to collect them with low cost. Then, we can collect and analyze often and early. We have observed, however, that in many cases the more useful the measures are, the less economic they are [215].

2.4.2 Aggregation Theory

Aggregation is a topic for not only software measures, but also in any area that needs to combine large data into smaller, more comprehensible or storable chunks. We describe the general theory of *aggregation aggregators.*

There is a large base of literature on aggregation in the area of soft computing where it is also called information fusion. They use aggregation operators in the construction and use of knowledge-based systems. Aggregation functions, which are often called *aggregation operators*, are used to combine several inputs into a single representative value, which can be subsequently used for various purposes, such as ranking alternatives or combining logical rules.

Informally, aggregation is the problem of combining n-tuples of elements belonging to a given set into a single element (often of the same set). In mathematical aggregation, this set can be, for example, the real numbers. Then an aggregation operator **A** is a function that assigns a y to any n-tuple $(\mathrm{x}_1; x_2; ....; x_n)$:

$$A(x_1; x_2; ...; x_n) = y$$

The central tendency, is a known aggregation operator that describes what colloquially is called the average. There are several aggregation operators we can use for determining this average of an input. They depend on the scale type of the measures they are aggregating. All of them are not associative but idempotent.

The mode is the only way for analyzing the central tendency for measures in a nominal scale. Intuitively, it gives the value that occurs most often in the input. Hence, for inputs with more than one maximum, the mode is not uniquely defined. If the result is then defined by the sequence of inputs, the mode is not symmetrical. The mode is useful for assessing the current state of a system and for comparisons of measures in a nominal scale. For the frequencies of the input values $(\mathrm{n}_1; ...; n_k),\ the\ mode\ (M_m)\ is\ defined\ as:$

$$M_m(x_1; ...; x_k) = x_j$$

$$\prec = \succ$$

$$n_j = max(n_1; ....; n_k)$$

The median is the central tendency for measures in an ordinal scale. An ordinal scale allows to enforce an order on the values and hence a value that is in the middle can be found. The median ensures that at most 50% of the values are smaller and at most 50% are greater or equal. The median is useful for assessing the current state and comparisons.

Only few contributions to the theory of aggregation operators in software measurement have been made. The main basis we can build on is the assignment of specific aggregation operators (especially for the central tendency) to scale types. The scales are classified into nominal, ordinal, interval, ratio and absolute. This classification provides a first justification for which aggregation operators can be used for which classes of scales. For example, consider the measures of central tendency such as median or mean. To calculate the mean value of a nominal measure does not make any sense. For instance, what is the mean of the names of the authors of modules in a software system? This is only part of the possible statistics that we can use as aggregation operators.

2.4.3 The measurement reference model: ISO/IEC 25020

Quality measure elements are described as an input for the measurement of the software quality measures of external quality, internal quality and quality

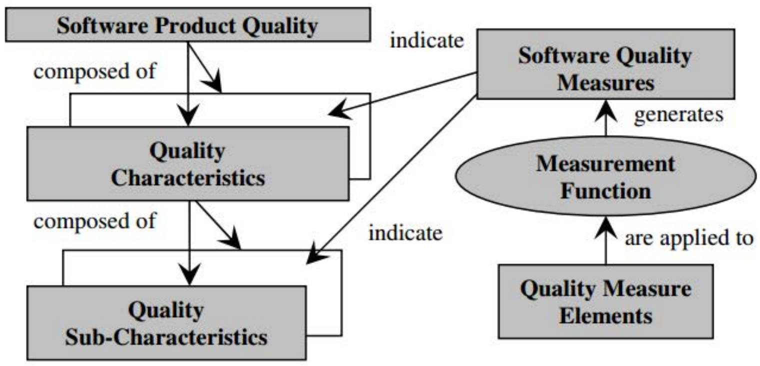

Figure 2.5: Quality Measure Elements Concept in the Software Product Quality Measurement, as defined in ISO/IEC FDIS 25020 [96]

in use [96]. Figure 2.5 shows the relationship between the quality measure elements and the software quality measures, and between the software quality measures and the quality characteristics and subcharacteristics (criteria). In metrology, these would correspond to base measures and derived measures, respectively. It can be observed that these measures, in particular the derived measures, are defined specifically to measure the sub-characteristics of internal and external quality or the characteristics of quality in use. None of these is directly related to the top level of software quality (which is itself broken down into three models, then into a set of characteristics and further into a large number of sub-characteristics). The evaluation methods and tools available are closely linked to the measurement of usability in order to detect specific problems of the user interfaces. Usability cannot be measured directly; however, its concept was surrounded by different criteria that can be measured [157], [105]. The definition of measures related to usability criteria has been widely exploited in the literature such as standards [96] and research works [130];[195]; [11], [227]. Therefore, many and various measures have been proposed.
For example, Seffah *et al.* (2006) have developed a usability synthesis model (called Quality in Use Integrated Measurement [QUIM] [195]) based on existing works that include conventional models (such as those of ISO 9126 [2] and ISO 9241-11 [204]) and conceptual models (such as Metrics for Usability Standards in Computing [MUSIC] ([36]).

Recently, many other measures on various quality criteria have been proposed by SQUARE standards (such as ISO/IEC 25022 published in 2012). As examples, we cite in Table 2.2 some measures defined by this standard on some criteria and sub-criteria quality.

2.4.4 Single Usability Metric

This method of usability evaluation of interactive systems is proposed by Sauro and Kindlund [188]. It aims to ensure a summative evaluation of software product and compare different tasks, studies or products. In addition, this method simplifies the different obtained measures to one summative standardized measure called SUM (Single Usability Metric).
This is based on the aggregation of different evaluation measures which correspond to the criteria of usability; measures are objective and subjective. These measures aretask time, the number of errors, the completion of the task and the average user satisfaction. These measures are further combined by using methods of normalizations. The aggregation concerns the calculation of the average of the measures normalized during every task.

Two major advantages support the use of this method. At first, this score supplies a continuous variable that can be exploited in the analysis of regression, the tests of hypotheses. In the same way, the existing measures can be used to inspect the usability. Secondly, a single measure based on the limits of logical specification gives an idea of the way with which a task or a product can be usable without having to reference historical data.
However, the use of this score depends on the data used in the study and cannot be compared with other scores resulting from other data sets. Besides, the aggregation of the subjective and objective measures in a unique score cannot be beneficial for the detection of usability problems [188]. In addition, another limit can be raised with regard to the equation of aggregation of SUM. This concerns the way of choosing the target level of each measure, so the interpretations during the declaration of the results require long studies to draw conclusions [188].

Table 2.2: Some examples of measures defined by ISO/IEC 25022

Criterion		Measure
Effectiveness		$Task\ effectiveness = 1 - \sum proportional\ value\ of\ each\ component\ is\ missing\ or\ incorrect\ for\ a\ task$
		$Error\ frequency = \frac{Number\ of\ errors\ made\ by\ the\ user}{number\ of\ tasks}$
Efficiency		$Execution\ Task\ time$
Flexiblity		$Flexible\ context\ of\ use = \frac{Number\ of\ additional\ contexts\ where\ the\ product\ is\ used}{Total\ number\ of\ additional\ contexts\ in\ which\ the\ product\ may\ be\ used}$
Satisfaction	Utility	$Satisfaction\ scale = \frac{Questionnaire\ producing\ psychometric\ scales}{Average\ population}$
		$Satisfaction\ Questionnaire = \frac{\sum(Response\ to\ question\ i)}{number\ of\ responses}$
	Pleasure	$Pleasure\ scale = \frac{Questionnaire\ producing\ psychometric\ scales}{Average\ population}$
	Comfort	$Comfort\ scale = \frac{Questionnaire\ producing\ psychometric\ scales}{Average\ population}$

2.4.5 Summary

We gave in this section a comprehensive but certainly not complete overview of software measures and measurement as well as a collection of useful aggregation operators for use in a variety of measurement systems. Moreover, we presented the measurement reference model(ISO/IEC 25020) and some measures defined by this standard on some usability criteria and sub-criteria. We also presented further properties as well as the limits of Single Usability Metric that simplifies the different obtained measures to one summative standardized measure. Such method can be considered as important for the measurement of the usability of interactive systems.

2.5 Evaluation of decision support systems

Several works have provided static decision support. One of the most important issues in Decision Support Systems (DSS) technology is in assessing their quality for future implementations and use [97]. Here, we can assume that their evaluation with respect to criteria was not progressing [213]. In fact, the evaluation of a DSS is, generally, empirical [213][198][55] and had usually usability, effectiveness, and cost effectiveness criteria [213][198].

Recently, authors have included other criteria in their DSS evaluation proposals. For example, Nadepur *et al.* [154] proposed six independent subjective sub-scales: *Mental, Physical, Temporal Demands, Frustration, Effort, and Performance.* Users (i.e. decision-makers) were asked to rate the perceived workload on a continuous scale with three anchors (low, medium, and high).
However, other researches have focused on objective evaluation criteria. For example, De Wit *et al.* [60] have defined their own evaluation criterion which is the *efficiency.* It was defined as the total number of relevant alerts divided by the total number of alerts provided by the DSS.

In a mobile environment, the decision being made follows a dynamic process. We cannot assume that the score we give for a particular alternative for a particular criterion will remain constant over time. Indeed, there is a need for not only providing decision support in the traditional sense as seen with static decisions, but also taking into consideration a possibility of some changes in data over time and giving some indication of the quality of the decision.

According to Pérez [169], MDSS should reduce the time constraint in the decision process. Thus, the time saved by using the MDSS can be used to perform an exhaustive analysis of the problem and obtain a better problem definition. This time could also be used to identify more feasible alternative solutions to the problem, and thus, the evaluation of a large set of alternatives would increase the possibility of finding a better solution. The MDSS should also help us in the resolution of problems providing a propitious environment for the communication, increasing the satisfaction of the user and, in this way, improving the final decisions.

From the other side, Padmanabhan *et al.* [163] have stated the issues that have impeded the growth and implementation of MDSS, such as user privacy, systems interoperability and integration, lack of worldwide standards, technological limitations and lack of standard rigorous evaluation frameworks.

MDSS enable users to achieve a large variety of tasks in several situations. Consequently, these systems are typically used in a highly dynamic environment. There is still a need to understand the factors impacting not only the acceptance of these systems by the decision makers, but also their efficient usage. To this end, we are interested in MDSS evaluation. The purpose is to evaluate the success of a MDSS by suggesting an approach that takes into account the continuous changes of the context of use.

2.6 Conclusion

Usability evaluation of interactive systems occupies a great interest for Human-Computer Interaction community. Indeed, several subjective and objective evaluation methods and tools have been proposed and applied in academy and industry. In parallel, an impressive development of quality models has taken place over the last decades. These efforts have resulted in many achievements in research and practice. The developments in quality definition models led to the standardization in ISO/IEC 25010, which defines well known quality factors and serve as the basis for many quality management approaches. It even integrates a quality evaluation process based on ISO/IEC 25040 standard.

In this chapter, we presented a state of the art about the evaluation in development processes in software engineering as well as in the field of Human-Computer Interaction. In doing so, we presented a state of the art on the usability evaluation techniques within the user-based testing methods that ensure subjective or objective evaluations. We also achieved a brief study about the usability evaluations of mobile applications. Moreover, we presented the basics of software metrics with interesting properties and scales and then focused on aggregation because that is very important for quality evaluation.
Finally, as we are interested in decision support systems, which are considered as interactive systems, we have introduced the most known works in the field of the evaluation of mobile decision support systems.

From this synthesis, we can deduce the following:

- There are some well established approaches to evaluate the performance of a software product and, more recently, there has been an increasing focus on user-oriented evaluation criteria and methods for evaluating systems within the context of human-computer interaction.
- Metrics are a common means to quantify quality aspects of software. To gain reasonable statements from metrics, a quality model which defines distinct characteristics and corresponds to sub-characteristics that relate to software quality is required.
- Quality models proposed in the literature are limited to only one or two fields among HCI, software engineering, Data mining, etc.
- MDSS are highly interactive and need to be evaluated as the other

interactive software products.

In this context, we will propose in the next chapter a novel approach of MDSS evaluation. This approach will consider only MDSSs that are based on the KDD process (MDSS/KDD presented in Chapter 1). It aims mainly at helping evaluators to detect defects as early as possible in order to enhance the quality of all the modules that constitute a MDSS/KDD. The proposed evaluation support modules evaluate not only the quality in the use of each module composing the KDD process, but also other criteria that reflect the objective(s) of each KDD module.

3. CONTRIBUTIONS TO THE EVALUATION OF MDSS BASED ON THE KDD PROCESS

3.1 Introduction

Few progresses have been realized to improve Decision Support Systems (DSS) evaluation methods within the Knowledge Discovery from Data process (KDD). Moreover, little effort has been deployed on the measurement aspects towards the assessment of the quality of such systems.

In the previous chapter, a state of the art on the methods and the models for evaluating interactive systems, as classic as enriched under the angle of the HCI, allowed to accentuate the absence of an approach of precise and appropriate evaluation of Mobile DSS based on the KDD process (MDSS/KDD).

We remind that we are concerned with the evaluation of MDSS/KDD process that can be defined as KDD-based systems that can support users in making decision, through their mobile devices.

In this chapter, we begin with an emphasis on the motivations behind this work. We will present the essential characteristics of such systems that made us think about a novel approach. Afterwards, we propose our first contribution for the evaluation in the KDD process, which can be appropriate to enhance the evaluation task within the process on which some developers are based to develop DSS. Then, as second contribution, we will conduct our second contribution dealing with MDSS/KDD. This latter considers the mobility aspect that is strongly related to the context of use. Finally, as third contribution, we present an evaluation support system that can help evaluator(s) in measuring the quality factors introduced in our first contribution, while considering the mobile aspect underlined in our second contribution.

3.2 Raising the issue of MDSS/KDD evaluation

MDSS is very beneficial when decisions are complex, critical and made under time pressure, as well as when decision-makers are on the move in dynamic environment. As explained in the first chapter (see section 1.4), KDD, which is the most used process for DSS, is composed of several stages. These stages are all susceptible to be exploited in mobile contexts.

Studies in software engineering have found that more than 30% of the faults were detected after release [185]. So, it would be necessary to improve the MDSS quality even after its implementation. Mainly, the evaluation should interest all the process on which developers are based to develop their DSS, because KDD modules are linked and joined [25]. KDD process adopts a centralized evaluation module localized after the Data Mining which is the focal module that generates patterns from large data bases. This evaluation module is provided to verify whether the patterns generated from the DM module are interesting.

Although the last years have seen an interest within the research community in the evaluation of interactive systems, MDSS has not been of a strong interest. Few researchers have underlined this gap and defined the criteria that must be measured and optimized to obtain a better quality of DSS. Nevertheless, MDSS has always been seen as either DSS or interactive systems. Besides, although previous works pertaining to the KDD process have clearly shown that each module in KDD should be designed, implemented and assessed [32], evaluation as proposed in the literature is, from our point of view, incomplete as it neglects several quality factors such as quality in use, quality of data, etc. These works remain limited to the DSSs which are not mobile and still concentrate only on the evaluation of the data mining stage.

Moreover, as for DSS, the evaluation of MDSS needs the support of the user to ensure a better quality. However, this seems to be insufficient especially in a continuously changing environment. In case of mobility, there are additional information that should also be taken into account, particularly that describing the context of use.

Figure 3.1 summarizes the points revealed in this section and determines the shortcoming of MDSS evaluation field. In this work, we will propose a quality model for each stage among KDD process, further each stage would be evaluated through an evaluation module. Indeed, each evaluation module concerns one stage from the KDD process and allows measurement criteria composing the quality model. Those measures are considered as a key for maintaining the MDSS.
The choice of the criteria used for the evaluation of each module will also be discussed. Moreover, a global evaluation of KDD modules is well needed to verify the acceptability of the MDSS offered to the user.

In the next section, we will detail our proposition for evaluating all KDD

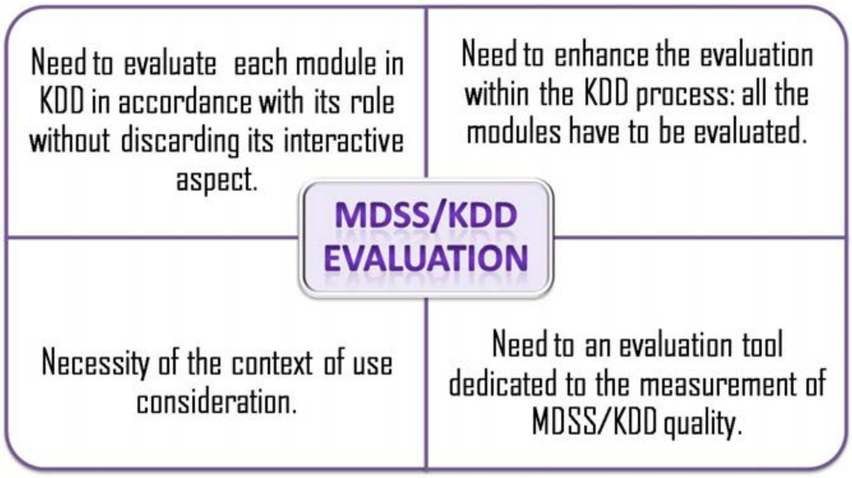

Figure 3.1: MDSS evaluation shortcomings

stages.

3.3 Enhanced evaluation in the KDD process

KDD modules are closely related since they compose the KDD process. Each module has its own objectives and specifications. As explained in the first chapter (see section 1.4), the KDD process follows these steps: (1) Data collection, selection, preprocessing and transformation, (2) Interesting patterns extraction with data mining algorithms. A post-processing activity (3) is important to interpret the patterns provided to get knowledge units. These units are to be modeled, stored and shared to help the decision-maker(s) get the best action(s) to perform.

Nevertheless, as presented in Figure 3.1, so much criticism can be revealed regarding the evaluation within the KDD process. In fact, among the most recent research works dealing with our subject, we based our approach based on the work of Ltifi et al. described in chapter 1 (section 1.4.3). Although we agree with [136] arguing hat evaluation is a main objective that should be achieved in all the steps of the KDD process, we consider that the evaluation of Decision support systems based on KDD process is an intention that should be treated by taking into account two main elements:

(1) The specifications of each step constituting the KDD process,
(2) The quality in the use of each step.

Mainly, our contribution consists in adding an evaluation support module in each module composing the KDD process. Thus, we can establish an evaluation support system that helps the evaluator(s) to have a global idea about the quality of the evaluated system. As shown in Figure 3.2, evaluation is no longer a centralized module in KDD process that was previously (in several research works) concerned only with the evaluation of extracted patterns [71] [79] [17]. Evaluation was integrated with the Data mining module. Therefore, the output of this module would be originally interesting. In our opinion, the data acquisition and storage module (the first module in the KDD process), as well as the knowledge management (the last module in the KDD process) module, should also integrate an evaluation support module.

Although the different modules of KDD seem to be similar in their form, as they are considered as software offered to end users, the role and the aim of each module put together the huge difference between those modules. Our objective is to support the evaluator(s) in the assessment of the quality of each module in KDD process by measuring a set of quality factors. Based on our proposal, the role of the evaluation in KDD is enhanced; i.e., the evaluation concerns all KDD modules. As a consequence, a global evaluation of a decision support system based on KDD process must be ensured.

Therefore, the evaluation support system of a DSS based on KDD process requires the development of, at least, three successive modules that are all evaluated as shown in Figure 3.2. The evaluation support modules integrated in the KDD process will not be the same in each module.

In the remaining part of this section, we present the quality factors that would be taken into account for the evaluation of a DSS based on KDD process. These factors would be further divided into a set of criteria As we are dealing with KDD which is a highly interactive process [136] [89] [132], we adopt the *quality in use* factor, defined by ISO 25010 [95] and recalled in section 2.3.4 (see table 2.1), in the evaluation of all the KDD

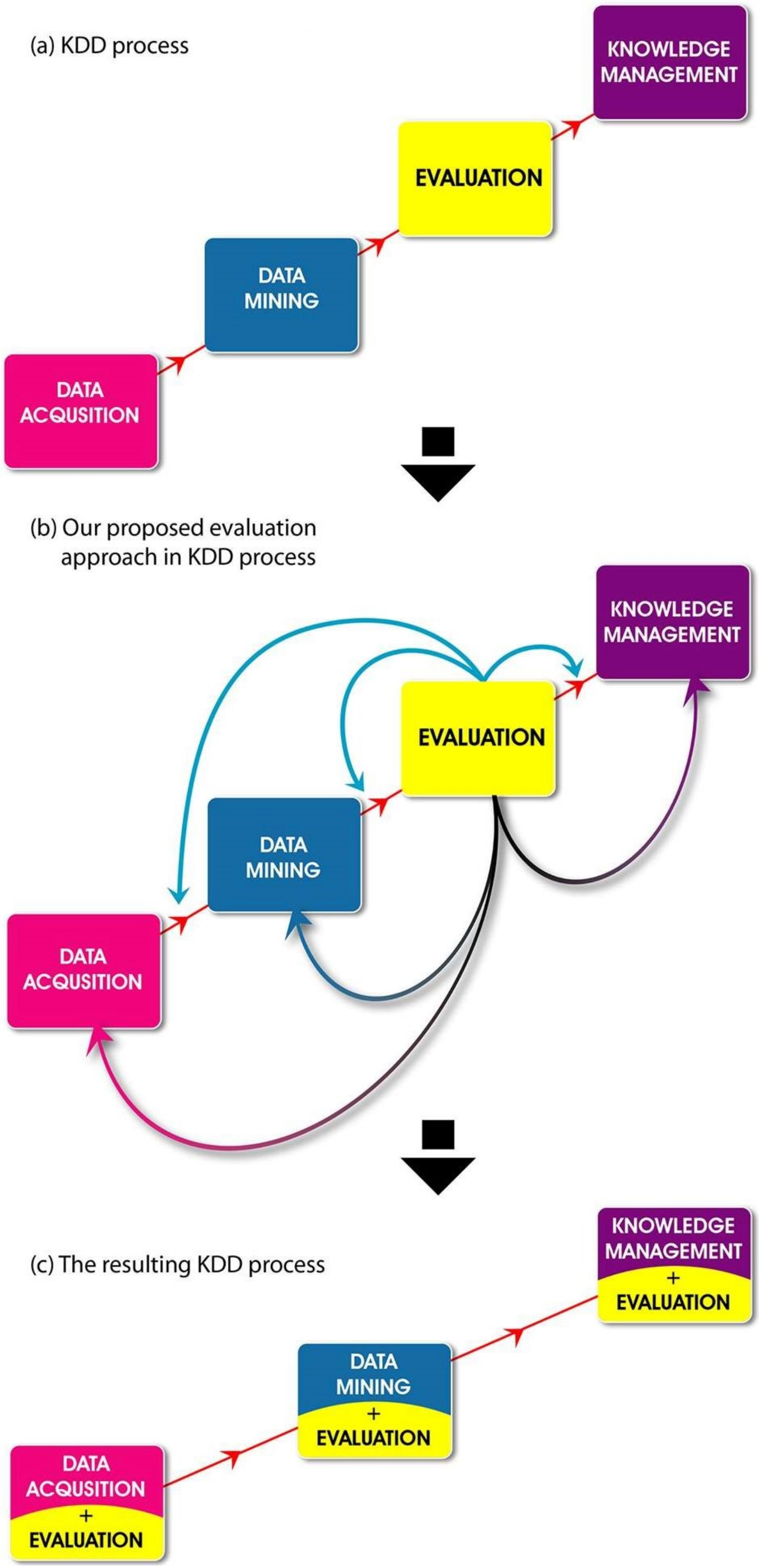

Figure 3.2: Enhanced evalution in KDD process

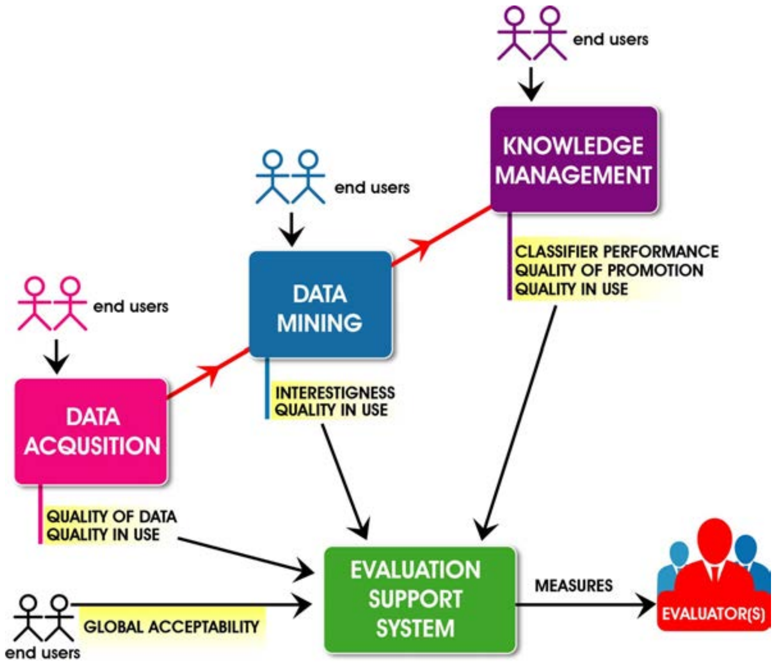

Figure 3.3: The proposed quality factors for the evaluation of KDD modules

modules because the end user (i.e the decision maker) will interact with at least one module. The other proposed quality factors (*Quality of Data, Interestingness, Classifier performance, and Quality of promotion*), depicted in Figure 3.3, are detailed in the next subsections. Thus three evaluation support modules are needed to achieve the measurement of these quality factors. Moreover, we propose to add the *Global acceptability* quality factor which allows a global evaluation of the whole DSS based on KDD process including its acceptability by the end user.

In doing so, we propose an Evaluation Support System (ESS) composed of four evaluation support modules. Indeed, three modules deal with the KDD in addition to a fourth one dealing with the global evaluation of the system. The proposed ESS is presented in section 3.4.

The rest of this section contains four parts in which the previously-mentioned evaluation modules are detailed. The first part is a description of the evaluation support module that concerns *Data acquisition and storage* which is the first module in KDD process. The second one presents the evaluation support module that concerns the *Data Mining* module. *Knowledge management* Concerning the evaluation support module, it is described in the third part of this section. As for the global evaluation support module, it is presented in the fourth part of this section.

3.3.1 Evaluation support of the Data acquisition and storage module

To the best of our knowledge, the potential of extracting the interesting criteria for the evaluation of data acquisition and storage module is currently inexistent in the literature. Since data given by such application and presented to the end user can affect the quality of patterns generated by the next module (Data Mining), we make use of the benefits of criteria extracted from ISO 25000 [95] to improve the quality of this module. In fact, we consider that this quality depends on two factors which are:

- The *quality of data* emanating from different sources and having many types of formats. These data need to be stored in the data base. In order to have a better quality of data attempting the next module of KDD (Data Mining), an evaluation module verifying the quality of data is subjoined.
- The *quality in use* of the application that allows end users entering, modifying or deleting data. It represents software quality from the user's point of view. Several user profiles would be involved to ensure a better quality in use evaluation.

Table 3.1 and Table 2.1[1] depict criteria published in ISO 25000 [95] related to both quality factors. As a result of this evaluation module, there will be a specification of not only the defects causing the impairment of this module, but also the cause of these defects, if ever they exist. Once problems are specified, evaluators should take an interest in and be responsible for the consideration of the found defects (leading to a maintenance of the

[1] see section 2.3.4 in chapter 2

Table 3.1: Quality of data criteria, as defined by the ISO 25012 standard [1], and the proposed technique of measurement

Criterion	Definition (from ISO 25012)	Proposed technique
Accuracy	Degree to which data have attributes that correctly represent that true value of the intended attribute of a concept or event in a specific context of use.	Questionnaire (Q.Data)
Consistency	Degree to which data have attributes that are free from contradiction and are coherent with other data in a specific context of use.	
Credibility	Degree to which data have attributes that are regarded as true and believable by users in a specific context of use.	
Currentness	Degree to which data have attributes that are of the right age in a specific context of use.	
Precision	Degree to which data have attributes that are exact or that provide discrimination in a specific context of use.	
Traceability	Degree to which data have attributes that provide an audit trail of access to the data and of any changes made to the data in a specific context of use.	
Portability	Degree to which data have attributes that enable them to be installed, replaced or moved from one system to another, while preserving the existing quality in a specific context of use.	
Recoverability	Degree to which data have attributes that enable them to maintain and preserve a specified level of operations and quality, even in the event of failure, in a specific context of use.	

considered modules), either in the quality of data or in the quality of the user interfaces.

As shown in Table 3.1 and Table 2.1, two techniques of measurement are used. The first one is the questionnaires which allow obtaining a set of the most subjective judgments. The collected data represent the user's problems in a structured form conducive to the analysis. As for the second technique, it is the ESS presented in the last part of this chapter (see section 3.5).

3.3.2 Evaluation support of the Data Mining module

Traditionally, evaluation in KDD is considered as a possible interpretation of the mined patterns to determine which patterns can be considered as new knowledge [71] [79] [17] [82]. Interpretations make out the concepts in the form of rules concluded by the experts after interpreting the patterns provided by the KDD process [71].

In the present research work, we propose to differentiate the interpretation from the evaluation. Indeed, it is considered that the evaluation of DM module should allow measuring a set of criteria for assessing its quality based on the following factors, detailed in Table 3.2:

- The *interestingness* of patterns: it includes not only the objective measures (O) based on probability but also the subjective criteria (S) based on the user's point of view. Measuring the interestingness factor is intended for selecting and ranking patterns according to their potential interest for the user. Thus, the proposed quality model for the evaluation of the Data Mining module would not discard traditional and substantial criteria (*Support*, *Confidence*, *Lift*, and *Kulczynski*). The utility of these criteria among several others was discussed in [82].
  In addition, the user's needs are considered when we evaluate this module. In fact, subjective measures (*Novelty*, *comprehensibility* and *surprisingness*) are added to our quality model as they take into account both the pattern and its user [24]. This kind of measures needs access to the user's domain or background knowledge about the data

by interacting with the user during the data mining process or by explicitly representing the user's knowledge or expectations [79]. Novel and surprising knowledge is potentially useful to lead to some benefits to the user or task. Besides, we consider that it is preferable not to present incomprehensible information (a rule for example) to the user, so that the patterns should be understandable by the final user.

- The *quality in use*: The data mining process should be highly interactive. Thus, it is important to build high quality user interfaces that facilitate the user's interaction with the system. Moreover, this focal module in KDD should allow the user to dynamically change the focus of his/her search, to refine mining requests based on returned results. The presentation and visualization of data mining results can be considered as solution to present these results to the final user [91] [137]. Our task, at this level, is to verify if a data mining system presents data mining results flexibly, and if the discovered knowledge is easily understood and directly usable by humans (i.e. decision makers). It would be required to adopt expressive knowledge representations, and visualization techniques.

Data mining can be seen not only as an algorithmic step into KDD, but also as a software generating useful information for the decision maker. Thus, the criteria that assess *quality in use*, introduced in the previous section, should also be assessed. By evaluating *interestingness* and *quality in use* of the Data Mining module, the knowledge units can be provided with better quality.

3.3.3 Evaluation support of the Knowledge Management module

Before describing our proposal to evaluate a Knowledge Management System (KMS), we note that we consider only the explicit knowledge types that are (or can be) articulated, codified, stored in documents, and can also be re-used [20]. Our focus is on the performance of a KMS used for prediction issues by inter-operating with the existing DM module. Our evaluation as-

Table 3.2: Criteria used for the evaluation of the Data mining module

Quality factor	Type	Criteria	Definition	Measurement	Source
Interestingness	O	Support	The number of records in the data-set for which the consequent and the antecedent evaluate to true.	$P(A \cup B)$(3.1)	[7]
		Confidence	The probability with which the consequent evaluates to be true given that the antecedent evaluates to be true in the input data-set.	$P(B/A)$(3.2)	[7]
		Lift	The degree to which the occurrence of the antecedent "lifts" the occurrence of the consequent.	$\frac{P(A \cup B)}{P(A).P(B)}$(3.3)	[217]
		Kulczynski	The average of two conditional probabilities: the probability of B given A, and the probability of A given B.	$\frac{P(B/A)+P(B/A)}{2}$(3.4)	[222]
	S	Novelty	Degree of which the pattern is novel by the final user.		[10]
		Surprisingness	Degree of which the pattern is surprising by the final user.		[79]
		Comprehensibilty	Degree of which the pattern is understood by the final user.	Questionnaire (Q.Interest)	[49]
Quality in use	O and S	Same criteria presented in Table 2.1			[95]

- O refers to Objective criteria
- S refers to Subjective criteria

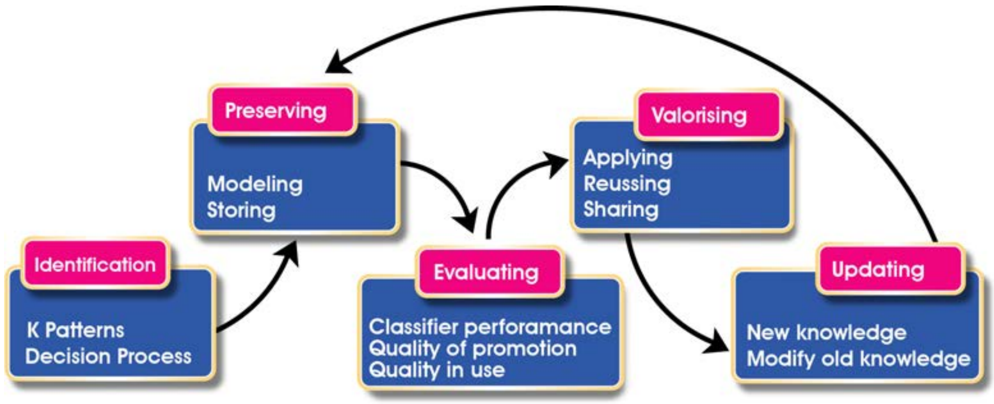

Figure 3.4: Evaluation in the KM process

pect is integrated in the knowledge life cycle from creation to its application in one side, and its promotion in the other side. In fact, managing a knowledge base is a process itself. We have integrated a central evaluation module in this process, as it is shown in Figure 3.4. This module guarantees the quality of a KMS and ensures the assessment of three quality factors which are:

- the *classifier performance* : it includes six criteria, described in Table 3.3, which are usually used after discovering knowledge in order to evaluate the performance of a KMS. The measures of these criteria depend on the value of the following items, in case we dispose of two classes of prediction (Class A and Class B):

  α: the number of instances, really in A and predicted in A.
  β: the number of instances, really in A and predicted in B.
  δ: the number of instances, really in B and predicted in A.
  η: the number of instances, really in B and predicted in B.

- the *quality of promotion*: it includes four criteria, also described in Table 3.3. The proposed criteria allows an evaluation of the interchangeability of knowledge, In fact, in knowledge sharing culture [33] [124], to share and collaborate, people work together more effectively to make organizational knowledge more productive

- the *quality in use*: As for the first and second modules in KDD, we include this quality factor not to overlook the point of view of the end user, who is highly concerned with this module as it helps him/her in making decision(s). Moreover, the goal of such evaluation is not only to guarantee a better quality inside the KMS, but also when using it.

3.3.4 Global evaluation

The purpose of the final stage of evaluation is to permit an evaluation of the totality of KDD modules in order to foster better communications between decision makers and the staff of any organization. Practically, this evaluation module aims at assessing the whole system that implements the KDD process which forms together a decision support system based on KDD process. Therefore, it enables users (i.e., decision makers) to attach appropriate comments to the resulting metrics. In fact, it would be fruitful to assess the interoperability between the KDD modules in addition to the social acceptation of the system such as its cost, priority, and trust. The criteria used here are defined in Table 3.4.

Thanks to this evaluation, based on the criteria mentioned in Table 3.4, in addition to the criteria defined for the different KDD modules, the evaluator(s) can determine if the DSS/KDD is promising (so it is a good investment) or it needs more time to reach higher quality.

3.3.5 Discussion

It is challenging for the evaluator to face the assessment of a complex DSS, implemented with reference to the KDD process, and to discover real gaps that intercept the existence of high quality of DSS/KDD process. Furthermore, such evaluation requires reliable sub-evaluations of the different systems that compose the KDD modules. Moreover, a global evaluation of the whole DSS/KDD process allows the evaluator to have a general idea about the quality of such complex system. Our proposition was to integrate an evaluation module in each module of the KDD process in order to support the evaluator(s) to have a global idea about the quality of the whole decision

Table 3.3: Criteria used for the evaluation support of the knowledge management module

Quality factors	**Criteria**	**Definition**	**Measurement**
Classifier performance	The Positive Prediction Rate (PPR)	$(\alpha + \eta)/card(M)$ (3.5)	ESS
	The Global Error Rate (GER)	$1 - PPR$ (3.6)	
	The Positive Prediction Ability Rate (PPAR)	$\alpha/(\alpha + \delta)$ (3.7)	
	The Negative Prediction Ability Rate (NPAR)	$\eta/(\beta + \eta)$ (3.8)	
	The Sensibility (S)	$\alpha/(\alpha + \beta)$ (3.9)	
	The Specificity (SP)	$\eta/(\delta + \eta)$ (3.10)	
Quality of promotion	Awareness	Awareness of the knowledge available. This criterion aims at verifying if the provider makes use of directories, maps, etc.	Questionnaire (Q.Promo)
	Access	Verifying if the final user has an access to the knowledge.	
	Guidance	Knowledge managers are often considered key in the build-up of a knowledge sharing system. They must assist users, and be responsible for the language used in publications and other communication material. This is so as to avoid a knowledge overload.	
	Completeness	Verifying if the expert (i.e final user) has an access to both centrally managed (by the DME) and self-published (by the expert) knowledge.	
Quality in use	Same criteria presented in Table 2.1		

ESS represents our proposed Evaluation Support System presented in section 3.5.

Table 3.4: Criteria used for the global evaluation

Criteria	**Definition**	**Source**	**Measurement**
Interoperability	The amount of cooperative work among the applications representing the KDD modules.	[121]	Questionnaire (Q.G)
Cost	This is how much the final user/organization is satisfied by the cost of the system.	[113]	
Response timeliness	This is how fast a service request can be completed. It is measured by response time and execution time.	[8]	
Intention to use	This is how likely are the users intend to use the system in the future.	[99]	
Priority	This is how much it is necessary and important to have such systems for the user/organization.	[5]	
Trust	This represents the level of confidence final users have regarding the whole system. This measure involves privacy, communication security (interception of secret information, and abuse of personal information).	[5]	

support system based on KDD process. This solution is partly automatic. In fact, while several evaluation criteria are measured using our ESS, the other criteria are measured through questionnaires to get directly the point of view of the final user.
The enhanced evaluation in the KDD process allows a significant support for the evaluator face to a complex DSS/KDD. Actually, unlike most of the previous studies that focused only on the evaluation of the patterns generated by the data mining algorithm, the present study offers a quality model for all the modules composing the KDD. Therefore, we propose four quality models:

- The first quality model based on ISO/IEC 25000 for the evaluation of the first module of KDD *Data acquisition and storage*. This model includes the quality of data inserted in the databases, which would be further used to extract useful knowledge, and the quality in use, which is strongly related to the quality of the user interfaces offered to the final user to achieve his/her objective(s).

- The second quality model allows an evaluation of the second module of KDD (*Data Mining*). It includes the interestingness factor, which is usually used by all researchers in the field of data mining, to extract only interesting patterns. In addition, we have considered the quality-in-use of the system, because, according to us, the interface that generates the data mining results should also be highly interactive and usable.

- Like the two previous modules, the third quality model allows an evaluation of the quality-in-use. In addition, as we are interested in the KMS used in the case of prediction, an evaluation of the classifier performance was conducted. Moreover, we have proposed to evaluate the quality of promotion of the knowledge inside the organization. This quality factor evaluates if the knowledge can be successfully shared between decision makers.

- The fourth quality model allows the evaluation of the whole DSS/KDD process. It gives the evaluator a general idea about the quality of the system as it is related to all the evaluation modules. Moreover, this evaluation module ensures an assessment of the interoperability between all the modules composing the KDD process. It also allows an assessment of the acceptability of the whole DSS/KDD.

The purpose of this contribution is the proposition of a KDD process with an emphasis on the evaluation of its modules. It is a generic approach that helps evaluators to evaluate DSS/KDD process by assessing all its components (systems). Usually, in such systems, decision makers think about the performance of the classifiers and the data miners think about the interestingness of patterns provided by the data mining algorithms rather than about the quality of the whole DSS since all components are related. This is because KDD modules are also strongly linked.
Our proposal motivates the system developer to envisage the three prescribed modules integrating evaluation modules when he/she builds up a DSS/KDD process.

3.4 Our proposition for the evaluation of MDSS based on KDD

Context-aware computing appeared along with mobile platforms in order to adapt the applications to these new more restricted devices. So far, many definitions and evolutions have covered the concept of context awareness. Indeed, the notion of context has been the object of numerous definitions and evolutions, among which that of Sottet et al. [201] was chosen to base our proposal. Actually, they define the context as a triplet (User, Platform, Environment).

From this triplet and a set of definitions from the state of the art, we proposed a context model (presented in section 3.4.1). As we are concerned with the evaluation of DSS used in the cases of mobility, this model was enriched to take into account the specifications of the MDSS based on KDD. Basically, it would be significant to be aware of the necessity to consider the context of use of each module during its evaluation. Mobility offers the opportunity to gain awareness of the individual and their interactions

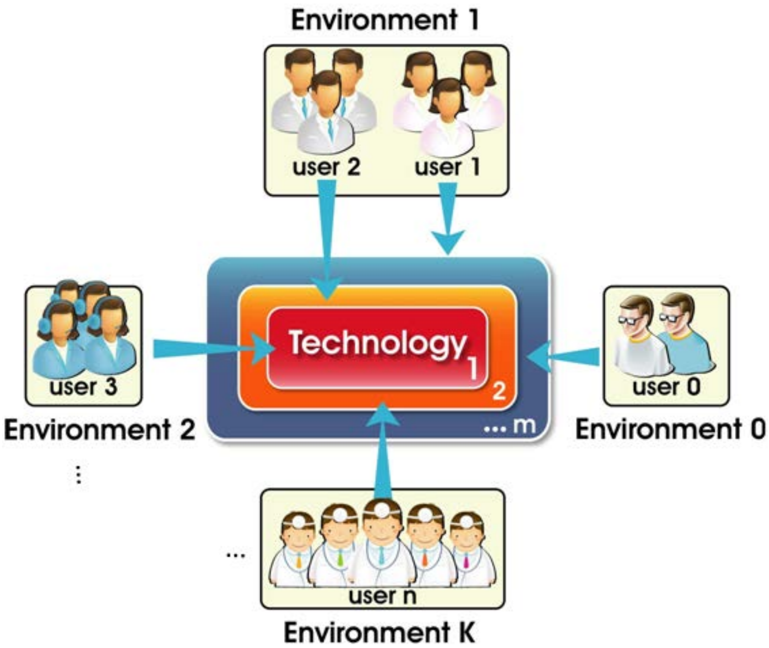

Figure 3.5: How does the context change (example)

with their ever changing surroundings. This context also includes situational awareness. That means variety of data about individuals, and any other relative data based on mobile context will be used to deliver a fuller study about the quality of MDSS.

In the next subsection, we concretize the context of use definition, based on the work of Sottet et al. [201], and we present our proposed context model. Afterward, we provide our context-based evaluation approach for MDSS/KDD.

3.4.1 Context modelling

As previously mentioned, we basically based our proposal on the definition of context proposed by Sottet et al. [201]. The change of the context of use is due to three reasons which are (see Figure 3.5):

- Variety of users types of the system,

- Variety of the used technology (i.e the interaction platforms),
- Variety in the environment in which the user is placed (located).

We propose to model the context of use by a class diagram, presented (User, Platform Environment) in Figure 3.6.
The main class of this diagram is **Context**. Three other classes have an aggregation relationship[2] with the context class. These consist of the classes representing the **User**, the **Platform** and finally the **Environment**. For each component of the class context, we propose the attributes used or defined in the state of the art (see section 1.5.1). We will specify, at the presentation of the classes, the possible attributes or the values of these attributes. However, we add other possible attributes (not coming from the state of the art), while letting this model rather generic and adaptable to the needs of each class (according to the MDSS to be evaluated).

Thus, the **User** beyond its characterization by possible attributes as his/her identity (name, first name) or his/her age, sex, etc., has a certain level of skills (e.g. expert, novice). Among the skills, we can distinguish professional skills (Competence), interaction skills (Familiarity with the platform) and experience in using the KDD module(s). Competence in interaction means that the user is generally neither a programmer nor a mobile developer (in computer science) but more often a non-expert who can be introduced to perform various interactions. Furthermore, we distinguish the type of interactions, mainly involving the use of keyboard and mouse or a touch screen. Finally, the user can perform one or more activities, each of which has an objective and may consist of a set of tasks.

The characteristics of the **Platform** are also taken into account and are important for the characterization of the context. Just like the *User*, the *Platform* can be defined as an attribute that characterizes the platform. A platform can be a smartphone, computer, tablet, etc. It generally proposes a surface of display and an audio feedback which completes the visual feedback of one screen for example.

[2] An aggregation relationship is used to model a relationship between objects where one object contains another.

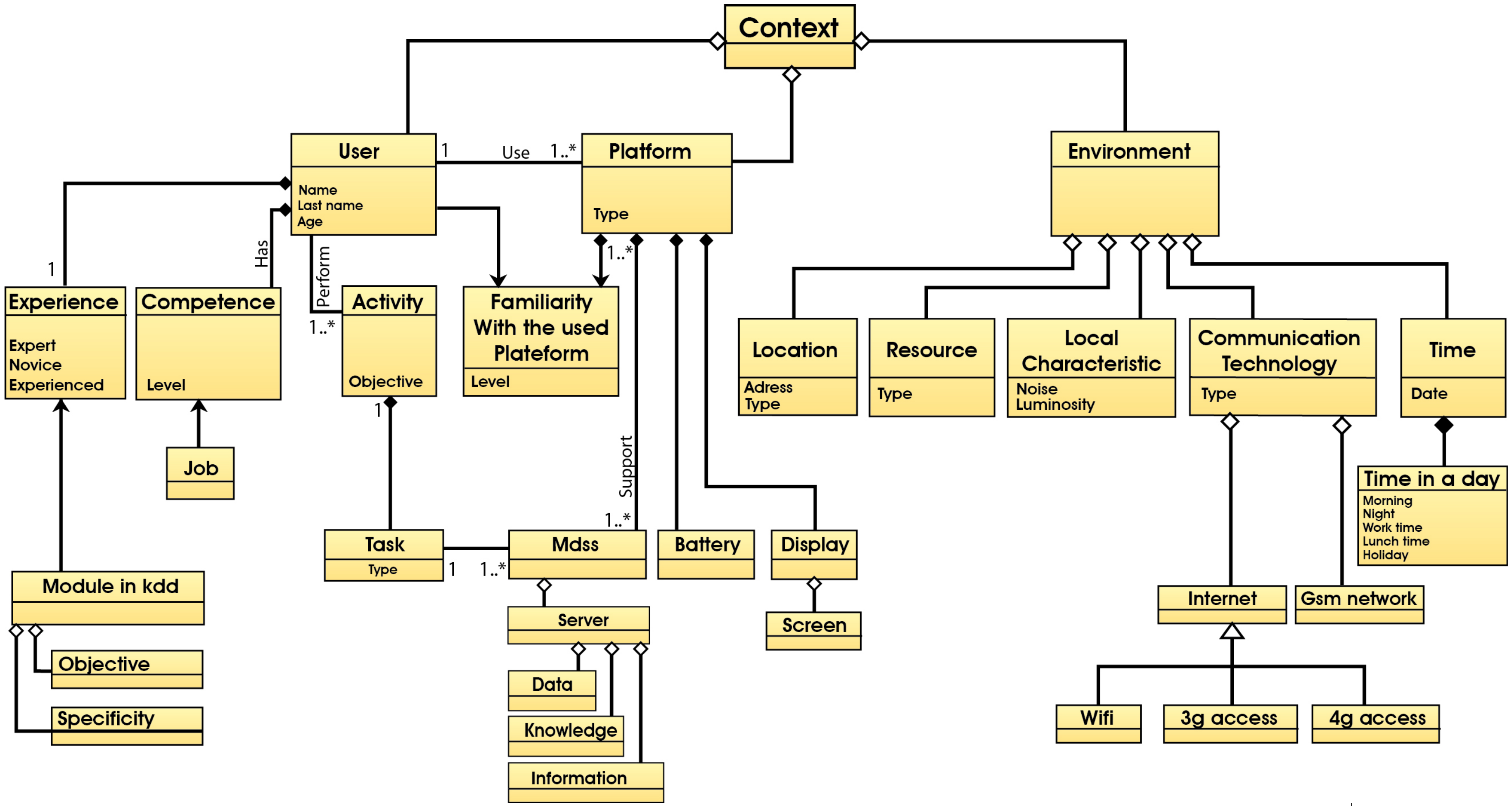

Figure 3.6: Global modelling of the context of use

As shown in this figure (Figure 3.6), MDSS/KDD is included in the context model; it is directly related to the *Platform* because it is already installed on it, and indirectly with the *User* because of the task and the interaction (set of actions constituting the task) between the user and the device when using the MDSS/KDD. The MDSS is connected to a server in which Data and Knowledge are stored. It is also worthy to note that the MDSS is generally linked to the task stemming from the activity of the *User*.

Finally, the **Environment** integrates the location (e.g. place where the user is situated), as well as the type of location (localization) (e.g. laboratory, school, home, etc.), the available resources nearby (e.g. printer). Finally, information about the environment is also represented (local characteristics) and the possibility of using technologies of communication, such as the use of a wireless network.

As shown in the literature, these relatively-generic classes allow to characterize a context of use of a Platform in a certain Environment by a User. Let us finally notice that the proposed model is general enough to be able to be instantiated according to the evaluator's needs. It can also be completed or adapted if needed.
We consider that a change of context is directly due to the modification of one (or of several) element (s) of the context (for example the modification on the element User involves a change of context). A change of context, thus, takes place during the modification of one of the elements (User, Platform Environment). The collected contextual data would be stored in a data base and further used in the interpretation of evaluation results. Thus, we can define gaps and their causes related to the context of use features (User, Platform, Environment).

3.4.2 Context-based evaluation of MDSS/KDD

Nowadays, MDSS are often used in changing environments [27]. Although moving away from the desktop model brings a variety of new situations in which an application may be used. When we are face to mobile situations, the context of use changes because of one or several factors. So, the manner

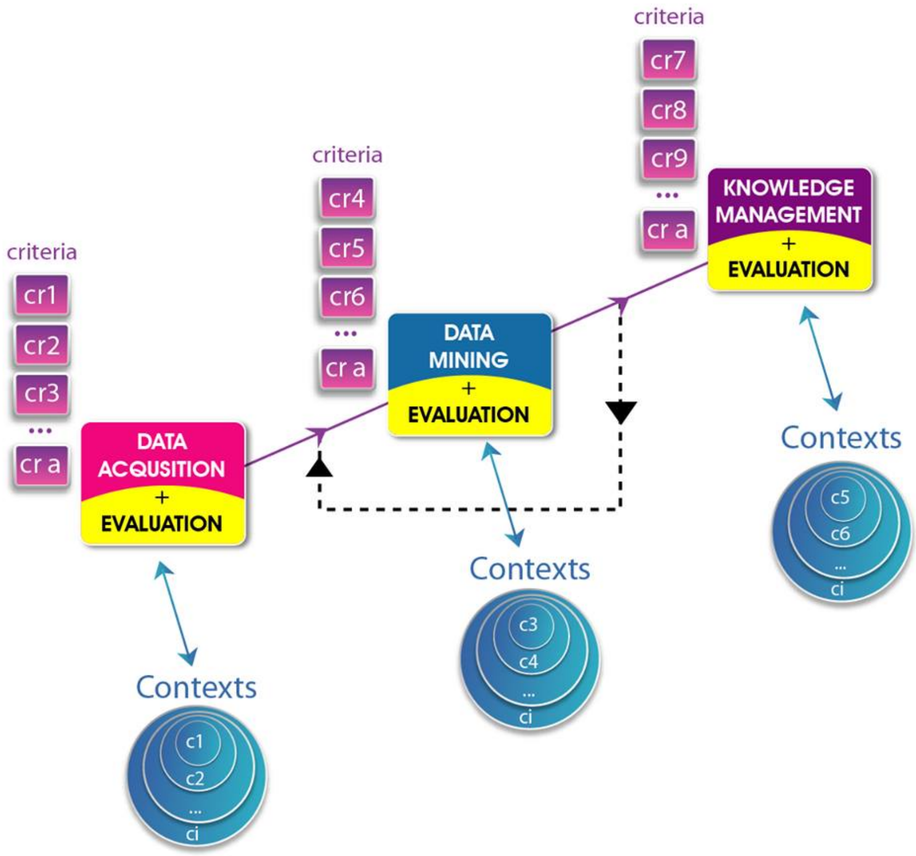

Figure 3.7: Context-based evaluation of MDSS/KDD process

of use of the MDSS will also change. MDSS are used in different situations, environments, under different conditions and, by different profile users. As shown in Figure 3.7, our proposition consists in achieving the evaluation test of the different KDD stages in different contexts of use.

Based on this proposal, the criteria defined in section 3.3 from this chapter need to be measured in several contexts of use. For each context, the criteria which are sensible to the context change would be measured. In the remaining part of this report, these criteria are called **context-aware** criteria [28].

When the context of use changes, the performance of the MDSS can also change; but there are some functional aspects that are independent of the context. For this reason, we consider that the criteria that are not context-aware can be sufficiently evaluated under only an instance of context of use. However, the evaluation of context-aware criteria should be evaluated in several (different) contexts of use. The variety of contexts is mainly due to three causes which are the user, the technology and the environment. It is to be recall that the context changes when, at least, one or several attributes (User, Platform and Environment) change its/their value.

As the *Quality-in-Use* was proposed as a common quality factor to be evaluated in all the KDD stages (see section 3.3), we present, in section 3.4.3, our proposed method for the evaluation of this quality factor.

3.4.3 Quality in use evaluation

In this subsection, we present the principle idea of our proposed method for Quality-in-Use (QinU) evaluation of mobile systems, particularly mobile software pertaining to the KDD process. This method, depicted in Figure 3.8, takes into account the change of the context of use. We evaluate such systems in different contexts for the consideration of the mobility during the evaluation. Thus, our method is based on two types of data:

- (i) Contextual data which describes the context in which the mobile application is used. These data are designed according to our proposed context model (see Figure 3.6). These data are acquired either by the user (who participates to the evaluation), or by the evaluator(s) (who manage the evaluation process) At best, they can also be automatically captured by the evaluation tool (proposed in section 3.5).
- (ii) Evaluation data which include data collected during the interaction with the mobile system in order to achieve a given task (a task is composed of a set of actions).

All these data should be collected and stored in a data base. This stage constitutes the first module, called **Acquisition module**, of our method. After preparing the evaluation data necessary for the measurement of the QinU

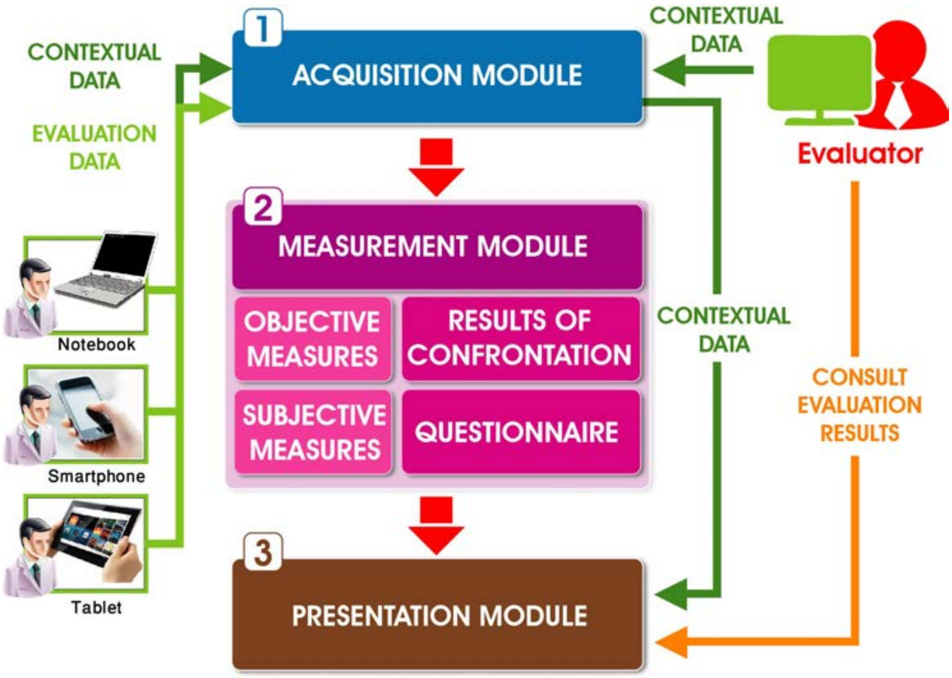

Figure 3.8: The proposed method for quality in use evaluation

criteria, the second module allows the measurement of the QinU criteria. The objective criteria which are *Efficiency* and *Effectiveness* are measured by the evaluation tool (proposed in section 3.5).

Effectiveness is measured by evaluating the degree of correspondence between the actions of test users compared to the actions specified by the evaluator for a specified task. The following equation can be adopted to give measure to the effectiveness criterion [26]:

$$\mathit{Effectiveness} = n/m * 100$$

With n is the number of correct user actions and m is the total number of actions that are accomplished by the user.

Let us present a simple example, one considers that a user was invited to achieve a task defined by the evaluator(s) as follows:

- Textfield, NameClient, Jean Jacques, window1, filling, 00:00:25

- Textfield, AgeClient, 26, window1, filling, 00:00:03
- Button, SaveButton, Save, Window1, Click, 00:00:01
- Button, SaveButton, Save, Window2, Click, 00:00:03

By assuming that the invited user filled only one text field, then pressed a button in the same window, the following information is immediately sent to the monitoring:

- Textfield, NameClient, Jean Jacques, window1, filling, 00:00:55
- Button, SaveButton, Save, Window1, OnClick, 00:00:03

In this case, the user achieved only 2 correct actions. Therefore, the effectiveness of this task is equal to 50%.

Likewise, **Efficiency** is measured by evaluating the correct user actions compared to those specified by the expert and not exceeding a given time (T) calculated from the following equation:

$$T = TE + \Delta$$

with TE is the time spent by the evaluator to achieve the same task under the same conditions (the same context of use) and Δ is a time margin, chosen by the evaluator to add a tolerance field (much time) to the user to achieve his/her required task. The following equation 3.4.3 can be adopted to give measure to the efficiency criterion [26]:

$$Efficiency = p/m * 100$$

With p is the number of correct user actions and not exceeding the time defined by the evaluator (T).

The subjective measures are quantified using the questionnaire technique in order to measure the user's attitudes towards the application.

The main objective of our method is to help the evaluators to detect QinU problems that can encounter the end user when performing his/her tasks in mobile context. The detection of these problems is useful to the designer to improve the quality of the mobile system. For this reason, we propose a third module which presents the evaluation results to the evaluator regarding the context of use. Using these results, we are able to validate the influence of the contextual information, collected in the first stage, on the measurement of the QinU criteria.

3.4.4 Conclusion

There is a continuous progress in the field of mobile technology which creates an evolution of MDSS that can be evaluated through several classical approaches. Nevertheless, several constraints can be considered, such as the characteristics of platforms, the profile of the end users or the specifications of the environment. In this view, our method is presented for the evaluation of MDSS/KDD process, which takes into account the continuous changes of the context of use. So, we have defined a context model that classifies contextual data that can influence the performance of this kind of application.

In the next section, we design an Evaluation Support System (ESS) which allows the capture of interaction data and contextual data during the use of the different modules of the MDSS/KDD process. In addition, it enables the measurement of the criteria approached in section 3.3.

3.5 Our Evaluation Support System

Controlling and monitoring the quality of software engineering processes are important aspects for all software development organizations [203]. In order to help evaluators, we conceive an Evaluation Support System (ESS) that monitors and measures all the criteria defined in the previous subsections [30]. It allows the capture of actions achieved by the user, the calculation of durations and the detection of errors [25] [26].
This system is intended for the evaluation of DSSs which are interactive

systems. It detects different types of interactions between the user and the machine, then compares tasks performed by the user with others set by the evaluator(s) in order to provide the measures of criteria.

The ESS is able to connect to the three sub-systems constituting the DSS (see Figure 3.9). Our ESS is able to connect to the repositories (data repository, patterns repository and the knowledge base) to evaluate the corresponding criteria as detailed in Table 3.5. It is worthy to recall that a quality factor is composed of a set of criteria.

Table 3.5: Objective criteria calculated from the different repositories

The corresponding repository	**Quality factor/Criterion**
Data repository	Recoverability criterion (Quality of data factor)
Patterns repository	Interstingness quality factor
Knowledge base	Quality of promotion and Classifier performance quality factors

For example, the ESS needs to be connected to the Patterns repository to measure Interstingness quality factor including the support criterion, the confidence criterion, etc.

Besides, all questionnaires were addressed through a set of user interfaces of the ESS. Indeed, there is absolutely no need for the use of a pen and paper. All evaluation data are stored in an evaluation database. Thus, through simple queries, the ESS calculates all the required measures. Moreover, it offers the user interfaces containing the different questionnaires.

Each time the user makes an event on a user interface, four messages are, continuously, sent to the ESS describing the following information:

- **Message 1:** it presents the details related to the user's interactions. It describes the name and the type of the graphical component (a button for example) used to perform a required action, the text written on the component (examples: save, cancel...), the time spent by the user

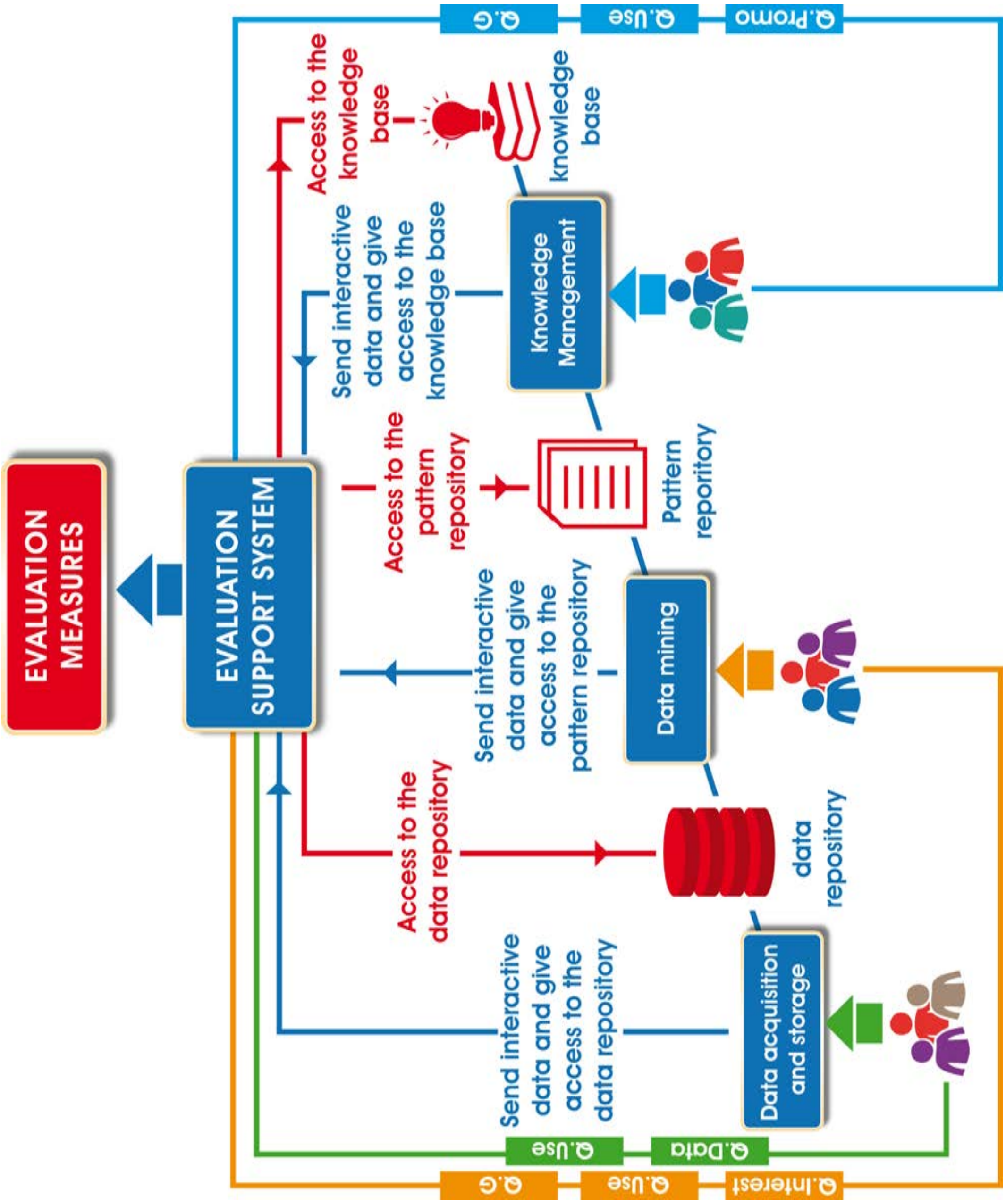

Figure 3.9: The evaluation principle

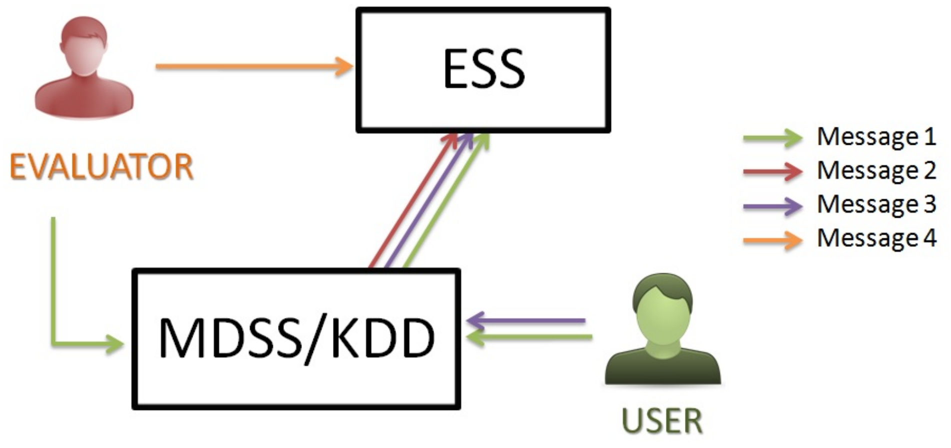

Figure 3.10: Messages received by the ESS

to achieve an action, and the type of this action (examples: click, fill in. . .).

- **Message 2:** it includes the characteristics related to the used platform. When the user starts to use the user interface that works on the mobile device, a message is automatically sent to the monitoring describing the contextual information of the platform being used.
- **Message 3:** it includes the contextual information related to the user. These data are entered by the user himself/herself before starting the test, using a form. Subsequently, this information will be sent to the monitoring to be stored in the evaluation database.
- **Message 4:** it includes the contextual data related to the environment in which the evaluation takes place. A pop up form is displayed in the user interface to allow the evaluator to enter all the required information.

These messages provide the necessary information to assist the evaluator in later assessments [25]. Figure 3.10 shows the different interactions between the mobile system, the developed ESS, the evaluator and the test subject. It is worthy to note that a well established connection between the mobile application and the ESS is required.

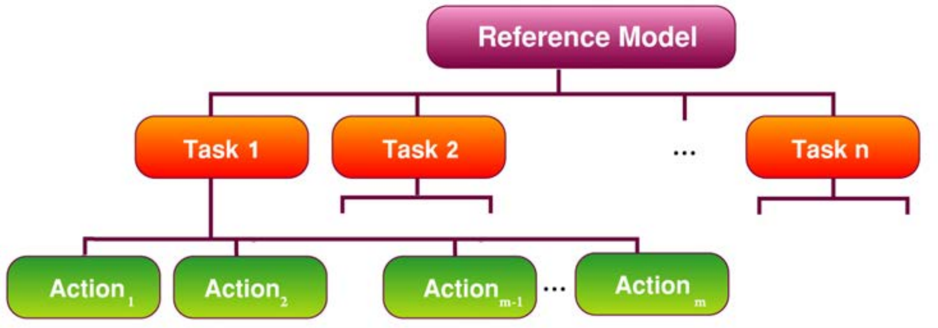

Figure 3.11: The reference model architecture

Before starting an evaluation test using the mobile application, contextual data are received (Messages 2, 3, and 4) and stored in the evaluation data base. These data represent the contextual data describing the triplet (user, platform and environment). Then, the test subject (user) is invited to perform a given task among the already defined tasks in the reference model. These tasks are prepared by the evaluator to compare them with others achieved by the users. As shown in Figure 3.11, each task is presented by logical actions performed by the evaluator during the execution of a given task (see the example given in section 3.4.3). When the test subject or the evaluator (when preparing his/her reference model) makes an action through the mobile interfaces, Message 1 will be sent to the ESS. It describes the user's actions (click on the keyboard, click on a button, a selection of an object, etc.).

Once the test subject finishes his/her required task, a confrontation (comparison) between the user activity model and the reference model specified by the evaluator will be performed by the ESS. It is to be noted that the confrontation is based on an algorithm, presented in chapter 4, that compares tasks performed by test subjects with others in the reference model. Afterward, the ESS uses all the received information to measure the different quality factors approached in section 3.3. This part will be detailed in the next chapter.

In this section, we have established a link between our enhanced KDD process and our conceived ESS. We revealed how our evaluation tool is strongly complying with all the KDD process and highly dependent on the user test. So, the ESS would be able to acquire to all the data needed for the measurement of the different quality factors and thus supports the evaluator(s) of the MDSS/KDD.

3.6 Synthesis and conclusion

DSS/KDD are widely complex systems. They contribute to the decision making process by providing highly potential knowledge. The extraction of this knowledge is not an obvious module, it is rather a sequence of modules which their design becomes more and more difficult because of the progressing needs of the users [90]. Nevertheless, these needs concern not only reliable knowledge, but also high quality level of all the modules that ensure the extraction of this knowledge. Despite the long time spent in the evaluation, all KDD process was assessed in order to discover defects in their implementation. Our objective is to save the cost of maintaining them, on the one hand, and to improve their ability of knowledge discovery, on the other. In this chapter, we have presented our proposed approach for the evaluation of mobile decision support systems based on knowledge discovery from data process.

Our first contribution lies in the evaluation of the whole KDD process to provide a framework to evaluate its applications in a distributed way, while the previous research works focused only on the evaluation of the Data Mining module and paid no attention to the other modules. Yet, building quality models is only one stage towards the evaluation in the KDD process. We have firstly enhanced the evaluation inside this process in order to evaluate all its modules (Data acquisition and storage, Data mining, and Knowledge management).

As all these modules are considered as interactive systems that can be used in mobility situations, we have established a context-based approach that takes into account the change of context of use due to this mobility. This approach was proposed in order to achieve an effective evaluation of the whole decision support system. Moreover, to facilitate the task of the evaluators, we have proposed an evaluation support system that monitors and measures all criteria presented in this chapter.

In order to help the evaluator(s) in applying this approach, we have also conceived an Evaluation Support System that monitors and measures all criteria previously defined in our proposed quality models.

In the next chapter, we will focus on the application of our proposed approach in a real case study. The objective of our case study is to evaluate the MDSS/KDD used in the intensive care unit of the Habib Bourguiba Hospital in Sfax, Tunisia. It was designed to help the doctors who are the current users of the system to understand, predict and prevent nosocomial infections.

4. CEVASM: Context-based EVAluation support System for Mobile decision support systems based on KDD process

4.1 Introduction

In the previous chapter, we presented an approach of evaluation of MDSS/KDD process which enhances the evaluation task in the KDD process and takes into account the variety of contexts due to the use of mobile devices. For this purpose, we have proposed to use subjective and objective criteria. Some of them come from the ISO/IEC 25000 standard, while the others are not extracted from the literature. The proposed evaluation approach deals with the whole KDD process and takes into account the specifications of each module. Moreover, we have shown that there is a possible solution to implement an interactive system that gives the evaluator(s) the required support.

This chapter presents the set of the realized developments to put into practice the proposed solution. It contributes to the existing tools by offering not only the support remotely, but also several detailed and summarized synthesis of the obtained evaluation measures. This system is called CEVASM: Context-based EVAluation support System for Mobile decision support systems based on KDD process.
This chapter contains three sections. We begin with the presentation of CEVASM and its main objectives. Then, we describe our evaluation process adopted from the standard ISO/IEC 25040 as well as developments allowing the creation of CEVASM. Finally, we conclude this chapter with a synthesis of our previously undertaken research work.

4.2 Presentation of the developed evaluation support system

CEVASM is a Web application that was developed using the programming language PHP 5. It allows measuring not only subjective evaluation criteria but also the objective ones referring to the whole MDSS/KDD process. The goal of developing CEVASM is to help evaluators to assess the quality of an MDSS/KDD process.
In the next subsection, we detail the implementation of our evaluation support system (CEVASM) which is composed of several components which are:

- *Quality in Use* evaluation component, which is executed when we are face to any modules among Data acquisition, Data mining, Knowledge management.

- *Quality of Data* evaluation component, which is executed when we are face to the Data acquisition module.
- *Interestingness* evaluation component, which is executed when we are face to the Data mining module.
- *Classifier performance* evaluation component, which is executed when we are face to the knowledge management module.
- *Quality of promotion* evaluation component, which is executed when we are face to the knowledge management module too.
- *Global evaluation* component, which is executed after the assessment of all the KDD modules. It concerns the whole MDSS/KDD process.

As explained in chapter 3 (see section 3.5 and Figure 3.9), each evaluation component contains three modules:

- The module of the detection of the contextual data in which the evaluation is achieved.
- The module of the evaluation which allows the execution of the evaluation.
- The module of presentation of the evaluation results, which serves to supply the obtained results.

In addition, CEVASM offers a fourth module which allows the preparation of the questionnaires as well as the so-called reference module, as explained in chapter 3. All these modules are detailed in the following subsections.

4.3 The evaluation process

The implementation of our system CEVASM is based on the evaluation process proposed by the standard ISO/IEC 25040 (presented in section 2.3.7, chapter 2). We recall that this process is structured around five stages: the specification of the requirements of the evaluation, the specification of the evaluation, the design of the evaluation, the execution and the conclusion of the evaluation. Our process extends over these stages as shown in Table 4.1 in order to evaluate all the KDD modules.

It is worthy to note that CEVASM provides its users (evaluators) with five sessions as illustrated in Figure 4.1:

- *Admin* session, which allows the administration of the whole web application (CEVASM).
- *New expert* session, which allows to add a new expert (evaluator) in the database (see Figure 4.1). He/she can have an access to CEVASM through the *Expert* session.
- *New experiment* session, which allows the execution of a new evaluation test.
- *Expert* session, which allows to a known (already registered) expert to have access to his/her session. This session offers to him/her the possibility to specify the requirements of the evaluation (see Table 4.1, stage 1) and to prepare the quality models, the questionnaires as well as the reference module (see Table 4.1, stage 2). Stage 1 and Stage 2 should be established before any evaluation test (Stage 3).
- *Experiments* session, which allows a known expert to have access to the evaluation database and be briefed about the collected results.

4.3.1 Stage 1: The specification of the evaluation requirements

This stage presents the phase of design of the evaluation. Table 4.1 specifies the objective and the product parts to be included in the evaluation. The context diagram, presented in Figure 4.2, presents the stakeholders who participate in the preparation of this stage. The success of the evaluation

Table 4.1: The evaluation process

The evaluation process as defined by ISO/IEC 25040		**Our evaluation process**
1. The specification of the requirements of evaluation	Establish the purpose of the evaluation	Support the evaluators in the MDSS/KDD assessment.
	Obtain the software product quality requirements	The stakeholders types of MDSS/KDD are: developer, evaluator, and user
	Identify product parts to be included in the evaluation	Three KDD modules (Data acquisition, Data mining, Knowledge management)
	Define the stringency [a] of the evaluation	All the calculated measures are calculated by CEVASM.
2. The specification of the evaluation	Select quality measures (evaluation modules)	Prepare the formulas that calculate the obtained measures.
	Define decision criteria for quality measures	Numerical targets are defined by the evaluator (by the definition of the reference model).
	Define decision criteria for evaluation	Preparation of the quality models (see chapter 3).
3. The evaluation design	Plan evaluation activities	The used techniques are: Monitoring, Questionnaires, and formulas
4. The execution of the evaluation	Make measurements	According to the evaluation plan, CEVASM detects data needed for the measurement
	Apply decision criteria for quality measures	We calculate final measures according to the defined numerical targets.
	Apply decision criteria for evaluation	Produce the evaluation results.
5. Conclude the evaluation	Review the evaluation result	The evaluator(s) carries out a review of the evaluation results.
	Create the evaluation report	The evaluation report is created, including the requirements of the evaluation, and analyses performed.
	Review quality evaluation and provide feedback to the organization	Feedback from the review should be used in order to improve the evaluation process and/or techniques
	Perform disposition of evaluation data	Evaluation data are archived.

[a] The evaluation of stringency shall be defined in order to provide confidence in the software product quality according to its intended use and purpose of the evaluation

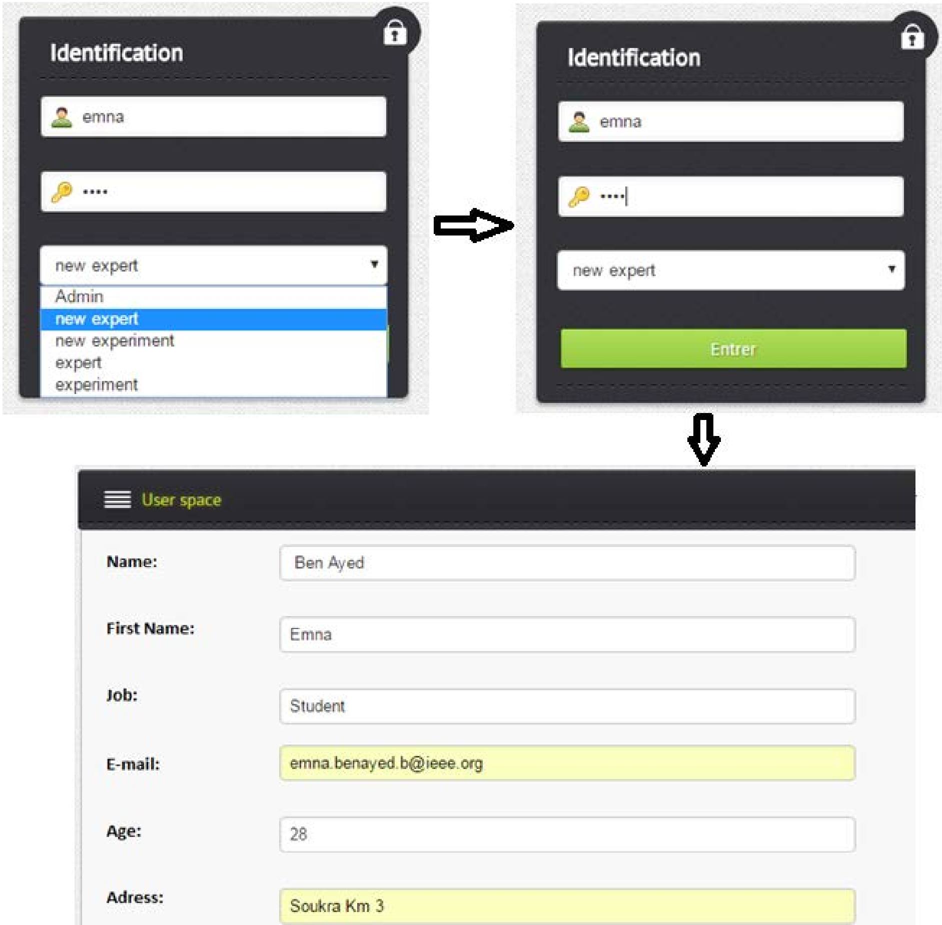

Figure 4.1: Screen-shots taken from CEVASM: Add a new expert

design is based on the implication of a project team, including the stakeholders from various domains. These stakeholders, mentioned in Figure 4.2, are successively described.

- **The KDD expert:** He/she is specialized in KDD and acquires knowledge and skills thanks to his/her experience in this domain.
- **The developer:** His/her first objective is to translate the needs of the evaluator(s) into ready functional specifications for implementation. The realization of a web application is also the role of this stakeholder who is specialized in computer science. He/she creates the system by implementing the user interfaces and the functional modules.

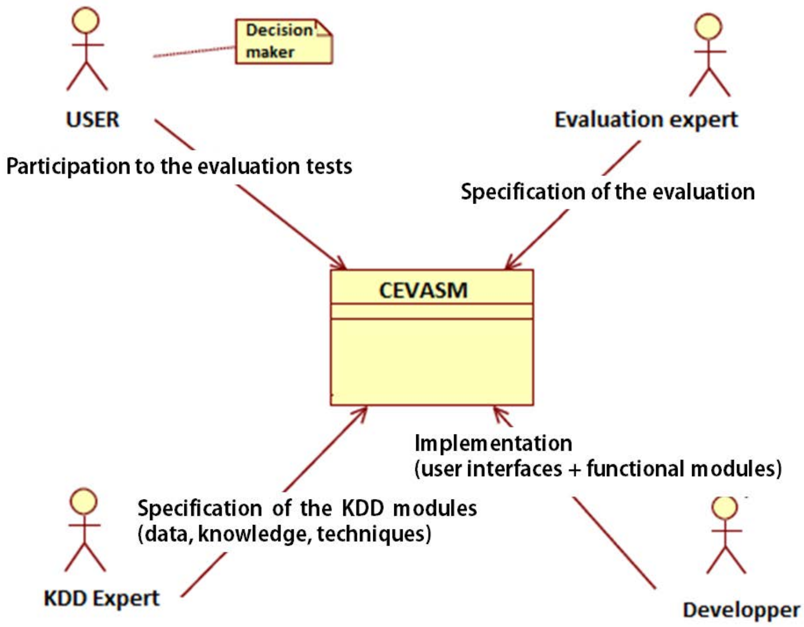

Figure 4.2: CEVASM context diagram

- **The user:** He/she is the future test user of the system. He/she is also the decision-maker who will be asked to evaluate the performance of the MDSS.
- **The evaluation expert:** He/she can, in fact, have variable skills (ergonomist, specialist in HCI, etc.). CEVASM would finally be offered to this stakeholder. For this reason, he/she is included at this stage. This allows us to better understand his/her needs and requirements.

As mentioned in Table 4.1, the evaluator(s) has/have to identify the products parts to be included in the evaluation. In our case, three modules are identified: Data acquisition, Data mining, Knowledge management. So, three connections have to be established:

- **Connection 1 (CEVASM, the database)**: CEVASM must be connected to the database in which the collected data are stored. For example, in our case, the connection was established thanks to the script presented in Figure 4.3.

- **Connection 2 (CEVASM, the patterns repository)**: CEVASM must be connected to the repository in which patterns generated from data using a data mining algorithm. This connection allows CEVASM to evaluate the interestingness of the patterns.
- **Connection 3 (CEVASM, the knowledge base)**: CEVASM must be connected to the knowledge base in order to assess the classifier performance of the Knowledge management module.

Thanks to these direct connections, we provide stringency and confidence to our developed CEVASM.
It would be necessary to mention that, in general, the repositories are stored in a distant server, which is configured to allow remote connections thanks to TCP/IP protocol[1] (see Figure 4.3).

4.3.2 Stage 2: The specification of the evaluation

As shown in Table 4.1, this stage is composed of three phases. The first one was clearly described in Chapter 3 (Figure 3.3). We have firstly specified the quality models which are composed of a set of criteria and prepare the formulas that calculate the objective measures. Then, the evaluation expert and the expert in KDD should define the questionnaires on which CEVASM is based to measure the subjective criteria.
In the following subsection we describe how the can evaluator(s) prepare his/her/their question(s). Then, we present how he/she/they prepare(s) the reference model(s).

[1]The Transmission Control Protocol (TCP) is one of the main protocols of the Internet protocol suite. It originates in the initial network implementation in which it complements the Internet Protocol (IP). Therefore, the entire suite is commonly referred to as TCP/IP. TCP provides reliable, ordered, and error-checked delivery of a stream of octets between applications running on hosts communicating by an IP network.

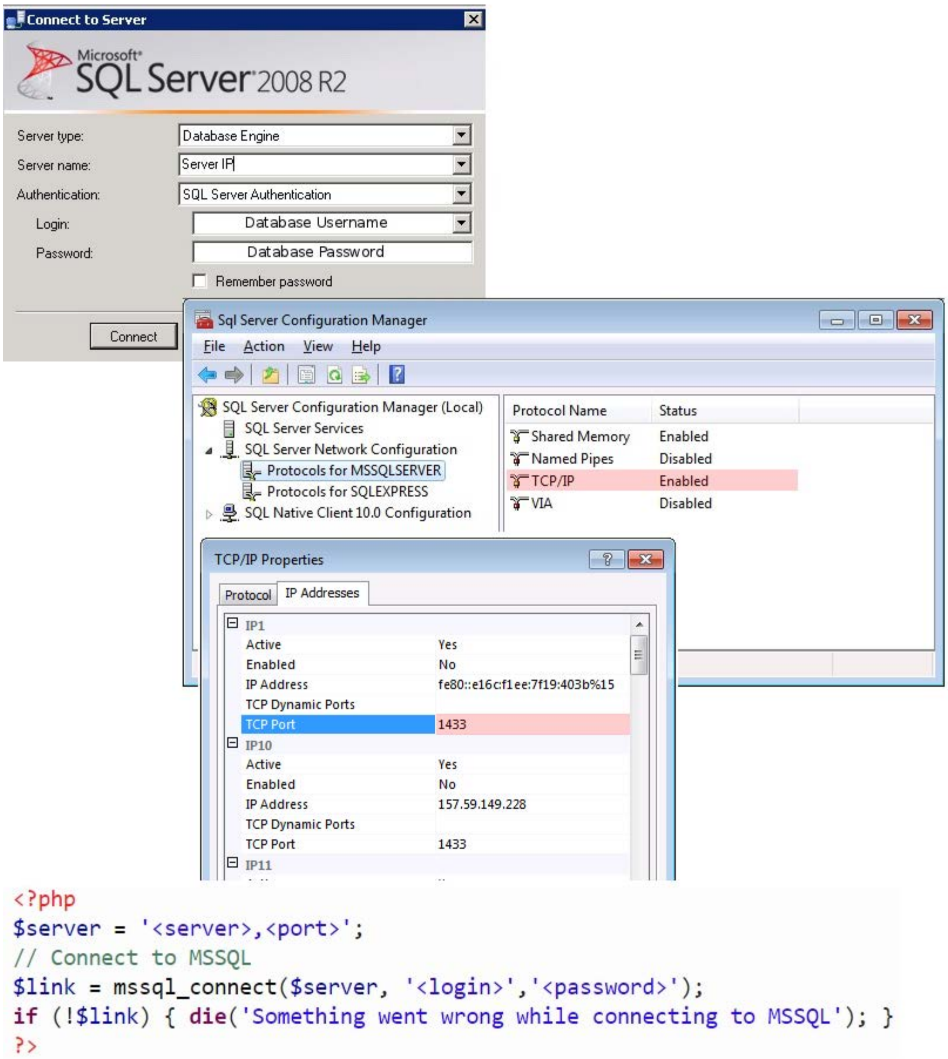

Figure 4.3: A possible configuration using SQL server

Preparation of the questionnaires

This module allows the construction of the questionnaires of evaluation which is a direct and purely subjective way for detecting the defects of an interactive system. Our tool allows the configuration of these questionnaires and their associated questions. Indeed, the creation of a question requires the identification of the statement, the associated criterion, the name of its questionnaire, as illustrated the Figure 4.4. Thus, CEVASM would be able

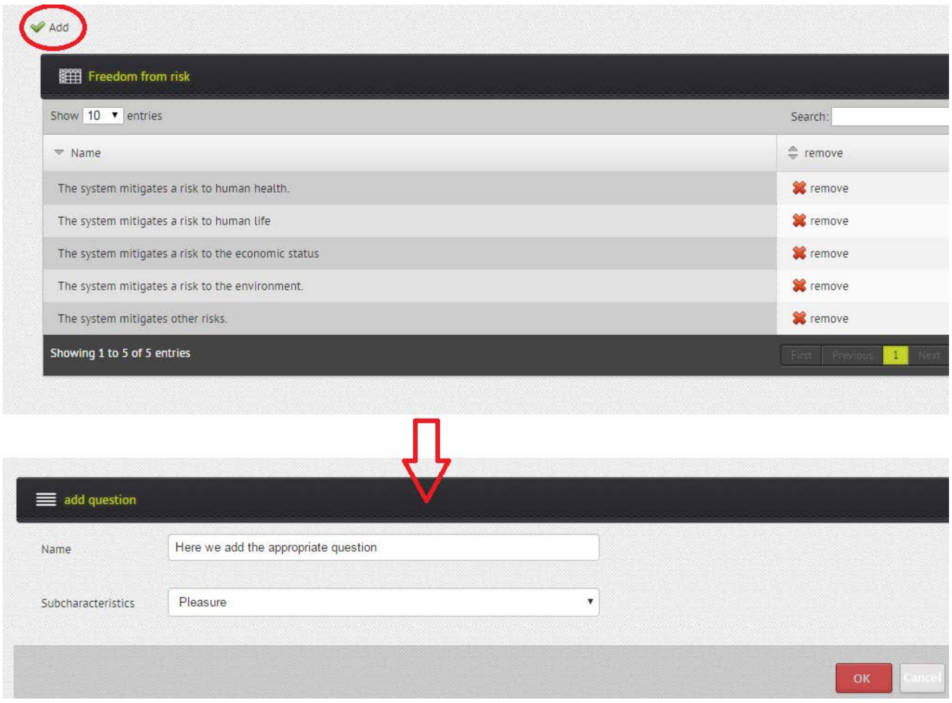

Figure 4.4: Screen-shots taken from CEVASM: preparation of the questionnaires

to generate the list of the questions referred to the selected criterion.

Preparation of the reference module

According to Senach [196], the evaluation is based on a comparison of the reference model (realized by the evaluator) and a model of the evaluated object (realized by the users who participated to the evaluation tests) in order to draw conclusions about the quality of the evaluated object.
As illustrated in Figure 4.5, CEVASM allows the evaluator(s) to prepare his/her/their reference model(s).

During this phase (preparation of the reference module), the evaluator is invited to add the tasks[2] in his/her reference by entering the name of the task, its objective, etc (see Figure 4.5). Then, the are detected by CEVASM

[2]It should be recalled that a reference model is composed of several tasks, each task is composed of a set of actions (see chapter 3, section 3.5

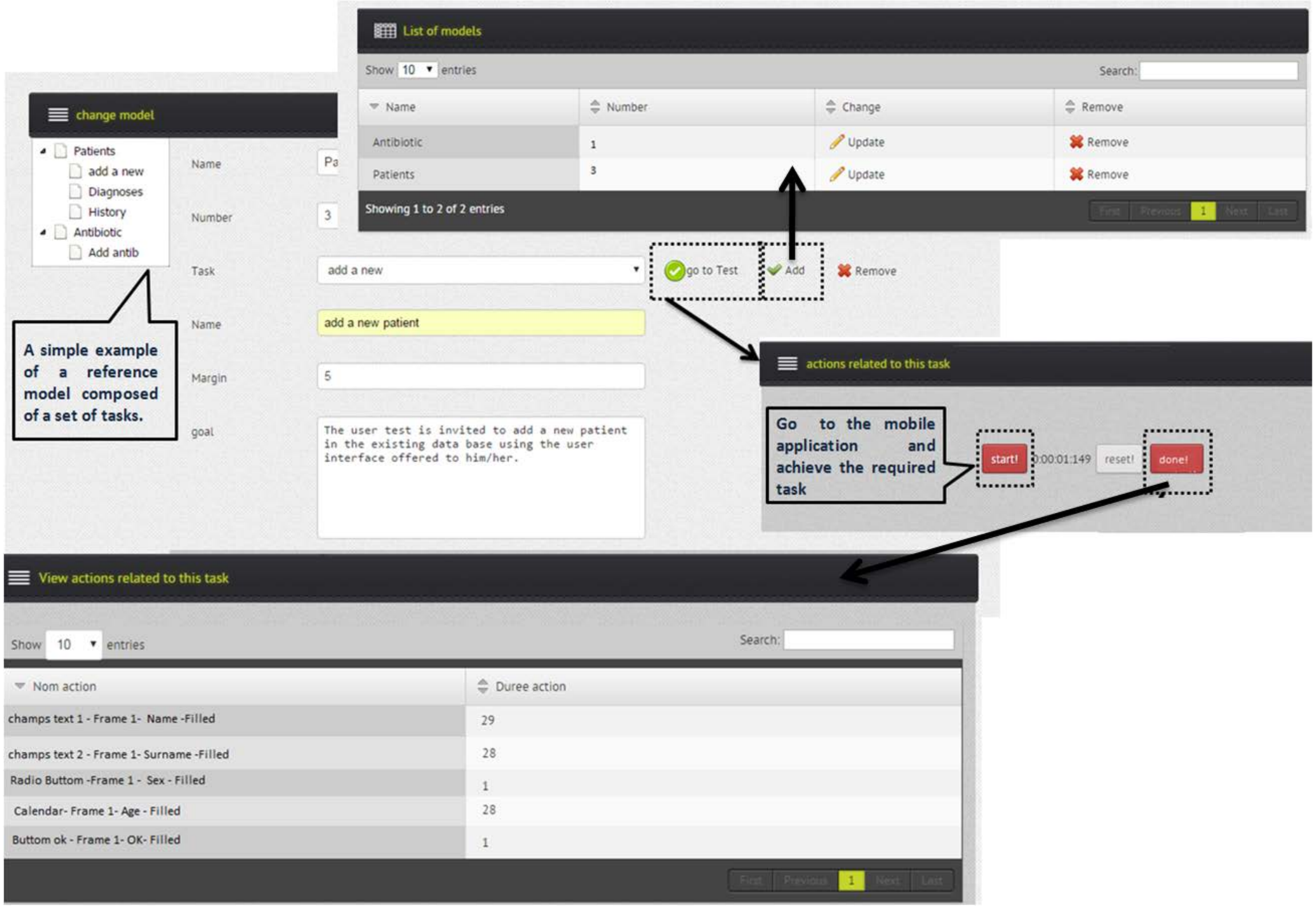

Figure 4.5: Screen-shots taken from CEVASM : preparation of the reference module

thanks to the latter's ability to connect to the MDSS/KDD via TCP/IP protocol. In the appendix B (see Figure B.1), we present the script that allows CEVASM to capture the sent data that would be further treated to be stored in the evaluation database. During the use of the evaluated system by the evaluator, the proposed system allows the capture of the following items, already presented in Chapter 3 (section 3.5). These data are received thanks to these two functions added in the MDSS:

- **F1:** allows the sending of the data which describe the interaction (human-MDSS). It includes the description of the achieved tasks such as the used component (button for instance), the time in which the action was performed, the fonts, and the form to which the component is pertaining. These data would be used to measure the criteria referring to the *quality in use* factor.
- **F2:** allows the sending of the data which describe the platform being used at the time of the evaluation. It includes the screen size, the battery performance characteristics, etc.

These captured data are saved in the evaluation database and would be further compared to other data captured when achieving the evaluation tests.
When the evaluator achieves the preparation of his/her reference model, the actions achieved are displayed to him/her (see Figure 4.5). Then, the numerical targets are immediately calculated (through the formulas defined in Chapter 3: section 3.4.3 and section 3.3) and saved in the evaluation database. After the preparation of the reference model, the evaluator(s) apply(ies) several tests to a given system in order to detect its defects.

4.3.3 Stage 3: The evaluation design

In this stage, the evaluator has to plan the activities during the evaluation. So, we present its corresponding sequence diagram (Figure 4.6) which shows the different interactions between the MDSS, CEVASM, the evaluator and the test subject user who is invited to participate to the evaluation tests.

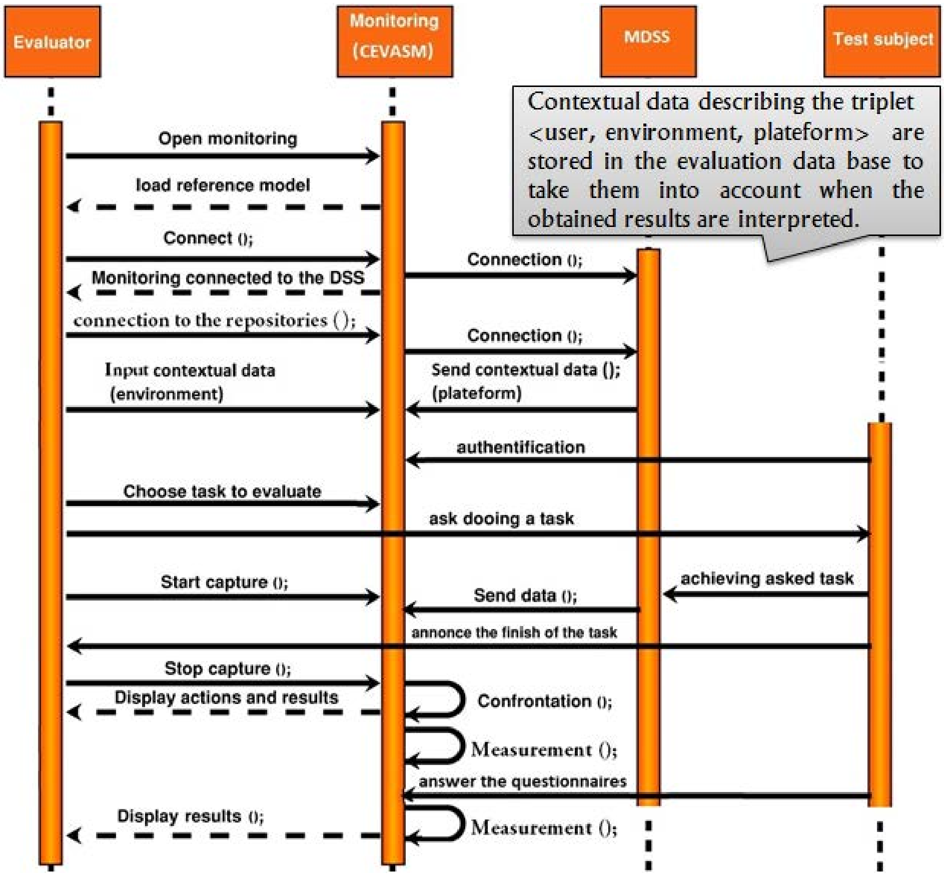

Figure 4.6: The evaluation design: A sequence diagram

As mentioned in this figure, before starting the evaluation tests, contextual data are entered and stored in the evaluation database, which represent the contextual data describing the triplet (user, platform, and environment). Afterwards, the user is invited to perform a given task among the tasks already defined in the reference model (He/she got 5 minutes of familiarization before starting the capture). The interaction data is collected by the same way used when collecting the data pertaining to the reference model. However, the interaction data is saved in another table in our evaluation database.

It is noteworthy to mention at this level that a user is invited to achieve several tasks in different contexts of use. Thus, we can collect a volume of data that allows us to draw significant conclusions and underline the stringency of our proposal.

4.3.4 Stage 4: The execution of the evaluation

This stage includes 4 steps, each of which is the following subsections.

1. Make measurement

As all data needed for the measurement is already collected by importation of the different repositories, remote capture, and subjective attitude (questionnaires), CEVASM has become able to make the required measurement of the criteria presented in section 3.3. The criteria are classified into two types (see Table 4.2): objective and subjective. They are measured through different modules of calculations which can be classified as follows:

- Objective measurement module: which calculates the objective criteria from data which do not depend on the user. The measured data depend only on the data stored in the repositories. The formulas we used are presented in Chapter 3 (section 3.4.3 and section 3.3).
- Subjective measurement module: which calculates the subjective criteria from the collected based on responses to the different questionnaires. As shown in Figure 4.7, each subjective criterion (such as context coverage) is measured through a set of questions. These questions are studied and proposed by the expert on the basis of the definition of each criterion, according to its source.
- Confrontation module: which performs the comparison between the reference model and the models performed by the users in order to measure *Effectiveness* and *Efficiency*. Mainly, this module checks the correspondence and the duration[3] between the tasks achieved by the users and the others achieved by the evaluator. The algorithm 1 explains better this module.

[3]The delay means that the user takes much time compared to the one taken by the evaluator to achieve the same task.

Table 4.2: Classification of the criteria

KDD module	**Quality factor**	**Criteria**	**Type**	**Measured from**
All modules		Effectiveness	Objective	Interaction data
		Efficiency	Objective	Interaction data
	Quality in use	Satisfaction	Subjective	Questionnaire (Q.Use)
		Freedom from risks	Subjective	Questionnaire (Q.Use)
		Context coverage	Subjective	Questionnaire (Q.Use)
	Global evaluation	All criteria	Subjective	Questionnaire (Q. G)
Data acquisition and storage	Quality of Data	All criteria	Subjective	Questionnaire (Q.Data), and the data base
Data Mining		Support	Objective	patterns repository
		Confidence	Objective	patterns repository
		Lift	Objective	patterns repository
	Interestingness	Kulczynski	Objective	patterns repository
		Novelty	Subjective	Questionnaire(Q.Interest)
		Surprisingness	Subjective	Questionnaire(Q.Interest)
		Comprehensibilty	Subjective	Questionnaire(Q.Interest)
Knowledge management	Classifier performance	All criteria	Objective	Knowledge base
	Quality of promotion	All criteria	Subjective	Questionnaire (Q.Promo)

Figure 4.7: The questionnaire used for measuring the context coverage criterion

Algorithm 1 Confrontation

```
Input: Act_ref: reference actions, Act_test: user actions, Time_ref: reference action duration, Time_test: test actions duration, s: test duration, t: tolerance, N: size of table;
Output : State, Time;
  j: Integer, k: Integer
  j ← 0;
  for i ← 1 to N do
    ▷ verify if the action is correct
    if (Act_ref [j] = Act_test [i]) then State [i] ← true;
    ▷ verify if there is a delay
      if (Time_ref [j]+t < Time_test [i]) then Time [i] ← delay;
      End if
    else
      k ← i+1;
      ▷ verify if the user made a mistake and was able to catch up later
      for (k ← i+1 to N) do
        if (Act_ref [j] = Act_test [k]) then
        ▷ the correct action was found
          State [k] ← true;
          Time [k] ← delay;
          j ← j+1;
          i ← k;
        else
        ▷ the correct action was not found
          status [k] ← false;
          mistake ← mistake+1;
        End if
      End
    End if
  End for
```

Algorithm 2 decision criteria for quality measures: example

1. **Input** : Time_ref: reference action duration, N: size of table;
2. **Output** : m: task reference duration
3. **Begin**
4. ▷ *Browse the reference action table to calculate the duration of the entire task*
5. **for** (i ← 1 **to** N)
6. **Begin** m ← m + Time_ref [i]; **End**
7. ▷ *m is a numerical target for this task*
8. **End**

2. Apply decision criteria for quality measures

In this phase, the system has to calculate the final measures according to the defined numerical targets (defined by the evaluator(s) or calculated from the reference model). The algorithm 2 presents an example of a numerical target which is the duration needed to achieve a requested task by the evaluator (m). In fact, to calculate the efficiency, we are based on duration. The duration needed to achieve a requested task by a user do not permit to deduce the efficiency measure unless compared with m. Moreover, the evaluator has to specify the two thresholds minconfidence and minsupport, on which the algorithm of association rules extraction[4] is based, are defined by the KDD expert at this stage.

[4]We remind that the support and confidence measure how interesting a rule is. The minimum support and minimum confidence are set by the users, and are parameters of the *A priori* algorithm for association rule generation. These parameters are used to exclude rules in the result that have a support or a confidence lower than the minimum support and minimum confidence respectively. The strengths of this algorithm is that it finds all the itemsets that meet the minimum support and the minimum confidence criteria

Algorithm 3 Produce the evaluation results: Example

```
Input : m : reference duration, s : test duration, mistake: number of mistakes N: size of table;
Output: Effectiveness, Efficiency
Begin
  if (s < m) and mistake = 0 then
    Efficiency:=100;
    Effectiveness:=100;
  else
    Effectiveness:= ((N − mistake)/j)∗100;
    Efficiency:= 100 - ((s/m)−1)∗100;
  End if
End
```

3. Apply decision criteria for evaluation

At this level, the developed system should be able to produce the final results. We present through algorithm 3 an example of producing the final results referring to *effectiveness* and *efficiency* criteria. In Chapter 3 (section 3.4.3), we present an example that explains this algorithm.

4.3.5 Stage 5: Conclude the evaluation

Once the results become available, the user can review them through CEVASM. In fact, the system allows the evaluator to query the evaluation database. The system is able to visualize the results and facilitate their interpretation. Actually, there are three options of visualization:

1. **Visualization with a focus on the criterion:**

   These user interfaces (see Figure 4.8 as example) allow the evaluator to directly get the required measures without writing SQL[5] queries.

[5] SQL (Structured Query Language) is a special-purpose programming language de-

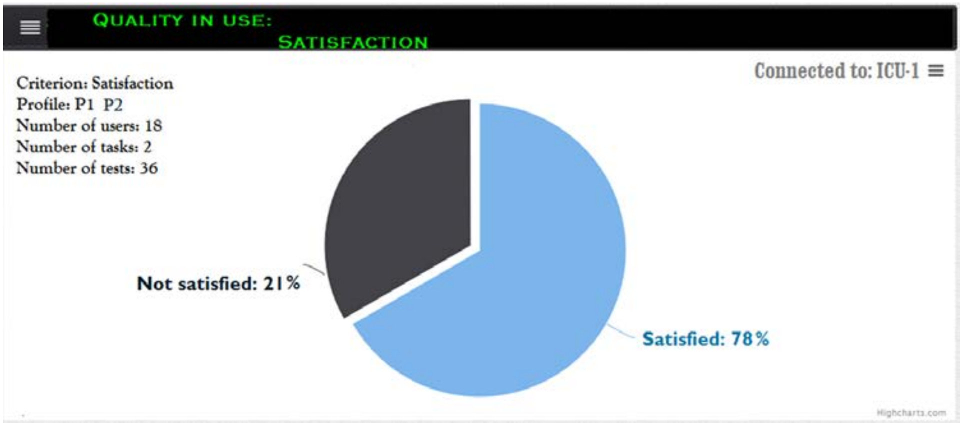

Figure 4.8: Example of a user interface allowing the visualization of *satisfaction* measure

From this point of view, the system gives support to the evaluators to make the appropriate decision to maintain the evaluated MDSS/KDD, if it is needed.

2. **Visualization with focus on the context:**

   These user interfaces (see Figure 4.9 as example) allow the evaluator to have a more accurate idea about the MDSS/KDD in relation to its context of use. In fact, these interfaces compare a set of measured criteria in different contexts of use. Figure 4.9, for example, compares the criteria pertaining to the knowledge management module. This comparison was established based on the change of the occupation of the user profile. As illustrated in this figure, the evaluator(s) can easily draw conclusions based on these user interfaces.

3. **Global visualization:**

   This user interface (see Figure 4.10) allows the evaluator to have a global idea about the whole MDSS/KDD. This view ignores all the possible changes that can affect the use of the different KDD modules, as well as the differentiation between the modules. For example, the *Satisfaction* measure, shown in Figure 4.10, presents the average of all the Satisfaction measures collected in the database and pertaining to

signed for managing data held in a relational database management system.

Figure 4.9: Example of the obtained results with focus on the context of use, particularly to the user profile

the three KDD modules (Data acquisition, Data mining and knowledge management).

Up to now, CEVASM is not able to create an automatic evaluation report. As perspective, we intend to add this functionality in our system in order to provide it with the ability to propose requirements and build analysis (see our future research works in chapter 5, section 5.7).

4.4 Conclusion

In this chapter, we presented all the developments realized for the implementation of our proposed evaluation support system (CEVASM) which gives support to the evaluators of MDSS/KDD process. We began by the presentation of the proposed system, its objectives and then the process followed for the evaluation. Next, we specified our realized evaluation support system which allows to make subjective and objective evaluations, as it was approached in the previous chapter. Through this chapter, we presented the stages proposed in the evaluation process, while detailing the various stages:

Figure 4.10: The user interface allowing a global visualization of the obtained measures

the two stages of specifications including their corresponding implemented module, the stage of the evaluation design, the stage of execution of the evaluation and the conclusion stage including their corresponding implemented modules. The proposed system considers all the quality factors proposed in the previous chapter.

In the following chapter, we will put into practice the developed system through a real case study with the aim of illustrating the complete functioning of the CEVASM as well as the feasibility of the proposed approach.

5. APPLICATION OF THE PROPOSED APPROACH FOR THE EVALUATION OF A MDSS BASED ON A KDD PROCESS

5.1 Introduction

In the previous chapter, we have presented the achieved developments which allowed the total realization of the context-based evaluation support system for the evaluation of MDSS/KDD process.

In this chapter, we are going to focus on the application of our approach through a case study. This latter concerns the evaluation of a Mobile Decision Support System based on a KDD process which is used in the Medical field (MDSSM/KDD). This mobile system is used in Habib Bourguiba hospital, Sfax, Tunisia, to support the staff in making appropriate decision(s) in order to prevent nosocomial infections.

This chapter is structured as follows. In the first section, we introduce the general context of the work by introducing the nosocomial infections and previous proposed systems dealing with this problem. In the second section, we present the MDSSM/KDD to be evaluated using our evaluation support system CEVASM. Then, in the last section, we discuss the risks of validity of our undertaken work. Finally, we finish this chapter by presenting our future works.

5.2 General context in medical field

As previously introduced, this section presents the general context of the work. We begin by introducing the nosocomial infections and then, we briefly describe previous proposed systems dealing with this problem.

5.2.1 Nosocomial infections

Nosocomial infections NI are the hospital-borne infections. They represent one of the major problems of the public health; they are contracted in a health care establishment. An infection is considered as such when it was absent at the time of the admission of the patient [78]. When the infectious

state of the patient in the admission is unknown, the infection is classically considered as hospital-borne if it appears after 48 hours of hospitalization. These infections can be directly caused by the care or simply arise during the hospitalization independently of any medical act. In the Intensive Care Unit (ICU), the problem of NI is worrying as the patients who are hospitalized there are fragile.
The fight against NI is a complex problem. Most of the patients admitted in an ICU require much care and a continuous supervision (24h/24h). They are often connected to machines (artificial breathing apparatus, electrocardiogram, electric syringe, etc.) and/or attached to catheters (venous catheters, urinary probe, etc). These patients are often very sensitive to any new germ. In every appearance of infection (nosocomial or not), a sample is sent to the laboratory to realize an antibiogram. According to the result of the antibiogram, an antibiotic treatment is prescribed. The problem of the antibiotic treatment is that a germ can be sensitive to an antibiotic in a period and resistant few weeks later. This sensibility is different between individuals.

5.2.2 Previous works

Several works have been carried out to fight against these NI. For example, a study on the prevalence of the risen NI in Habib Bourguiba hospital, Sfax, Tunisia, showed that 17,9% of 280 patients hospitalized in the hospital, between April 17th, 2002 (midnight) and April 18th, 2002 (midnight), were victims of an NI [111].
Several theses and research works have been published by the physicians of this Unit [75, 85]. For the realization of these studies at that time, physicians were asked to fill information in forms. Such information was then seized and stored in an Excel file and analyzed, further, by a statistics software such as SPSS[1]. Such tool allows to obtain only *classic* statistics (percentages, averages, comparison of averages, etc.) [32].

Later, based on the requirements of physicians, some other research works proposed information systems to supervise NI based on the techniques of data Mining [125, 31]. These works show their efficiency and their

[1] SPSS Statistics is a software package used for logical statistical analysis.

capacity to produce useful rules. But, as mentioned in the articles, their direct use by physicians was difficult.

Therefore, Ltifi *et al.* suggest a novel HCI-enriched approach for DSS development. It is called the UP/U approach [134]; it was applied in this same context (NI). Indeed, a medical DSS based on KDD was implemented. The objective was to predict the case of a NI every day during the period of hospitalization. Thanks to this system, the patient can keep track of his/her state getting worse or better from the point of view of the risk of having an NI during his/her stay in an ICU. Using this DSS, data are extracted from the microbiological reports, from the medical interviews with nurses and physicians, etc. These data are generally temporal. The implemented system allows the exploitation of the gathered temporal data, for the dynamic acquisition of the useful knowledge for making decision(s).

With the appearance of mobile devices that proved their ability to support information systems, physicians expressed their needs to have a mobile application that offers the same functionalities on their mobile devices.

5.3 MDSSM/KDD: Mobile Decision Support System based on KDD process and used in the Medical field

The increase in the use of Mobile Decision Support Systems (MDSS) in healthcare has stimulated the need for the ICU for such systems to predict NI. The growing demand for such system and its implication to the healthcare services is evident. Physicians need to have idea about their patients status wherever they are. Moreover, they need to be notified if there is a serious risk of NI. The nurses also need to input or update the data that describe the status of the patient, which are stored in database.
For the reasons cited above, we have started our thesis project by analyzing, designing and developing such system. Then, we have focused on its evaluation based on our proposed approach presented in Chapter 3.

In the next subsections, we detail the three modules composing the MDSSM/KDD. We remind that designing and developing an MDSS do not present a contribution. However, these phases are needed to obtain an MDSS that, after its evaluation,can be based on our proposed approach, used in the ICU of Habib Bourguiba Hospital.

As it was presented in our first contribution, KDD process can be composed of only three modules (Data acquisition and storage, Data mining, and Knowledge management) instead of four ones (Data acquisition and storage, Data Mining, evaluation and Knowledge management). In the Appendix C, we present the implementation activity used for each module of the MDSSM/KDD process. In the following paragraphs, we present successively the referred KDD process as approached in Chapter 3.

1. Data acquisition and storage

This module allows the seizure of the personal data of the patients as well as the data concerning their hospitalization. The stages of pre-processing consist in building a corpus of the specific data, processing absent data, cleaning the data, selecting attributes and then transforming these data to be usable by an algorithm of knowledge extraction. These stages are crucial for the search for the relevant information necessary for the decision-making.
It is worthy to note that the sub-modules of selection, cleaning and transformation of the data were developed within the framework of Another Master's research work [208]. Our task was to provide a mobile application that allows the acquisition of the personal data of the patients as well as the data concerning their hospitalization via a mobile device (see Figure 5.1).

2. Data Mining

In this module, association rules technique is used as a data mining technique within the framework of the prevention of NI. It allows the discovery of the necessary knowledge to interpret the meditative data of the patients. In this case, the discovered knowledge is expressed with rules that supply the

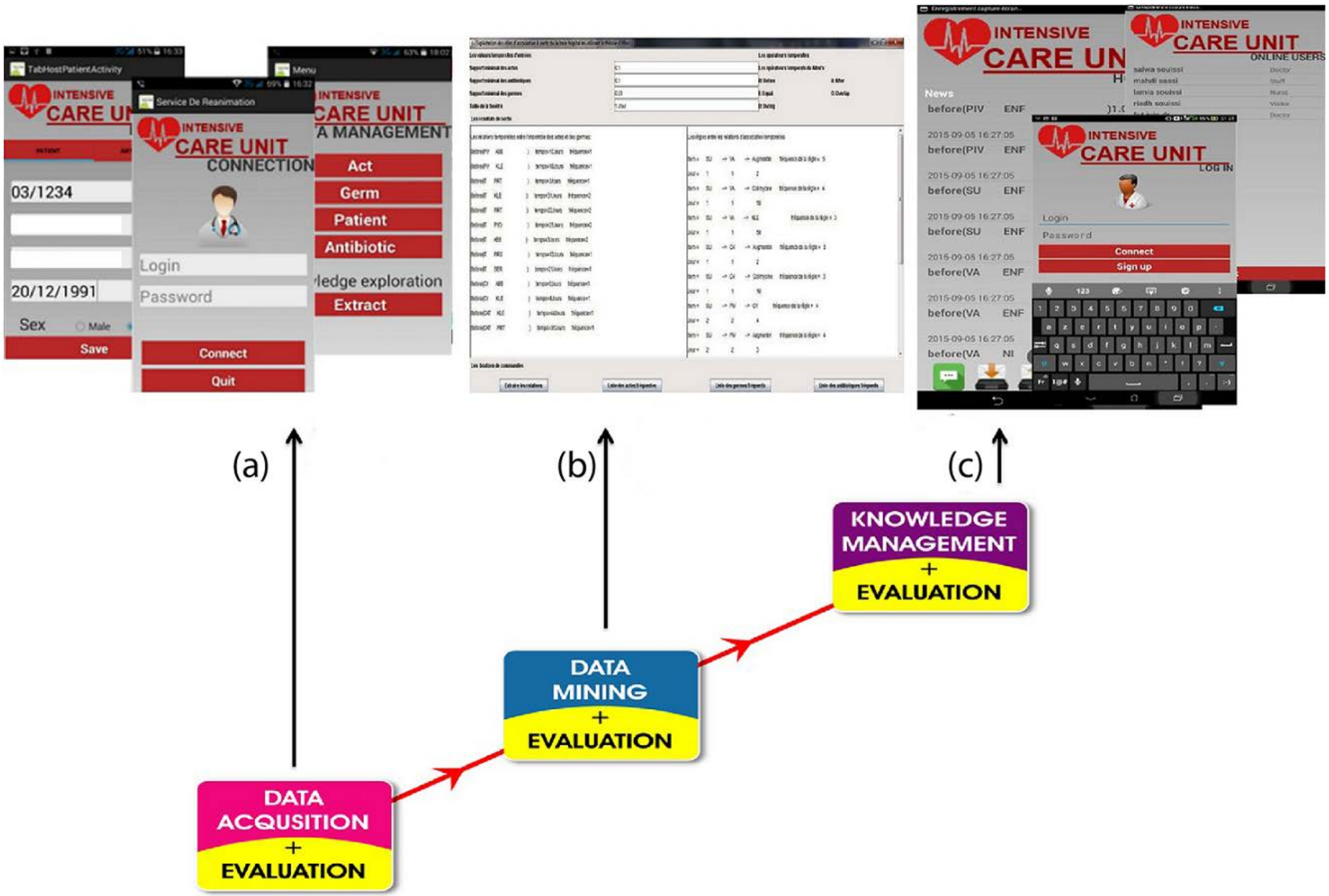

Figure 5.1: The implemented sub-systems

decision-maker with an explanation of the decision. The association rules algorithm was implemented within the framework of another work of Master's degree [190]. At this level, our task was to offer the physicians free access to the discovered rules stored in the knowledge base using their mobile devices (see Figure 5.1).

3. Knowledge Management

This module has the following objectives:

1. the management of the knowledge base,
2. the integration of the knowledge for the generation of the possible solutions to guide the resolution of the problem of medical decision,
3. the knowledge sharing between physicians.

At this level, our task was to provide a mobile application that allows the first objective as well as the third one (see Figure 5.1). Concerning the second objective, it seems important to develop another mobile application dealing with the integration of the knowledge for the generation of the possible solutions to resolve and/or to prevent NI.

5.4 Case study: Evaluation of the MDSSM/KDD

In this section, we apply our evaluation approach to a concrete case in the medical domain. In doing so, we begin by presenting the different contexts of use in which the evaluation was performed.

5.4.1 Definition of the different contexts of use

During our evaluation tests, we were based on several contexts of use, which were already designed through a context model (see section 3.4.1). We remind that we have adopted the definition of Sottet *et al.* [201]. According to them, the context of use can be defined through three features that are the user profile, the environment and the platform. In this section, we define the characteristics of each feature.

1. Characteristics of the user profiles

The number of participants in the evaluation process was eighteen for each prototype[2]. The evaluation tests were achieved by participants susceptible to use this DSS. Their characteristics are presented in Table 5.1. During the evaluation tests, 33 users participated as users of the MDSS/KDD. The sample of the users is composed of women and men in 20-50 year old. They have different profiles:

- **P1:** physicians from the hospital or the private sector.
- **P2:** trainees in the hospital (not yet considered as physicians)
- **P3:** outside the medical field: It includes students and researchers not in the medical field, and staff members.

Concerning their experience regarding the use of mobile technology, it was determined via the following scale: high and medium. In fact, most of participants (22 among 33) have already used mobile devices in their daily lives.

2. Characteristics of the environments

As we are interested in mobile systems that can be used in different environments, we have repeated the evaluation process in several ones (see table 5.2). Each environment is characterized by its location (at the ICU, at the hospital but outside the ICU, or at home), the time in which the evaluation test was achieved (throughout the working day[3] or after work schedules), Availability of Internet, and Noise.

[2] Each prototype refers to a module among Data acquisition and storage, Data mining, and Knowledge management

[3] According to the schedule of the user

Table 5.1: Profile of the participants in the evaluation process

The prototype	**Gender**	**Average age**	**Familiarity with mobile devices**	**Competence**	**Profile**
Prototype 1	12 men	34	5 high	3 physicians from the hospital	P1
				2 trainees	P2
			7 medium	7 outside the medical field	P3
	13 women	29	8 high	1 physicians from the hospital	P1
				1 staff member	P3
				7 trainees	P2
			5 medium	4 outside the medical field	P3
Prototype 2	8 men	31	8 high	3 physicians from the hospital	P1
				5 trainees	P2
	10 women	27	10 high	1 physicians from the hospital	P1
				9 trainees	P2
Prototype 3	9 men	38	9 high	3 physicians from the hospital + 2 physicians from the private sector	P1
				4 trainees	P2
	9 women	32	9 high	1 physicians from the hospital + 3 physicians from the private sector	P1
				5 trainees	P2

Table 5.2: Environments in which the evaluation was achieved

Environment	**Availability of Internet**	**Place**	**Time**	**Noise**
1	Yes (wifi)	ICU	throughout the working day	Noise[4]
2	Yes (wifi)	ICU	after work schedules	No noise
3	No	ICU	after work schedules	No noise
4	Yes (wifi)	Outside the ICU	throughout the working day	Noise
5	yes (3G)	Home	after work schedules	No noise
6	yes (3G)	Home	after work schedules	Noise

3. Characteristics of the used platforms

The third feature, constituting the context of use is the platform. In Table 5.3, we characterize the mobile devices used in the evaluation tests.

Table 5.3: Mobile devices used for the evaluation

Device	**Operating System**	**Ram Memory**	**Luminosity**	**Screen size**
Smartphone	Android	1 GB	65%	5.5
Tablet	Android	2 GB	65%	10

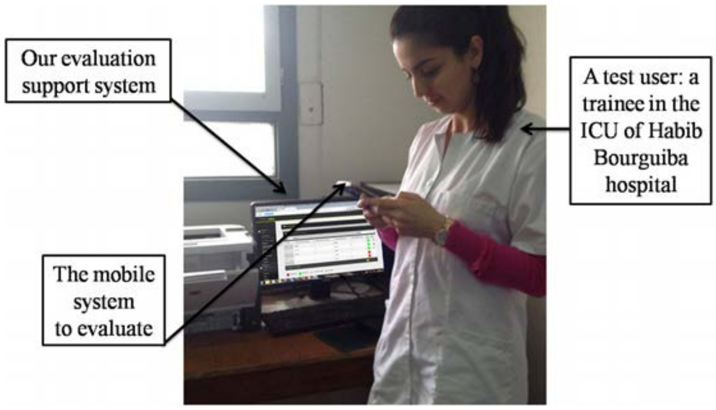

Figure 5.2: A real user test in the ICU

5.4.2 The achieved evaluation tests using CEVASM

Allparticipants were invited to achieve a task or a number of tasks using a prototype (see Figure 5.2). Each prototype is evaluated by 18 participants (as presented in Table 5.1), in several contexts of use. A participant can participate to several evaluation tests, according to his/her availability. As shown in Figure 5.3, before starting the evaluation tests, the evaluator has to introduce the prototype and the purpose of the evaluation to the participant. In addition, he/she has to explain to him/her the evaluation process. The participants hadfive minutes for familiarization before starting his/her first evaluation test using a prototype. It consists in letting the user navigate through the different user interfaces of the prototype. The objective is to give the user tests the possibility to discover it.

Afterwards, the evaluator collects the information about the environment (location, time, availability of internet, etc). After performing the required tasks, the participants were then invited to give answers to the questionnaires in order to express their subjective opinions [15]. These answers were used later for measuring the corresponding criteria, as detailed in Chapter 3. Needless to mention that the evaluator has to connect CEVASM to all the repositories to measure the objective criteria and prepare the reference models before positioning the participant.

In the next paragraphs, we present the different instructions introduced

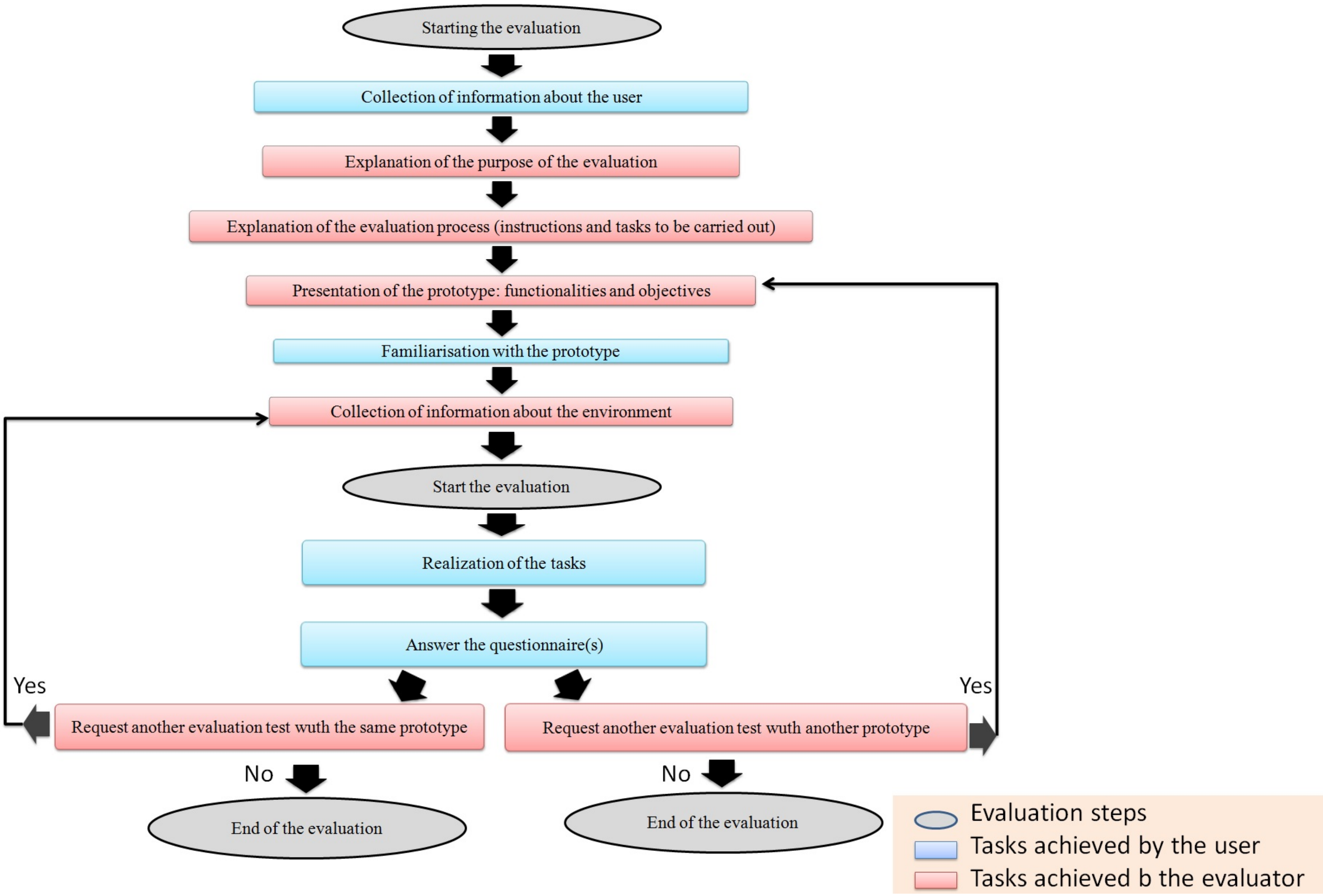

Figure 5.3: Events sequence of the evaluation tests

to the users tests to carry out the evaluation of each prototype referring to the KDD modules.

1. Data acquisition and storage

For this first prototype that refers to the Data acquisition and storage module, we have defined three tasks. Each user test would be invited to achieve at least[5] one task among the following ones[6]:

- Task 1: A new patient is just entering the ICU for the first time. Please enter the required data and store them in database.
- Task 2: A known patient (ID = 19934) left the ICU two years ago. He/she is just entering. Please enter the required data and store them in database.
- Task 3: Please, imagine that you have checked the state of a patient in bed. Try to modify his/her temperature value and store the new value in the database.

2. Data mining

Concerning this second prototype which refers to the Data mining module, we have defined only one task because this prototype is not highly interactive.

- Task 1: This is an MDSSM/KDD that is able to generate rules from the data stored in the database. Please, try to consult these rules.

3. Knowledge management

Concerning this third prototype that refers to the Data mining module, we have defined the following two tasks:

[5] according to his/her availability
[6] The choice of the task is made at random

- Task 1: Please, send a warning to a connected user (physician) to inform him that the patient (ID = 19934) is infected (by an NI)
- Task 2: Please, try to check the newest extracted knowledge (discovered today).

4. Global evaluation

At this level, the user was invited to answer our proposed questionnaire (Q.G). In fact, this questionnaire is composed of two parts: the first one is proposed to the user at the end of the evaluation session to have a global idea about the whole system (MDSSM/KDD), and the second one is proposed to the evaluator. The latter concerns the interoperability between the modules. The evaluator has to check if the sub-systems are well connected to the different repositories (Data base, Knowledge base and patterns repository).

5.5 The obtained results

In this section, we summarize the obtained evaluation measures of the whole MDSSM/KDD and present the drawn conclusions.

5.5.1 The obtained results: Prototype 1

In Table 5.4, we present the results obtained when we established the evaluation of the first prototype referring to the Data acquisition and storage drawing on three user's profiles. The results of evaluating this application show that this prototype presents, in general, high measures in terms of quality in use. This can be further interpreted by the high quality level of the user interface offered to the decision maker(s). Nevertheless, as shown in Figure 5.4 the results show that the average of measures given by users hav-

Table 5.4: Results of quality in use evaluation: first prototype

Criteria	**task 1**	**task 2**	**task 3**	**Profile 1**	**Profile 2**	**Profile 3**
Effectiveness	100	100	87.9	100	100	87.9
Efficiency	85.3	69.4	37.1	54.3	71.3	66.1
Satisfaction				39.6	96.2	98.0
Freedom from risks				64.1	-[a]	-[b]
Context coverage				14.5	-	-

[a]Profile 2 was not invited to answer this question
[b]Profile 3 was not invited to answer this question

ing visual impairments[7] presents more than 75% regarding measures given by those who do not have visual impairments. Thus, we can conclude that user interfaces are easy to be used by users with low or medium visual impairment [30].

Moreover, the average of measures given by users who achieved the tests using the tablet presents 80% regarding measures given by those who used the smartphone (see Figure 5.5).

Thus, we can conclude that the quality of the user interfaces is not very sensitive to the characteristics of the platform being used. However, the change of the user profile has a significant impact on the quality of use. Hence, the change of the context clearly affects the quality in use of this prototype. However, no dependency is reported between the quality of data and the change of the context of use. In Table 5.5, we summarize the drawn conclusions about the dependency between the criteria and the context of use features. It can be noted that **dependency** means that the change of the context feature leads to the change of the measure of the criterion and the absolute value of the variation rate is $\succ$ 15%. For example, the experience of the user in using such prototype (for data acquisition and storage) as well as his/her familiarity with the used platform influence the *effectiveness, efficiency* and *satisfaction* criteria.

[7]Users having visual impairments are invited to participate to the evaluation tests without glasses.

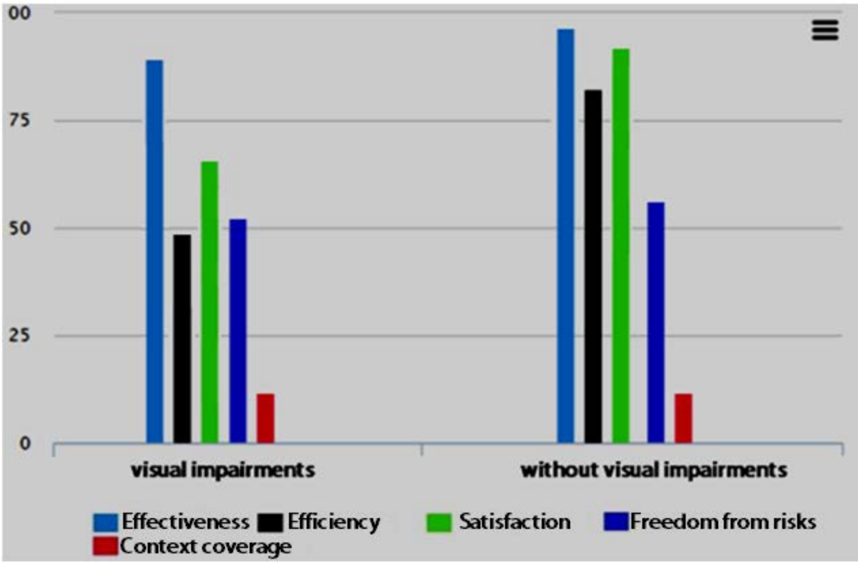

Figure 5.4: The obtained results of the first prototype with focus on the user profiles

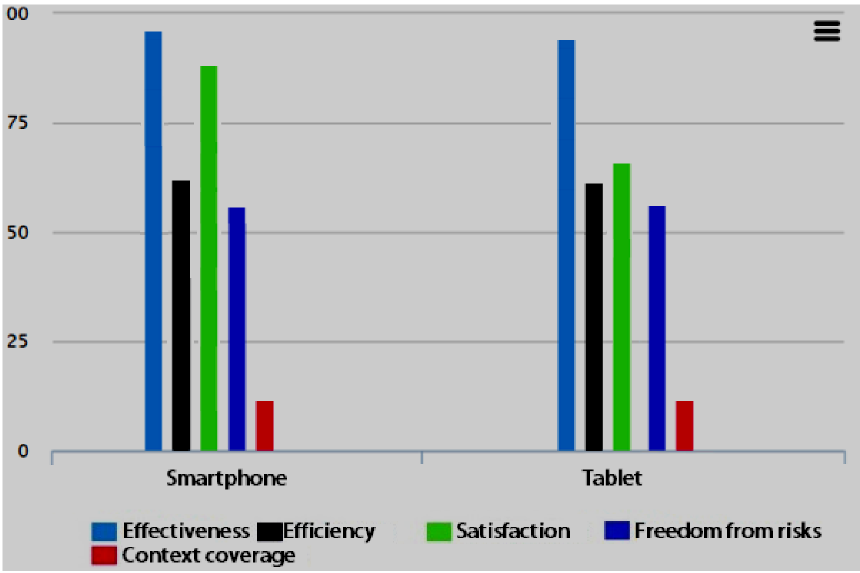

Figure 5.5: The obtained results of the first prototype with focus on the used platform

Table 5.5: The obtained results pertaining to the data acquisition prototype

Quality factor	**Criterion**	**User profile dependency**	**Platform dependency**	**Environment dependency**
Quality of data	all criteria	No dependency	No dependency	No dependency
Quality in Use	Effectiveness	Experience[a], Activity[b], Glasses[c], Familiarity[d]	Screen size, RAM[e]	Noise, Time, Location, communication technology
	Efficiency	Experience, Activity, Glasses, Familiarity	Screen size, RAM	Noise, Time, Location, communication technology
	Satisfaction	Experience, Familiarity, Age, Glasses, Competence[f]	Screen size, RAM	Noise, Time, Location, communication technology
	Freedom from risks	No dependency	No dependency	No dependency
	Context coverage	No dependency	No dependency	Communication technology

[a] Experience in using such prototype

[b] The required task

[c] The test subject has visual impairments or not

[d] Familiarity with the used platform

[e] Random-access memory is a form of computer data storage. It allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory.

[f] Knowledge level in the medical field: the job

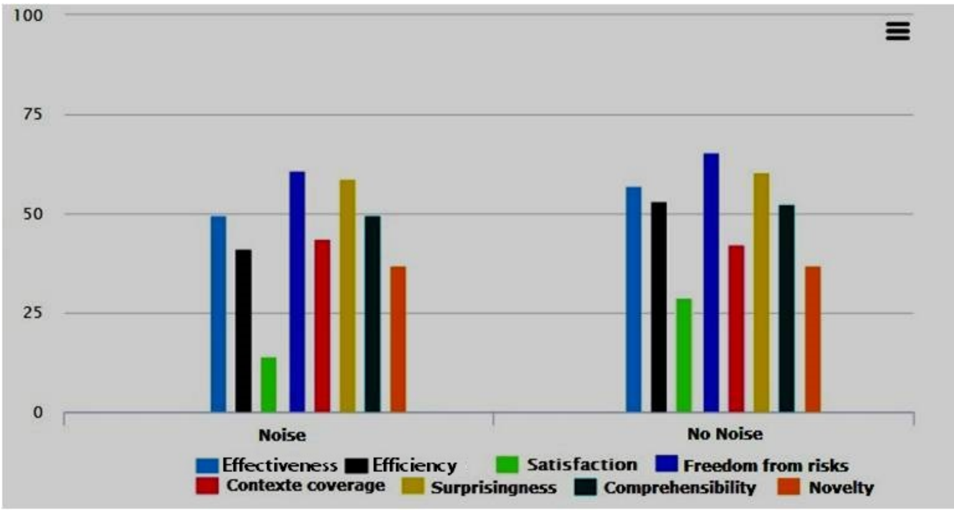

Figure 5.6: The obtained results referring to the prototype of the Data mining module

5.5.2 The obtained results: Prototype 2

Our proposed context-based approach for the evaluation of MDSS uses a larger amount of data that describes the context of use to build precise conclusions. For example, as presented in Figure 5.6 in which we present an example of the obtained results, effectiveness and efficiency present their lowest levels in an environment which is *noisy*. However, we have better values in a *calm* environment. This observation can be interpreted as follows: Effectiveness and efficiency are sensitive to the change of the environment. In fact, a noisy environment can lead to lower levels of Effectiveness and efficiency. So, it would be better to simplify the user interface of the relative prototype in order to increase the obtained values.

Effectiveness and efficiency present two criteria among several other criteria, defined in Chapter 3 (section 3.3), that was measured using our developed tool CEVASM. These criteria were measured in several contexts of use. Each context presents different values from those measured in another context of use. Therefore, many observations allowing further interpretations can be drawn.

Table 5.6: The obtained results pertaining to the data mining prototype

Quality factor	Criterion	User profile dependency	Platform dependency	Environment dependency
Quality in Use	Effectiveness	Activity[a], Glasses[b], Age, Competence[c]	Screen size, RAM[d]	Noise, Time, Location, communication[e]
	Efficiency	Activity, Glasses, Age	Screen size, RAM	Noise, Time, Location, communication
	Satisfaction	Age, Glasses, Competence	Screen size, RAM	Noise, Time, Location, communication
	Freedom from risks	–[f]	No dependency	No dependency
	Context coverage	No dependency	No dependency	Communication
Interestingness	Objective criteria	No dependency	No dependency	Communication
	Novelty	Experience[g], Age, Competence[h]	No dependency	Communication
	Surprisingness	Experience, Age, Competence	No dependency	Communication
	Comprehensibility	Experience, Age, Competence	No dependency	Communication

[a] The required task
[b] The test subject has visual impairments or not
[c] Knowledge level in the medical field: the job
[d] Random-access memory is a form of computer data storage.
[e] communication technology
[f] Only physicians were invited to answer this question
[g] Experience in using such prototype
[h] Knowledge level in the medical field: the job

Concerning the *Interestingness* quality factor, it is dependent only on the Communication technology. In fact, if there is no connection between the prototype and the server because of the unavailability of internet, the discovered rules cannot be displayed in the user interface. Therefore, users would not be able to answer the questionnaire (Q.interest). As a result, a dependency was raised between the communication technology and the Interstingness quality factor. However, no dependency was raised for the objective criteria pertaining to the interestingness quality factor [25].
All discovered dependencies are summarized in Table 5.6.

5.5.3 The obtained results: Prototype 3

Regarding the third prototype referring to the Knowledge management module, we have assessed the *classifier performance* which is the quality factor that evaluates the data stored in the knowledge base using equations presented in Chapter 3 (see Table 3.3). Before presenting results obtained by examining the knowledge base, it is worthy to mention that our study is based on two classes:

- Class A represents the *healthy* patients and,
- Class B represents the *affected* patients with the NI.

Hereafter, we present the classification of these classes:

- Class A= uncertain NI and improbable NI and impossible NI.
- Class B=Fifty-Fifty and expected NI and probable NI and sure NI.

The results are put in the confusion matrix presented in Table 5.7.

It is well known that we cannot extract conclusions from the results dealing with classifier performance. These results need to be compared with other generated results using a different algorithm of knowledge discovery.

Table 5.7: the confusion matrix

	Observed Classes	A	B
Predicted Classes	**A**	4	3
	B	1	15

Table 5.8: Classifier performance comparison

Criteria	Classifier performance in [137]	The current classifier performance
PPR	Ungiven	0.8
PPAR	0.71	0.83
NPAR	0.8	0.83
S	Ungiven	0.57
SP	Ungiven	0.94
GER	0.21	0.2

That is why we propose to compare the results generated by the current prototype with other results presented in [137].
We note that Ltifi *et al.* [137] have used *Dynamic Bayesian Networks* as algorithm for the same goal (predicting nosocamial infections). The results of comparison are given in Table 5.8 [29]. The results of comparison show that our evaluated prototype generates satisfactory prediction results. Thus, we come to conclusion that physicians can rely on this prototype for making appropriate decision(s). The obtained results are purely objective and depend only on the knowledge stored in the knowledge base. By changing the context of use, the results do not change. Consequently, there is no dependency between the classifier performance and the context of use (see Table 5.10).

The same conclusion for the *quality of promotion* can be drawn.Indeed, Table 5.9 presents satisfactory values when the internet connection is available. knowledge can be successfully be shared between physicians if an internet connection is available. Otherwise, knowledge cannot be shared between physicians.

Table 5.9: Quality of promotion results if the internet connection is available

Criteria	**Measures**	**The used technique**
Access	100	
Awareness	98.67	The questionnaire (Q.G)
Completeness	97.89	

We can deduce that the quality of promotion is only sensible to the availability of internet. However, the change of the user's profile or/and the platform does not affect the measures of this quality factor. So, there is dependency neither between the user's profile and the quality of promotion nor between the user's profile and the quality of promotion. The quality of promotion is only dependent on the availability of internet (see Table 5.10).

As regards the quality in use of this prototype, Figure 4.9 in Chapter 4 shows that effectiveness, efficiency and satisfaction are dependent on the user profile, particularly on the competence in the medical field. Moreover, our results have shown that the required task, the visual impairments, and the age influence the obtained measures (in terms of effectiveness, efficiency and satisfaction).

Table 5.10 summarizes all the discovered dependencies between the context features and the measured criteria [29].

5.5.4 The obtained results: Global evaluation

The evaluation presented in this research work is not limited to the evaluation of the criteria cited above. Unlike the previous works, we present a global evaluation of an MDSS/KDD that assesses a larger number of criteria using CEVASM. The questionnaire (Q.promo) was proposed only to physicians (P1) who participated to the evaluation of the 3 prototypes. Table 5.11 and Figure 4.10 which was presented in Chapter 4 shows the obtained global evaluation results of the whole MDSSM/KDD.

These results can be interpreted as follows. Despite the advantages of the

Table 5.10: The obtained results pertaining to the knowledge management prototype

Quality factor	**Criterion**	**User profile dependency**	**Platform dependency**	**Environment dependency**
Quality in use	All criteria	Same conclusions as those presented in Table 5.6	Same conclusions as those presented in Table 5.5	Same conclusions as those presented in Table 5.5
Classifier performance	All criteria	No dependency	No dependency	No dependency
Quality of promotion	Awareness	No dependency	No dependency	Communication technology, location
	Access	No dependency	No dependency	Communication technology, location
	Guidance	No dependency	No dependency	Communication technology, location
	Completeness	No dependency	No dependency	Communication technology, location

Table 5.11: Global evaluation results: results obtained from the evaluation tests

Criteria	Measures (%)	The used technique
Cost	100 (free)	
Response timeliness	96	
Intention to use	68	The questionnaire (Q.promo)
Priority	87	
Trust	65	

MDSS (being free, having a satisfactory response timeliness and considered as a priority by the ICU physicians), the physicians did neither trust it enough, nor express a real intention to use it.

5.5.5 Discussion about the obtained results

From the results presented previously, we can conclude that:

- There is a need to simplify the user interfaces of the whole MDSS. In fact, if the user test was in a noisy environment (i.e not fully concentrated with the prototype), the obtained results (particularly in terms of quality in use and interstingness) have presented their lowest levels.
- The quality in use of the second prototype referring to the data mining did not present satisfactory values. In fact, in addition to its design problems, users found that the developed user interface is not appropriate to their needs. In fact, there is no possibility to consult the previous discovered rules. Overall, there is a need to exploit human recognition capabilities to increase confidence and improve comprehensibility of the data.
- Users who have visual impairments found difficulties in using the prototypes. Labels of the user interfaces need to be larger to facilitate the use of the user interfaces.
- Internet is key factor for the use of the MDSSM. The prototypes cannot

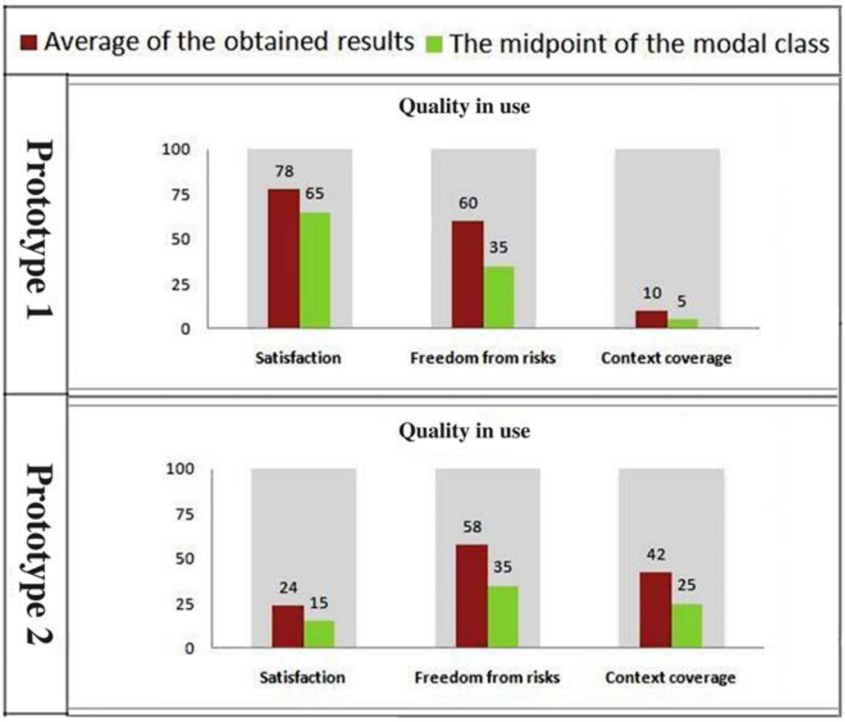

Figure 5.7: An example of the obtained results without using a context-based approach

be used if there is no connection between the MDSS and the server in which the repositories are stored.

It would be significant to mention that if we had not established a context-based approach, the results would be presented[8] as those in Figure 5.7. In this case, the evaluator(s) can retrieve ambiguous conclusions about the quality of the evaluated prototypes. For example, the first prototype, referring to the Data acquisition and storage module, presents high level of satisfaction. However, the second one, referring to the Data mining module, does not satisfy the end users. Here, the drawn conclusions are very general. Evaluator(s) need to have more precise knowledge about the quality of the prototypes. In fact, the first question that can be required is: *why did we get low level of satisfaction for the second prototype?* Our proposal aims to give, in some way, response to such kind of questions.

[8]These results are not visualized by CEVASM.

5.6 Discussion

In this section, we present a synthesis about the most important points drawn in this chapter. Afterwards, we discuss the possible bias and risks of validity that can affect our work.

5.6.1 Synthesis

In reality, MDSSs are used in different situations, environments, under different conditions and, by different users profiles. When the context of use changes because of one or several factors, the performance of the MDSS can also change; but there are some functional aspects that are independent of the context. For this reason, we consider that the criteria which are not context-aware can be sufficiently evaluated under only an instance of context of use. Nonetheless, the evaluation of context-aware criteria should be sensitive to the context changing. This variety is mainly due to three causes, namely the user, the technology and the environment.

We have carried out this research study in order to search for context-aware criteria that are mainly sensible to the features characterizing the context of use (the user profile, the platform and the environment). We specified the awareness feature of each criterion and proposed a set of complementary criteria that can contribute in the evaluation of MDSS/KDD process. As a final step of this study, it would be necessary to classify the criteria used for the evaluation of an MDSS/KDD. Most of these criteria are sensible to the context in which the prototype is used (see Table 5.12). Then, we can conclude that the quality factors which are purely related to the repositories (the database, the patterns repository, the knowledge base) are not sensible to the context of use.

Table 5.12: Classification of the criteria

Quality factor	**Criteria**	**Context awareness**[a]
Quality of data	All criteria	No
Quality in Use	All criteria	Yes
Interestingness	Objective criteria	No
	Subjective criteria	Yes
Classifier performance	All criteria	No
Quality of promotion	All criteria	Yes

[a]We consider that the internet is available

5.6.2 Discussion about the possible bias in the evaluation of the quality in use

In this study, the evaluation of the quality in use of all the KDD modules was clearly underlined; however, according to Albert and Tullis [11], seven bias sources can affect a usability study. These include the **participants** and the **environment** clearly evoked in our proposition. Albert and Tullis also argue that the nature of **the prototype** to be estimated has an enormous impact on the results. Our proposal is outside this area because our approach is dealing with a specific type of systems (MDSS/KDD). The **type of interaction** can also vary. In our case, we are concerned only with the interactions related to mobile devices. Four other concerns **the tasks**, **the evaluation methods**, **the moderators** and **the expectations**.

Indeed, the chosen tasks can have an impact on the identified problems. In our case study, we are based on well-defined task(s) for each prototype. They delineate a real and critical case study. Our main objective was to detect the main usability problems that can disturb the user during the execution of his/her tasks. It is noteworthy to mention that we know that the task can influence the results of the effectiveness and the efficiency, especially when the participants were involved in the evaluation. That is why we assumed this risk because we considered that it is important to achieve the evaluation because MDSS are nowadays highly interactive. As a result, its end users are involved in the evaluation process.

The type of the identified problems depends on the chosen evaluation methods. This bias was minimized by the use of various evaluation methods having two different aspects: i) objective and subjective ii) based on the user (decision-maker) and on the expert (evaluator).

Besides, to minimize the bias of the implication of the various **moderators** during the evaluation, we decided to imply a single moderator in all the sessions of evaluation. The moderator has made an impartial conduct in order to discard her influence on the participants during the evaluation. Moreover, since the measurement phase is established by the developed tool, rather than by the moderator (who is the expert in our case), we consider it as an acceptable bias.

Finally, concerning the **expectations** which concern the treated criteria, since we give the access to the evaluator to enter his/her expected measures, the corresponding measures are highly influenced by this aspect.

5.6.3 Risks of validity

Based on the statements of Wohlin *et al.* [220], There are four aspects of validity:

- the validity of construction which "focus[es] on the relation between the theory behind the experiment and the observation(s). Even if we have established that there is a casual relationship between the treatment of our experiment and the observed outcome, the treatment might not correspond to the cause we think we have controlled and altered. Similarly, the observed outcome might not correspond to the effect we think we are measuring."
- the internal validity which "focus[es]on how sure we can be that the treatment actually caused the outcome. There can be other factors that have caused the outcome, factors that we do not have control over or have not measured".

- the external validity is "concerned with whether we can generalize the results outside the scope of our study. Even if we have established a statistically significant casual relation between a treatment and an outcome and they correspond to the cause and effect we set out to investigate the results are of little use if the cause and effect we have established does not hold in other situations."
- "the validity of conclusion which focus on how sure we can be that the treatment we used in an experiment really is related to the actual outcome we observed. Typically this concerns if there is a statistically significant effect on the outcome."

In this subsection, we study the validity of our proposed approach. In doing so, we present not only the features that boost the validity of our work, but also those which may raise suspicions.

Concerning the **validity of construction**, we are aware that the immoderate use of the questionnaires affects the global reliability of the study. However, we considered that each question was defined with a specific purpose which we explored to measure a specific criterion that we did not find in the literature any objective way to measure it. Proposing some new formulas to such criteria needs rises a new research subject dealing with the objectivity and subjectivity of the techniques of measure. So, we are currently searching to contribute in this axis. More details about this point are given in the next section. From another side, it is worthy to mention that the use of the international standards (ISO 25010 and ISO 25020) can support the strength of our theoretical proposal. In fact, the criteria that we have used are recognized by the community. However, they can be insufficient especially when we are dealing with *mobile* systems. So, we have proposed to add new criteria (access, guidance, etc.) to be assessed in the future.
Otherwise, the model of context that we have proposed is based on the proposal of Sottet *et al.* [201]. We recall that these authors defined the context as a triplet (User, Platform, Environment). This choice was not arbitrary. In fact, the state of the art that we have established shows that several research works have been based on the same triplet. Although we tried to be general in our conception, our proposed model can be criticized. It can miss some

attributes or features if we are face to a highly specific system.

Concerning the **external validity**, although all the measures can be applicable in all domains, our proposed evaluation support system allows the evaluation of MDSS/KDD process and was specifically dedicated to the ICU of Habib Bourguiba hospital to evaluate the medical MDSS/KDD which is in turn dedicated to it. The generalization of the approach proposed in other domains needs the implementation of plug-ins [9]. These plug-ins must allow to the evaluated system to communicate with our proposed tool (CEVASM). Thus, our tool can access to the repository of data, patterns and knowledge and achieve the measurement of the corresponding criteria.
At this level, we should mention that CEVASM is able to assess MDSS/KDD that are dealing with classification problems; such as in the presented case, in which the patient is infected or not infected.

Concerning the **internal validity**, in our case, it is associated with the participants involved in the evaluation tests. The participants were selected by suitability. However, the evaluation could be biased by the users during the evaluation of the system because it is possible that some of them did not believe that it was a real case.

Finally, we present the risks of **validity of the conclusion** which are the ones which affect the capacity to draw good conclusions. It evaluates the relation between the treatment and the results of our study. To face this risk, we claimed that the criterion is sensible to the change of context only if the absolute value of the variation rate is $\succ 15\%$. If this condition is satisfied, then we proceed to the interpretation and the analysis.

5.7 Future research works

Among our perspectives of research, we aim to extend and enrich the proposed approach. In doing so, we set as starting point, the problems detected in the first part of this chapter.

[9] A plugin is a program conceived to add features to another software (called software host).

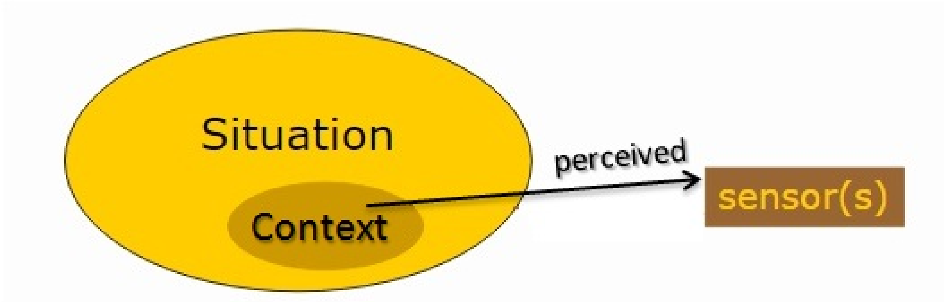

Figure 5.8: The relationship between the sensors and the context

5.7.1 The generalization of the proposed approach

To make possible a wide use of our proposed approach, it is essential to evolve and to generalize it. Given that in its current state, the approach is dedicated to MDSS/KDD used in case of prediction. However, our developed tool CEVASM can be used only by the MDSSM/KDD used in the ICU of Habib Bourguiba Hospital. Once we add the necessary plug-ins to the MDSS that we would like to evaluate, CEVASM is able to measure the criteria defined in our approach.

As future work, we intend to generalize the approach for the evaluation of various types of MDSS (not only used in prediction cases) and considering various contexts of use.
This work requires certain effort for the implementation of the appropriate evaluation criteria and the necessary plug-ins to be added to the MDSS/KDD process.

5.7.2 The context data acquisition

As described in Figure 5.8, sensors can be used to capture the context and construct high level context models [94]. Advances in sensor technologies suggest alternative approaches to real world context acquisition based on embedded or body-worn sensor infrastructures. TEA (Technology Enabling Awareness) is the project concerned with adding awareness of surrounding usage situations to personal mobile devices. The TEA system is based on a layered architecture, illustrated in Figure 5.9, which provides increasing levels of abstraction from sensors to the application:

1. The sensor layer is defined by an open array of sensors including both environmental sensors for the perception of the real world and logical sensors for monitoring (the platform and the MDSS, in our case).
2. A second layer abstracts information from individual sensors to a number of the so-called cues.
3. The third layer provides for multi-sensor fusion based on synthesis of the so-called contexts from cues.

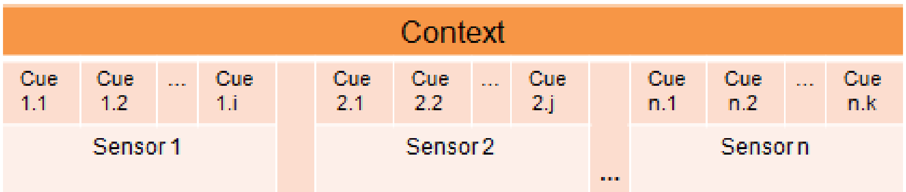

Figure 5.9: TEA architecture as presented in the literature

Thanks to such technology, it becomes possible to automatically, easily and rapidly collect much more context data that can help us to detect the defects of an MDSS. To this end, we need to follow the ensuing steps:

1. Specification: At this level, we need to specify the context model, so we determine the context features.
2. Acquisition: At this level, we need to install sensors (ie, the cues), determine the context representation and finally store the contextual data (eventually in a server).
3. Transport[10]: The collected data need to have available a network and a transport mechanism to send the data to the evaluation system.

[10]In computer networking, the transport layer is a conceptual division of methods in the layered architecture of protocols in the network stack in the Internet Protocol Suite and the Open Systems Interconnection (OSI). The protocols of the layer provide host-to-host communication services for applications.

4. Reception: assuming that the evaluation system can locate the context sensors, the latter send requests to the server, periodically, via queries. Then, the evaluation system would achieve additional processing such as aggregation, filtering, fusion, etc.

5. Treatment: The evaluation system should be able to combine received contexts with previous ones, and compare evaluation results with previous ones.

6. Analysis: In this last step, the evaluation system extracts useful results and produces an evaluation report.

5.7.3 Objective evaluation of the quality of data

The discovery of knowledge from data having poor quality (containing errors, doubles, incoherence, missing values, etc.) has direct consequences on the knowledge and then the user. For that purpose, treating the quality of the data is highly important at the first KDD module (Data acquisition and storage). However, we relied on the expert (evaluator) to evaluate this quality, which makes the evaluation purely subjective. This statement can be considered as bias. Moreover, this task is often considered by the experts as a heavy burden.

A possible solution consists in helping the KDD expert in assessing the quality of data criteria by calculating automatically measures of quality. These quality measures can also be conceived to combine two dimensions: an objective dimension and a subjective one.

We aim, in our future research works, to propose a methodology of evaluation of the quality of temporal data. To this end, we plan to achieve a study about the methods, the techniques of analysis and cleaning data. Such a study will allow us to understand, explore the data, detect and correct the quality problems of the data, and thusget better quality of the knowledge extracted from these data.

Itshould also be mentioned that the evaluation of this quality has to take into account the used technique of data mining. This proposal can

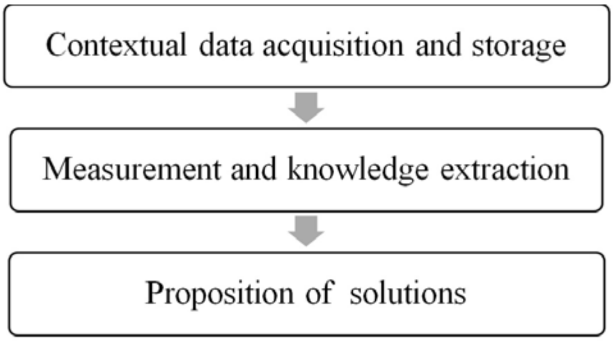

Figure 5.10: The extended evaluation process

be a progressive approach that goes from the evaluation of quality to the proposition of corrective actions. This project seems important for a good discovery of the knowledge.

In the next subsection, we present another perspective dealing with the proposition of corrective task.

5.7.4 Proposition of an intelligent evaluation support system for the evaluation of MDSS/KDD process

Although the proposed approach allows to identify the problems of the prototypes that constitute the MDSS/KDD process, it does not propose corrections if there are many problems to be solved (when several measures present low values). It is due to the fact that our presented approach deals only with the choice of criteria of each KDD module and propose a possibility to measure it.

To overcome this limit, we propose to extend our evaluation process to include two final steps: extract knowledge and propose possible solution(s) (see Figure 5.10). To achieve this goal, an intelligent evaluation is needed.

Intelligent evaluation is the one performed by an evaluation expert. In the intelligent evaluation systems, which we intend to propose, the expert knowledge is stored in a knowledge database using some representation, mostly by rules. The rules utilization control will be done by an artificial

intelligent method using an inference engine or a data mining technique. The ESS aims at assisting, rather than replacing, the evaluator by providing rational models to support his/her by reasoning abilities and extracting relevant patterns in vast volumes of overabundant information to help his/her to find a solution.
Given the unlimited number of possible situations that can face the evaluator, he/she, helped by his/her decision support artifacts, will develop scenarios, a small number compared to all those possible [173]. Hence, the need for an ESS. The use of scenarios appears to have been both the most common and the surest way to provide the appropriate decision [48]. So, as perspective we intend to design, implement and evaluate such intelligent evaluation system that would provide more significant support to the evaluator.

5.8 Conclusion

In this chapter, we put into practice our proposed approach for the evaluation of MDSS/KDD process. The implementation of this approach gave birth to the realized system CESEVASM. The illustration of the application of these proposed proposals is performed through a case study which concerns the evaluation of an MDSS/KDD used in the Medical field (MDSSM/KDD). Based on this use case, we were able to show the feasibility of our proposal. It was clearly shown that CEVASM gave support to the evaluators of MDSS/KDD to establish a global evaluation by taking into account our proposed context model.

Although the obtained results are promising, there are some limits that are discussed in this chapter. So, we have exposed our future works in the final part of this chapter. We aim in future research works to generalize the proposed approach to evaluate other types of MDSS and to automatically capture the context data using sensors. Moreover, we intend to perform an objective evaluation of the quality of data. Finally, we plan to propose an intelligent evaluation support system for the evaluation of MDSS/KDD process.

CONCLUSION

Our perfection consists in always striving forward,
in the endless inspiration for the better.
(Saint Bernard of Clairvaux)

The research work in this thesis pertains to the theme of interactive system evaluation, in particular mobile systems which support the Knowledge Discovery from Data process (KDD) [71]. In fact, since developers are trying to develop MDSS in several fields of application, their evaluation had become an essential task, as for many other interactive systems.

In the KDD, there are several important stages that allow the extraction of useful knowledge for making the appropriate decision(s). Evaluating systems based on KDD is presently framed in one stage in the KDD process [74] [71]. It is actually a centralized evaluation module, localized after the Data Mining module, which verifies whether the patterns generated from the Data Mining module are interesting.

Our established state of the art endorses that previous works upon the KDD process have underlined that each module in KDD should be assessed [32]. However, from our point of view, their proposal was incomplete as it did not define what we should evaluate and how to perform this evaluation. Moreover, these works remain limited to the DSS which are not mobile and still concentrate only on the evaluation of the data mining stage.

Therefore, it was important to propose an approach that allows an enhanced evaluation in the KDD process, taking into account the mobility aspect that characterizes a MDSS. The main objective is to help evaluators to detect defects as early as possible in order to enhance the quality of all the modules that constitute an MDSS/KDD.

In this thesis, we have proposed a novel approach which allowed us to achieve our main objective. In doing so, we contributed in the theoretical and applied levels.

- On the theoretical level:
  - We ensure a global evaluation of decision support systems. This permits an evaluation of the totality of KDD modules in order to foster better communications between the decision makers and the staff of any organization. Practically, this allows an assessment of the whole system that implements the KDD Process that forms together a decision support system based on the KDD process.
  - We append an evaluation support module for each module composing the KDD process based on quality models. The proposed evaluation support modules evaluate not only the quality in use of each module composing the KDD process, but also other criteria that reflect the objective(s) of each KDD module. This proposal allows a more enhanced evaluation of all the KDD process in order to evaluate all its modules (Data acquisition and storage, Data Mining, and Knowledge management).
  - We have presented a context based method that takes into account the change of context of use due to this mobility. Our proposition consists in achieving the evaluation test of the different KDD stages in different contexts of use. The different criteria

are measured in several contexts of use.

  - We have also conceived an Evaluation Support System that monitors and measures all criteria that were previously defined in our proposed quality models, in order to help the evaluator(s) in applying this approach.

- On the practical level:

  - We have proposed an evaluation support system that monitors and measures the proposed criteria. In doing so, we approached a possible implementation of our proposal. We presented all the realized developments to put into practice the proposed approach. These developments concern mainly the tool of evaluation called: Context-based EVAluation support System for Mobile decision support systems based on KDD process (CEVASM). It contributes to the existing tools by offering not only remote support but also detailed and summarized synthesis of the obtained measures of evaluation. In fact, contrary to the majority of the existing approaches, we suggested in our approach to supplying a database that supports the evaluators to interpret the obtained results.

  - Theapproachwe propose is applied for the evaluation of the modules of an MDSS/KDD for the fight against nosocomial infections, representing one of the major problems in the intensive care unit of the hospital Habib Bourguiba in Sfax, Tunisia. For every KDD module, we are interested in the phases of evaluation. We follow the evaluation process, defined in Chapter 4 and based on the standard ISO/IEC 25040. The objective is to be able to validate, *a priori*, the realized evaluation tool (CEVASM) and consequently, the proposed approach.

As a critical study, we presented a discussion which concerns the risks of validity of our led work. Therefore, we distinguished some weaknesses, cited below, to be considered during the use of our approach:

- The use of our approach requires the source code of the system to be evaluated for the capture of the required data. Nevertheless, the source code is not always available.

- During the use of CEVASM, the implication of an expert of evaluation is required. Moreover, several evaluation tests need to be achieved to evaluate the whole MDSS/KDD. This can engender high costs of evaluation.

- Although our approach allows to identify the MDSS/KDD problems (defects), it does not propose a way of prioritization of the efforts of correction in case there are many problems to be approached.

- The evaluation of the *quality of data* is, until now, purely subjective. We rely on the evaluator to answer a questionnaire. This statement can be considered as adrawback. Moreover, this task is often considered by the experts as a heavy burden.

- CEVASM is not able to propose recommendations. In fact, except the approach of Assila et al. [16] which proposed a set of recommendations to help the evaluator correct the detected problems and improve the quality of the user interfaces user of the estimated system, the other approaches, including ours supplied nothing.

- It is possible that some users who participated to the evaluation tests were not serious enough in their responses to the questionnaires or in their achievements of the required tasks. So, the evaluation could be biased by these users.

On the basis of these limits, we introduce afterward, the main perspectives of our research, which concern our evaluation support system components presented in Figure 5.11. As illustrated in this Figure, the input level includes the MDSS/KDD to be evaluated, the contextual data and the interactive data which describe the manner in which the MDSS/KDD user interfaces is used. At the input level, we propose the following perspectives :

- we intend to generalize the approach for the evaluation of various types

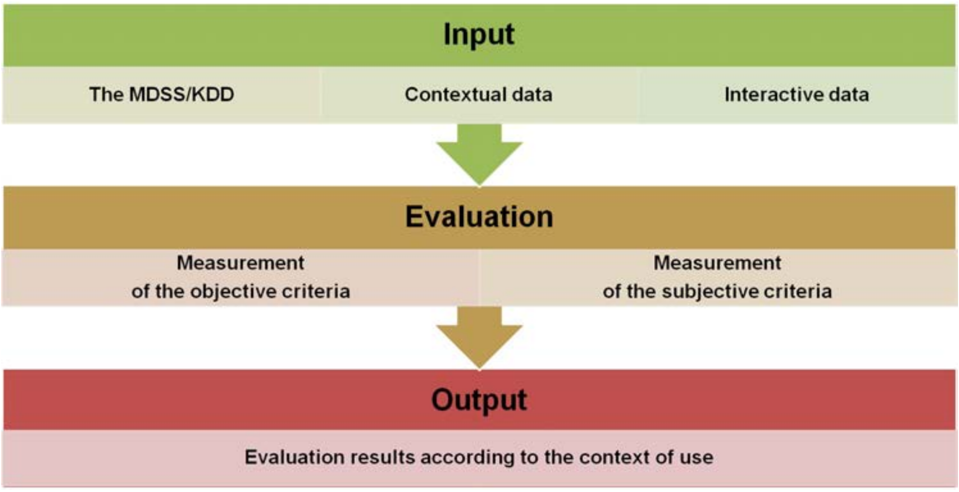

Figure 5.11: The evaluation support system components

of MDSS (used not only in prediction cases) while considering various contexts of use.

- In order to improve the contextual data acquisition and facilitate the task of the evaluator, it would be useful to use sensors that capture the contextual data automatically, easily and rapidly. Thanks to the-sensors, we can collect much more context data that can help us to detect the defects of an MDSS.

According to the methodology of evaluation:

- We aim to propose a methodology of *quality of data* evaluation. The evaluation of this quality factor needs to be achieved automatically and has to take into account the used technique of data mining.

- We intend to propose an intelligent evaluation system. Thanks to such system, the expert knowledge can be stored in a knowledge base using some representation. Using an inference engine or a data mining technique, the intelligent evaluation system would be able to assist the evaluator in making the appropriate decisions regarding the evaluated MDSS/KDD.

When the inputs and the evaluation method are improved, we expect more valuable quality of the obtained results. However, it would be significant to append a recommendation module to our proposed evaluation system which can propose recommendations to its end users (the evaluators of MDSS).

Bibliography

[1] ISO/IEC 25012, Software engineering – software product quality requirements and evaluation (SQuaRE) – data quality model url = http://www.iso.org/iso/catalogue$_d etail.htm?csnumber = 35736.Accessed18thDecember2016.year = 2008.Technicalreport.$

[2] ISO/IEC 9126: 1991. Technical report, 1991.

[3] G. D. Abowd, J. Coutaz, and L. Nigay. Structuring the space of interactive system properties. *Engineering for Human-Computer Interaction*, 18:113–129, 1992.

[4] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, HUC '99, pages 304–307, London, UK, UK, 1999. Springer-Verlag.

[5] G. Adamson. One hundred reasons socially beneficial technology might not work. In *IEEE International Symposium on Technology and Society (ISTAS)*, pages 384–398. IEEE, 2010.

[6] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules

between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, 1993.

[7] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. *Proc. 20th int. conf. very large data bases, VLDB*, 1215:487–499, 1994.

[8] Q. N. Ahmed and S. V. Vrbsky. Maintaining security and timeliness in real-time database system. *Journal of Systems and software*, 61(1):15–29, 2002.

[9] A. Akharraz, J. Montmain, and G. Mauris. A cybernetic modeling of a multi-criteria decision process in organization. In *EURO XX, 20th European Conference on Operational Research*, Island of Rhodes, Greece, 2004.

[10] A. S. Al-Hegami, V. Bhatnagar, and N. Kumar. Novelty framework for knowledge discovery in databases. In Y. Kambayashi, M. Mohania, and W. Wöß, editors, *International Conference on Data Warehousing and Knowledge Discovery*, pages 48–57. Springer, Berlin, Heidelberg, 2004.

[11] W. Albert and T. Tullis. *Measuring the user experience: collecting, analyzing, and presenting usability metrics.* Newnes, 2013.

[12] F. Antunes, M. Freire, and J. P. Costa. Semantic web and decision support systems. *Journal of Decision Systems*, 25(1):79–93, 2016.

[13] D. Arnott. *Personal Decision Support Systems*, pages 127–150. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[14] D. Arnott, G. Pervan, and G. Pervan. A critical analysis of decision support systems research. *Journal of Information Technology*, 20:67–87, 2005.

[15] A. Assila, H. Ezzedine, and M. Bouhlel. A web questionnaire generating tool to aid for interactive systems quality subjective assessment. In *Control, Decision and Information Technologies (CoDIT), 2013 International Conference on*, pages 815–821, May 2013.

[16] A. Assila, K. Oliveira, and H. Ezzedine. Standardized usability questionnaires : Features and quality focus. *Electronic Journal of Computer Science and Information Technology*, 6(1):15–31, 2016.

[17] M. Baena-García and R. Morales-Bueno. Mining interestingness measures for string pattern mining. *Knowledge-Based Systems*, 25(1):45–50, 2012.

[18] S. Balbo. *Evaluation ergonomique des interfaces utilisateur: un pas vers l'automatisation.* PhD Thesis, Grenoble, France, 1994.

[19] L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci. Context relevance assessment and exploitation in mobile recommender systems. *Personal and Ubiquitous Computing*, 16(5):507–526, 2012.

[20] M. Barcelo-Valenzuela, P. S. Carrillo-Villafaña, A. Perez-Soltero, and G. Sanchez-Schmitz. A framework to acquire explicit knowledge stored on different versions of software. *Information and Software Technology*, 70:40–48, 2016.

[21] J. Bastien and D. Scapin. Évaluation des systèmes d'information et critères ergonomiques. *In Kolski C. (Ed.), Environnements évolués et évaluation de l'IHM, Interaction homme-machine pour les SI (Hermes)*, 2:53–80, 2001.

[22] K. Beck. Embracing change with extreme programming. *Computer*, 32(10):70–77, Oct. 1999.

[23] B. Behkamal, M. Kahani, and M. K. Akbari. Customizing iso 9126 quality model for evaluation of B2B applications. *Information and software technology*, 51(3):599–609, 2009.

[24] E. Ben Ayed and M. Ben Ayed. Towards a quality model for the evaluation of DSS based on KDD process. In *2013 International Conference on Advanced Logistics and Transport, ICALT 2013*, pages 187–192, 2013.

[25] E. Ben Ayed and M. Ben Ayed. A quality model for the evaluation of decision support systems based on a knowledge discovery from data process. *Journal of Decision Systems*, 25(02):95–117, 2016.

[26] E. Ben Ayed, M. Ben Ayed, C. Kolski, and H. Ezzedine. Towards a quality model for the evaluation of mobile decision support systems. In *4th International Conference on Advanced Logistics and Transport (ICALT)*, pages 187–192. IEEE, 2015.

[27] E. Ben Ayed, M. Ben Ayed, C. Kolski, H. Ezzedine, and F. Gargouri. Evaluation of mobile decision support systems: preliminary basic concepts. In *Workshop international: Interaction Homme-Machine et Imagerie (IHMIM'14)*, 2014.

[28] E. Ben Ayed, M. Ben Ayed, C. Kolski, H. Ezzedine, and F. Gargouri. Context aware criteria for the evaluation of mobile decision support systems. In *2015 IEEE/ACIS 14th International Conference on Computer and Information Science (ICIS)*, pages 1–6, 2015.

[29] E. Ben Ayed, C. Kolski, and H. Ezzedine. Context-based evaluation of mobile knowledge management systems. In *13th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA), Agadir, Morocco*. IEEE, 2016.

[30] E. Ben Ayed, R. Magdich, C. Kolski, and H. Ezzedine. Towards a context based evaluation support system for quality in use assessment of mobile systems. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC) Budapest, Hungary, ISBN 978-1-5090-1819-2*, pages 4350–4355. IEEE, 2016.

[31] M. Ben Ayed, M. A. Lajnef, H. Kallel, and A. M. Alimi. Towards a conception of data bases and application of data mining techniques for decision making in hospital infection control project. In *International symposium on Computational Intelligence and Intelligent Informatics, ISCIII 2005*, pages 168–174, 2005.

[32] M. Ben Ayed, H. Ltifi, C. Kolski, and A. M. Alimi. A user-centered approach for the design and implementation of KDD-based DSS: A case study in the healthcare domain. *Decision Support Systems*, 50(1):64–78, 2010.

[33] H. Benbya, G. Passiante, and N. A. Belbaly. Corporate portal: A tool

for knowledge management synchronization. *International Journal of Information Management*, 24(3):201–220, 2004.

[34] V. Benjamins. Information is not knowledge, knowledge is not wisdom, wisdom is not truth. *International Journal of Human-Computer Studies*, 71(2):166 – 170, 2013. 25 Years of Knowledge Acquisition.

[35] N. O. Bernsen and L. Dybkjær. Multimodal usability: Conclusions and future work. In *Multimodal Usability*, pages 413–419. Springer, London, 2010.

[36] N. Bevan. International standards for hci and usability. *International Journal of Human-Computer Studies*, 55(4):533–552, 2001.

[37] N. Bevan, J. Carter, and S. Harker. Iso 9241-11 revised: What have we learnt about usability since 1998? In M. Kurosu, editor, *Human-Computer Interaction: Design and Evaluation: 17th International Conference, HCI International 2015, Los Angeles, CA, USA, August 2-7, 2015, Proceedings, Part I*, pages 143–151. Springer International Publishing, Cham, 2015.

[38] B. Biel, T. Grill, and V. Gruhn. Exploring the benefits of the combination of a software architecture analysis and a usability evaluation of a mobile application. *Journal of Systems and Software*, 83(11):2031–2044, 2010.

[39] A. Blandford, S. Keith, I. Connell, and H. Edwards. Analytical usability evaluation for digital libraries: a case study. In *Digital Libraries, 2004. Proceedings of the 2004 Joint ACM/IEEE Conference on*, pages 27–36. IEEE, 2004.

[40] B. W. Boehm. A spiral model of software development and enhancement. *Computer*, 21(5):61–72, 1988.

[41] B. W. Boehm, J. R. Brown, and H. Kaspar. *Characteristics of software quality*. TRW Series of Software Technology. North-Holland, Amsterdam, 1978.

[42] A. Bourouis, M. Feham, M. Hossain, and L. Zhang. An intelligent mobile based decision support system for retinal disease diagnosis. *Decision Support Systems*, 59:341 – 350, 2014.

[43] P. J. Brown. The stick-e document: a framework for creating context-aware applications. In *Proceedings of EP"96, Palo Alto*, pages 259–272. also published in it EP–odd, January 1996.

[44] P. J. Brown, J. D. Bovey, and X. Chen. Context-aware applications: from the laboratory to the marketplace. *IEEE personal communications*, 4(5):58–64, 1997.

[45] M. N. Burns, M. Begale, J. Duffecy, D. Gergle, C. J. Karr, E. Giangrande, and D. C. Mohr. Harnessing context sensing to develop a mobile intervention for depression. *Journal of medical Internet research*, 13(3):e55, 2011.

[46] R. Buyya, S. Pandey, and C. Vecchiola. *Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings*, chapter Cloudbus Toolkit for Market-Oriented Cloud Computing, pages 24–44. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[47] L. C.-C. An empirical study of mobile commerce in insurance industry: Task-technology fit and individual differences. *Decision Support Systems*, 43:95, 2007.

[48] G. Camilleri, J.-L. Soubie, and P. Zarate. *A Replanning Support for Critical Decision Making Situations: A Modelling Approach*, pages 173–192. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[49] D. R. Carvalho, A. A. Freitas, and N. Ebecken. Evaluating the correlation between objective rule interestingness measures and real human interest. In A. M. Jorge, L. Torgo, P. Brazdil, R. Camacho, and J. Gama, editors, *Knowledge Discovery in Databases: 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 453–461. Springer, Berlin, Heidelberg, 2005.

[50] R. S. Castillo and A. Kelemen. Considerations for a successful clinical decision support system. *Computers, informatics, nursing : CIN*, 31(7):319–26; quiz 327–8, 2013.

[51] S. Charfi, H. Ezzedine, C. Kolski, and F. Moussa. Towards an automatic analysis of interaction data for HCI evaluation application to a transport network supervision system. In *Human-Computer Interaction. Design and Development Approaches*, pages 175–184. Springer, 2011.

[52] S. Chatterjee, S. Chakraborty, S. Sarker, S. Sarker, and F. Y. Lau. Examining the success factors for mobile work in healthcare: a deductive study. *Decision Support Systems*, 46(3):620–633, 2009.

[53] G. Chen and D. Kotz. A survey of context-aware mobile computing research. Technical report, Hanover, NH, USA, 2000.

[54] L. Chen and P. S. Fong. Evaluation of knowledge management performance: An organic approach. *Information and Management*, 52(4):431 – 453, 2015.

[55] M. Chen, Y. Liou, C.-W. Wang, Y.-W. Fan, and Y.-P. J. Chi. Teamspirit: Design, implementation, and evaluation of a web-based group decision support system. *Decision Support Systems*, 43(4):1186–1202, 2007.

[56] R. E. Clark, D. Feldon, J. J. van Merriënboer, K. Yates, and S. Early. Cognitive task analysis. In *Handbook of research on educational communications and technology*, volume 3, pages 577–593. Lawrence Erlbaum Associates, Mahwah, NJ, 2008.

[57] T. Comes, M. Hiete, N. Wijngaards, and F. Schultmann. Decision maps: A framework for multi-criteria decision support under severe uncertainty. *Decision Support Systems*, 52(1):108–118, 2011.

[58] G. E. Company, J. A. McCall, P. K. Richards, and G. F. Walters. *Factors in software quality: Final report.* Information Systems Programs, General Electric Company, 1977.

[59] M.-A. Côté, W. Suryn, and E. Georgiadou. In search for a widely applicable and accepted software quality model for software quality engineering. *Software Quality Journal*, 15(4):401–416, 2007.

[60] H. A. de Wit, C. M. Gonzalvo, J. Cardenas, H. J. Derijks, R. Janknegt, P.-H. M. van der Kuy, B. Winkens, and J. M. Schols. Evaluation of clinical rules in a standalone pharmacy based clinical decision support system for hospitalized and nursing home patients. *International Journal of Medical Informatics*, 84(6):396–405, 2015.

[61] A. K. Dey. Understanding and using context. *Personal Ubiquitous Comput.*, 5(1):4–7, 2001.

[62] A. K. Dey and J. Mankoff. Designing mediation for context-aware applications. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12(1):53–80, 2005.

[63] R. Dhar. The Effect of Decision Strategy on Deciding to Defer Choice. *Journal of Behavioral Decision Making*, 9(January):265–281, 1996.

[64] A. Dix. *Human-computer interaction.* Springer, 2009.

[65] A. N. Ella, C. Kolski, F. Wawak, C. Jacques, and P. Yim. An approach of computer-aided choice of ui evaluation criteria and methods. In *Proceedings of the Third International Conference on Computer-aided Design of User Interfaces*, pages 319–329. Kluwer Academic Publishers, 1999.

[66] J. P. Espada, V. G. Díaz, R. G. Crespo, O. S. Martínez, B. P. G-Bustelo, and J. M. C. Lovelle. Using extended web technologies to develop bluetooth multi-platform mobile applications for interact with smart things. *Information Fusion*, 21:30 – 41, 2015.

[67] H. Ezzedine and M. Abed. Une méthode d'évaluation d'interface homme machine de supervision d'un procédé industriel. *Revue JESA (RAIRO-APII-JESA)*, 31(7):1078–1110, 1997.

[68] H. Ezzedine and A. Trabelsi. From the design to the evaluation of an agent-based human-machine interface. application to supervision for

urban transport system. In *IMACS World Congress" Scientific Computation, Applied Mathematics and Simulation"(July 11-15, Paris), ECL*, pages 717–725, 2005.

[69] S. Fahmy, N. Haslinda, W. Roslina, and Z. Fariha. Evaluating the quality of software in e-book using the iso 9126 model. *International Journal of Control and Automation*, 5(2):115–122, 2012.

[70] W. Fan and H. Perros. A novel trust management framework for multi-cloud environments based on trust service providers. *Knowledge-Based Systems*, 70:392 – 406, 2014.

[71] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–53, 1996.

[72] U. Fayyad and P. Stolorz. Data mining and KDD: Promise and challenges. *Future Generation Computer Systems*, 13(2–3):99 – 115, 1997. Data Mining.

[73] U. M. Fayyad. Data mining and knowledge discovery in databases: Applications in astronomy and planetary science. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 2*, AAAI'96, pages 1590–1592. AAAI Press, 1996.

[74] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus. Knowledge discovery in databases: An overview. *AI magazine*, 13(3):57, 1992.

[75] S. Gafsi-Moalla. *Les infections acquises en réanimation, étude prospective réalisée dans le service de réanimation de Sfax sur une période de 3 mois.* PhD thesis, Faculty of Medicine of Sfax, Tunisia, 2005.

[76] V. Ganneau, G. Calvary, and R. Demumieux. Métamodèle de règles d'adaptation pour la plasticité des interfaces homme-machine. In *Proceedings of the 19th Conference on L'Interaction Homme-Machine*, IHM '07, pages 91–98, New York, NY, USA, 2007. ACM.

[77] S. Gao. Mobile decision support systems research: a literature analysis. *Journal of Decision Systems*, 22(1):10–27, 2013.

[78] J. S. Garner, W. R. Jarvis, T. G. Emori, T. C. Horan, and J. M. Hughes. Cdc definitions for nosocomial infections, 1988. *American journal of infection control*, 16(3):128–140, 1988.

[79] L. Geng and H. J. Hamilton. Interestingness measures for data mining. *ACM Computing Surveys*, 38(3):1–32, 2006.

[80] R. Gregory, L. Failing, M. Harstone, G. Long, T. McDaniels, and D. Ohlson. *Structured decision making: a practical guide to environmental management choices.* John Wiley & Sons, 2012.

[81] B. Habert. Compte rendu de daniel kayser, la représentation des connaissances, paris, hermès, 1998. *TAL*, 40(1):168–170, 1999.

[82] J. Han, M. Kamber, and J. Pei. Classification: Basic concepts. In J. H. Kamber and J. Pei, editors, *Data Mining (Third Edition)*, The Morgan Kaufmann Series in Data Management Systems, pages 327 – 391. Morgan Kaufmann, Boston, 2012.

[83] H. R. Hartson, T. S. Andre, and R. C. Williges. Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction*, 15(1):145–181, 2003.

[84] S. Harun, M. Hamid, A. Talib, and Z. Rahim. "usability evaluation": Criteria for quality architecture in-use. *Procedia Engineering*, 20:135 – 146, 2011.

[85] L. Hergafi. *Présentation et validation d'un nouveau système pour la surveillance de l'infection acquise en réanimation.* PhD thesis, PhD Thesis, Faculty of Medicine of Sfax, Tunisia, 2006.

[86] D. M. Hilbert and D. F. Redmiles. Extracting usability information from user interface events. *ACM Computing Surveys (CSUR)*, 32(4):384–421, 2000.

[87] D. Hix and H. R. Hartson. *Developing user interfaces: ensuring usability through product & process.* John Wiley & Sons, Inc., 1993.

[88] A. Holzinger. Rapid prototyping for a virtual medical campus interface. *IEEE software*, 21(1):92–99, 2004.

[89] A. Holzinger. *Availability, Reliability, and Security in Information Systems and HCI: IFIP WG 8.4, 8.9, TC 5 International Cross-Domain Conference, CD-ARES 2013, Regensburg, Germany, September 2-6, 2013. Proceedings*, chapter Human-Computer Interaction and Knowledge Discovery (HCI-KDD): What Is the Benefit of Bringing Those Two Fields to Work Together?, pages 319–328. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[90] A. Holzinger. Human-computer interaction and knowledge discovery (HCI-KDD): What is the benefit of bringing those two fields to work together? In *Availability, Reliability, and security in Information Systems and HCI*, pages 319–328. Springer, 2013.

[91] A. Holzinger. Trends in interactive knowledge discovery for personalized medicine: Cognitive science meets machine learning. *IEEE Intelligent Informatics Bulletin*, 15(01):6–14, 2014.

[92] J. I. Hong, J. Heer, S. Waterson, and J. A. Landay. Webquilt: A proxy-based approach to remote web usability testing. *ACM Transactions on Information Systems*, 19(3):263–285, 2001.

[93] B. Hosack, D. Hall, D. Paradice, and J. F. Courtney. A look toward the future: Decision support systems research is alive and well. *Journal of the Association for Information Systems*, 13(5):315–340, 2012.

[94] M. Hussain, M. F. Shafeeq, S. Jabbar, A. H. Akbar, and S. Khalid. Cram: A conditioned reflex action inspired adaptive model for context addition in wireless sensor networks. *Journal of Sensors*, vol. 2016, Article ID 6319830, 24 pages, 2016.

[95] International Organization For Standardization Iso. ISO/IEC 25010: 2011. Technical report, 2011.

[96] I. ISO. Iec 25000 software and system engineering–software product quality requirements and evaluation (square)–guide to square. *International Organization for Standarization*, 2005.

[97] S. V. Ivanov and K. V. Knyazkov. Evaluation of in-vehicle decision support system for emergency evacuation. *Procedia Computer Science*, 29:1656–1666, 2014.

[98] M. Y. Ivory and M. A. Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys (CSUR)*, 33(4):470–516, 2001.

[99] C. M. Jackson, S. Chow, and R. A. Leitch. Toward an understanding of the behavioral intention to use an information system. *Decision sciences*, 28(2):357–389, 1997.

[100] I. Jacobson, G. Booch, J. Rumbaugh, J. Rumbaugh, and G. Booch. *The unified software development process*, volume 1. Addison-wesley Reading, 1999.

[101] M. W. Jaspers. A comparison of usability methods for testing interactive health technologies: methodological aspects and empirical evidence. *International journal of medical informatics*, 78(5):340–353, 2009.

[102] R. Jeffries, J. R. Miller, C. Wharton, and K. Uyeda. User interface evaluation in the real world: a comparison of four techniques. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 119–124. ACM, 1991.

[103] J.-H. Jhang-Li and I. R. Chiang. Resource allocation and revenue optimization for cloud service providers. *Decision Support Systems*, 77:55 – 66, 2015.

[104] S. Jiang, L. Fang, and X. Huang. *Cloud Computing: First International Conference, CloudCom 2009, Beijing, China, December 1-4, 2009. Proceedings*, chapter An Idea of Special Cloud Computing in Forest Pests' Control, pages 615–620. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[105] G. H. J. Johannessen and K. Hornbæk. Must evaluation methods be about usability? devising and assessing the utility inspection method. *Behaviour & Information Technology*, 33(2):195–206, 2014.

[106] B. E. John and D. E. Kieras. The goms family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 3(4):320–351, 1996.

[107] I. Junglas, C. Abraham, and R. T. Watson. Task-technology fit for mobile locatable information systems. *Decis. Support Syst.*, 45(4):1046–1057, 2008.

[108] S. K. Mobile commerce: Promises, challenges and research agenda. *Journal of Database Management*, 12:4, 2001.

[109] M. Kakihara and C. Sørensen. Expanding the'mobility'concept. *ACM SIGGroup bulletin*, 22(3):33–37, 2001.

[110] A. Kaklauskas. *Intelligent Decision Support Systems*, pages 31–85. Springer International Publishing, Cham, 2015.

[111] H. Kallel, M. Bahoul, H. Ksibi, H. Dammak, H. Chelly, C. B. Hamida, A. Chaari, N. Rekik, and M. Bouaziz. Prevalence of hospital-acquired infection in a tunisian hospital. *Journal of Hospital Infection*, 59(4):343–347, 2005.

[112] S. H. Kan. *Metrics and models in software quality engineering.* Addison-Wesley Longman Publishing Co., Inc., 2002.

[113] R. S. Kaplan and R. Cooper. *Cost & effect: using integrated cost systems to drive profitability and performance.* Harvard Business Press, 1998.

[114] D. J. Kasik and H. G. George. Toward automatic generation of novice user test scripts. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 244–251. ACM, 1996.

[115] D. Kelly. Methods for evaluating interactive information retrieval systems with users. *Foundations and Trends in Information Retrieval*, 3(1—2):1–224, 2009.

[116] J. Kjeldskov and C. Graham. A review of mobile hci research methods. In *International Conference on Mobile Human-Computer Interaction*, pages 317–335. Springer, 2003.

[117] J. Kjeldskov, M. B. Skov, B. S. Als, and R. T. Høegh. Is it worth the hassle? exploring the added value of evaluating the usability of context-aware mobile systems in the field. In *International Conference on Mobile Human-Computer Interaction*, pages 61–73. Springer, 2004.

[118] J. Kjeldskov, M. B. Skov, and J. Stage. Instant data analysis: conducting usability evaluations in a day. In *Proceedings of the third Nordic conference on Human-computer interaction*, pages 233–240. ACM, 2004.

[119] C. Kolski. *Interfaces Homme-machine: application aux systèmes industriels complexes*. Hermès, 1997.

[120] C. Kolski. A "call for answers" around the proposition of an hci-enriched model. *ACM SIGSOFT Software Engineering Notes*, 23(3):93–96, 1998.

[121] K. Kosanke. *Interoperability of Enterprise Software and Applications*, chapter ISO Standards for Interoperability: a Comparison, pages 55–64. Springer London, London, 2006.

[122] L. a. Kurgan and P. Musilek. A survey of Knowledge Discovery and Data Mining process models. *The Knowledge Engineering Review*, 21(01):1–24, 2006.

[123] M. Kurosu. *Usability, Quality in Use and the Model of Quality Characteristics*, pages 227–237. Springer International Publishing, Cham, 2015.

[124] S. H. Kwok and S. Gao. Attitude towards knowledge sharing behavior. *Journal of Computer Information Systems*, 46(2):45–51, 2005.

[125] M. A. Lajnef, M. B. Ayed, and C. Kolski. Convergence possible des processus du data mining et de conception-évaluation d'ihm: adaptation du modèle en u. In *Proceedings of the 17th Conference on l'Interaction Homme-Machine*, pages 243–246. ACM Press, Toulouse, 2005.

[126] S. Lepreux, M. Abed, and C. Kolski. A human-centred methodology applied to decision support system design and evaluation in a railway network context. *Cognition, Technology & Work*, 5(4):248–271, 2003.

[127] S. Lepreux, C. Kolski, and G. Queric. *Towards a Methodology for DSS User-Centered Design*. 2003.

[128] P. Lévine and J. Pomerol. *Systèmes interactifs d'aide à la décision et systèmes experts*. Hermes, 1989.

[129] J. R. Lewis. IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction*, 7(1):57–78, 1995.

[130] J. R. Lewis. Usability: lessons learned... and yet to be learned. *International Journal of Human-Computer Interaction*, 30(9):663–684, 2014.

[131] J. R. Lewis and M. L. Hardzinski. Investigating the psychometric properties of the speech user interface service quality questionnaire. *International Journal of Speech Technology*, 18(3):479–487, 2015.

[132] K. Lin, J. Xu, I. M. Baytas, S. Ji, and J. Zhou. Multi-task feature interaction learning. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1735–1744, 2016.

[133] Z. Liu, Q. Min, and S. Ji. An empirical study of mobile securities management systems adoption: a task technology fit perspective. *Int. J. Mob. Commun.*, 8(2):230–243, Feb. 2010.

[134] H. Ltifi, M. B. Ayed, C. Kolski, and A. M. Alimi. HCI-enriched approach for DSS development: the UP/U approach. In *IEEE Symposium on Computers and Communications*, pages 895–900. IEEE, 2009.

[135] H. Ltifi, M. Ben Ayed, C. Kolski, and A. M. Alimi. Prise en compte de l'utilisateur pour la conception d'un SIAD basé sur un processus d'ECD. *ERGO'IA*, pages 85–92, 2008.

[136] H. Ltifi, C. Kolski, M. B. Ayed, and A. M. Alimi. A human-centred design approach for developing dynamic decision support system based on knowledge discovery in databases. *Journal of Decision Systems*, 22(2):69–96, 2013.

[137] H. Ltifi, C. Kolski, and M. Ben Ayed. Combination of cognitive and HCI modeling for the design of KDD-based DSS used in dynamic situations. *Decision Support Systems*, 78:51–64, 2015.

[138] I. Lungu and A. Bara. Executive information systems. *Academy of Economic Studies Publisher*, 2007.

[139] M. Maguire. Methods to support human-centred design. *International journal of human-computer studies*, 55(4):587–634, 2001.

[140] B. Malakooti. Decision making process: Typology, intelligence, and optimization. *Journal of Intelligent Manufacturing*, 23(3):733–746, 2012.

[141] N. Mallat, M. Rossi, V. K. Tuunainen, and A. Öörni. The impact of use context on mobile services acceptance: The case of mobile ticketing. *Information Management*, 46(3):190 – 195, 2009.

[142] C. Mariage et al. *MetroWeb: logiciel de support à l'évaluation de la qualité ergonomique des sites web.* PhD thesis, UCL, Belgium, 2005.

[143] G. Mariscal, Ó. Marbán, and C. Fernández. A survey of data mining and knowledge discovery process models and methodologies. *The Knowledge Engineering Review*, 25(02):137–166, 2010.

[144] J. Martin. *Rapid application development*, volume 8. Macmillan New York, 1991.

[145] D. E. L. Maryam Alavi. Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues. *MIS Quarterly*, 25(1):107–136, 2001.

[146] J. McDermid and K. Ripken. *Life cycle support in the ADA environment.* CUP Archive, 1984.

[147] S. J. Miah. Cloud-based intelligent dss design for emergency professionals. In S. Ali, N. Abbadeni, and M. Batouche, editors, *Multidisciplinary Computational Intelligence Techniques : Applications in Business, Engineering, and Medicine*, pages 47–60. IGI Global, Hershey, Pennsylvania, 2012.

[148] A. B. Miled, D. Monticolo, V. Hilaire, and A. Koukam. A Comparison of Knowledge Management Approaches Based on Multi Agent Systems. In *5th International Conference on Signal Image Technology and Internet Based Systems*, pages 479–484, 2009.

[149] D. Milicic. Software quality models and philosophies. In C. W. Lars Lundberg, Michael Mattsson, editor, *Software quality attributes and trade-offs*, pages 3–19. Blekinge Institute of Technology, 2005.

[150] I. I. Mitroff, R. O. Mason, and V. P. Barabba. Policy as argument—a logic for ill-structured decision problems. *Management Science*, 28(12):1391–1404, 1982.

[151] H. Moshkovich, A. Mechitov, and D. Olson. Verbal decision analysis. In *Multiple Criteria Decision Analysis*, pages 605–636. Springer, 2016.

[152] G. W. Musumba and H. O. Nyongesa. Context awareness in mobile computing: A review. *International Journal of Machine Learning and Applications*, 2(1):5–pages, 2013.

[153] B. A. Myers. User interface software tools. *ACM Trans. Comput.-Hum. Interact.*, 2(1):64–103, Mar. 1995.

[154] M. Naderpour, J. Lu, and G. Zhang. A safety-critical decision support system evaluation using situation awareness and workload measures. *Reliability Engineering System Safety*, 150:147–159, 2016.

[155] A. S. Neal and R. M. Simons. Playback: A method for evaluating the usability of software and its documentation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 78–82. ACM, 1983.

[156] E. Ngai and E. Chan. Evaluation of knowledge management tools using ahp. *Expert Systems with Applications*, 29(4):889–899, 2005.

[157] J. Nielsen. *Usability Engineering*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[158] J. Nielsen. *Heuristic evaluation*, pages 25–62. John Wiley Sons, 1994.

[159] N. Nishiuchi and Y. Takahashi. Objective evaluation method of usability using parameters of user's fingertip movement. In *Transactions on Computational Science XXV*, pages 77–89. Springer, 2015.

[160] E. L. Olmsted-Hawala, E. D. Murphy, S. Hawala, and K. T. Ashenfelter. Think-aloud protocols: a comparison of three think-aloud protocols for use in testing data-dissemination web sites for usability. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 2381–2390. ACM, 2010.

[161] K.-M. Osei-Bryson. Towards supporting expert evaluation of clustering results using a data mining process model. *Information Sciences*, 180(3):414–431, Feb. 2010.

[162] A. A. Ozok, D. Benson, J. Chakraborty, and A. F. Norcio. A comparative study between tablet and laptop pcs: User satisfaction and preferences. *International Journal of Human-Computer Interaction*, 24(3):329–352, 2008.

[163] N. Padmanabhan, F. Burstein, L. Churilov, J. Wassertheil, B. Hornblower, and N. Parker. A mobile emergency triage decision support system evaluation. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, volume 5, pages 96b–96b, Jan 2006.

[164] L. Paganelli and F. Paternò. Tools for remote usability evaluation of web applications through browser logs and task models. *Behavior Research Methods, Instruments, & Computers*, 35(3):369–378, 2003.

[165] J. Pascoe, K. Thomson, and H. Rodrigues. Context-awareness in the wild: an investigation into the existing uses of context in everyday life. In *OTM Confederated International Conferences" On the Move to Meaningful Internet Systems"*, pages 193–202. Springer, 2007.

[166] M. J. Pascoe. Adding generic contextual capabilities to wearable computers. In *Proceedings of the 2Nd IEEE International Symposium on Wearable Computers*, ISWC '98, pages 92–92, Washington, DC, USA, 1998. IEEE Computer Society.

[167] F. Paternò, A. Russino, and C. Santoro. Remote evaluation of mobile applications. In *International Workshop on Task Models and Diagrams for User Interface Design*, pages 155–169. Springer, 2007.

[168] I. J. Perez, S. Alonso, F. J. Cabrerizo, and E. Herrera-Viedma. A decision support system based on mobile internet. pages 241–247, Mieres - Langreo, September 2008.

[169] I. J. Pérez, F. J. Cabrerizo, and E. Herrera-Viedma. A mobile decision support system for dynamic group decision-making problems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(6):1244–1256, 2010.

[170] C. Pete, C. Julian, K. Randy, K. Thomas, R. Thomas, S. Colin, and R. Wirth. Crisp-Dm 1.0. *CRISP-DM Consortium*, page 76, 2000.

[171] S. L. Pfleeger and J. M. Atlee. *Software Engineering: Theory and Practice*. Prentice Hall PTR, Prentice Hall: Upper Saddle River, New Jersey, USA, 4th edition, 2009.

[172] G. Piatetsky-Shapiro and W. Frawley. *Knowledge discovery in databases*. MIT Press, Menlo Park, Calif. AAAI Press, 1991. Edited by Gregory Piatetsky-Shapiro and William J. Frawley.

[173] J.-C. Pomerol. Scenario development and practical decision making under uncertainty. *Decision Support Systems*, 31(2):197–204, 2001.

[174] H. Preskill and D. Russ-Eft. *Evaluation in Organizations: A Systematic Approach to Enhancing Learning, Performance, and Change*. New York: Basics Books, 2001.

[175] S. Puuronen and V. Savolainen. Mobile information systems—executives' view. *Information Systems Journal*, 7(1):3–20, 1997.

[176] S. Qureshi, V. Hlupic, and R. O. Briggs. On the Convergence of Knowledge Management and Groupware. In *Groupware: Design, Implementation and Use: 10th International Workshop, CRIWG 2004, San Carlos, Costa Rica, September 5-9, 2004. Proceedings*, number 3198, pages 25–33. 2004.

[177] M. Rauterberg and R. Aeppli. Learning in man-machine systems: the measurement of behavioural and cognitive complexity. In *IEEE In-*

ternational Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century, volume 5, pages 4685–4690. IEEE, 1995.

[178] T. Rebele, F. Suchanek, J. Hoffart, J. Biega, E. Kuzey, and G. Weikum. Yago: a multilingual knowledge base from wikipedia, wordnet, and geonames. In *International Semantic Web Conference*, pages 177–185. Springer, 2016.

[179] J. A. Recio-García, L. Quijano, and B. Díaz-Agudo. Including social factors in an argumentative model for group decision support systems. *Decision Support Systems*, 56:48–55, 2013.

[180] R. A. Rocha, S. M. Maviglia, M. Sordo, and B. H. Rocha. *Clinical Decision Support*, pages 773–817. 2014.

[181] D. Roselin Selvarani. *Recent Trends in Network Security and Applications: Third International Conference, CNSA 2010, Chennai, India, July 23-25, 2010. Proceedings*, chapter Secured Paradigm for Mobile Databases, pages 164–171. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[182] W. W. Royce. Managing the development of large software systems. In *proceedings of IEEE WESCON*, volume 26, pages 1–9. Los Angeles, 1970.

[183] Y. S. Ryu. Mobile phone usability questionnaire (mpuq) and automated usability evaluation. In *International Conference on Human-Computer Interaction*, pages 349–351. Springer, 2009.

[184] Y. S. Ryu and T. L. Smith-Jackson. Reliability and validity of the mobile phone usability questionnaire (mpuq). *Journal of Usability Studies*, 2(1):39–53, 2006.

[185] H. Sahraoui, L. C. Briand, Y.-G. Guéhéneuc, and O. Beaurepaire. Investigating the impact of a measurement program on software quality. *Information and Software Technology*, 52(9):923–933, Sept. 2010.

[186] O. Salo and P. Abrahamsson. Agile methods in european embedded software development organisations: a survey on the actual use and

usefulness of extreme programming and scrum. *IET software*, 2(1):58–64, 2008.

[187] A. C. Santos, J. M. Cardoso, D. R. Ferreira, P. C. Diniz, and P. Chaínho. Providing user context for mobile and social networking applications. *Pervasive and Mobile Computing*, 6(3):324–341, 2010.

[188] J. Sauro and E. Kindlund. A method to standardize usability metrics into a single score. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–409. ACM, 2005.

[189] J. Sauro and J. R. Lewis. *Quantifying the user experience: Practical statistics for user research.* Elsevier, 2012.

[190] L. Sbaiti. Proposition d'un algorithme de régles d'association pour l'extraction des connaissances à partir de données temporelles. Technical report, Master's degree report, Gabes, 2013.

[191] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, WMCSA '94, pages 85–90, Washington, DC, USA, 1994. IEEE Computer Society.

[192] K. Schwaber. Scrum development process. In *Business Object Design and Implementation*, pages 117–134. Springer, 1997.

[193] M. S. Scott Morton. *Management decision systems : computer-based support for decision making.* Division of research, Graduate School of business administration, Harvard University, Boston, 1971. Réimpr. : 1973.

[194] A. Sears and J. A. Jacko. *Human-Computer Interaction: Development Process.* CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2009.

[195] A. Seffah, M. Donyaee, R. B. Kline, and H. K. Padda. Usability measurement and metrics: A consolidated model. *Software Quality Journal*, 14(2):159–178, 2006.

[196] B. Senach. *Evaluation ergonomique des interfaces homme-machine: une revue de la littérature.* INRIA-00075378 Report [Ergonomic evalu-

ation of human machine interfaces: A literature review] France: Sophia Antipolis, 1990.

[197] S. Shan, X. Wang, and M. Sui. Mining association rules: A continuous incremental updating technique. *Proceedings - 2010 International Conference on Web Information Systems and Mining, WISM 2010*, 1:62–66, 2010.

[198] R. Sharda, S. H. Barr, and J. C. MCDonnell. Decision support system effectiveness: a review and an empirical test. *Management science*, 34(2):139–159, 1988.

[199] A. Silberschatz and A. Tuzhilin. What makes patterns interesting in knowledge discovery systems. *IEEE Transactions on Knowledge and data engineering*, 8(6):970–974, 1996.

[200] S. Smithson and R. Hirschheim. Analysing information systems evaluation: another look at an old problem. *European Journal of Information Systems*, 7(3):158–174, 1998.

[201] J.-S. Sottet, G. Calvary, J.-M. Favre, J. Coutaz, A. Demeure, and L. Balme. Towards model driven engineering of plastic user interfaces. In J.-M. Bruel, editor, *Satellite Events at the MoDELS 2005 Conference: MoDELS 2005 International Workshops Doctoral Symposium, Educators Symposium Montego Bay, Jamaica, October 2-7, 2005 Revised Selected Papers*, pages 191–200. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[202] R. H. Sprague and E. D. Carlson. *Building effective decision support systems*. Englewood Cliffs, N.J. Prentice-Hall, 1982.

[203] M. Staron. Critical role of measures in decision processes: Managerial and technical measures in the context of large software development organizations. *Information and Software Technology*, 54(8):887 – 899, 2012.

[204] T. Stewart. Ergonomic requirements for office work with visual display terminals (vdts): Part 11: Guidance on usability. *International Organization for Standardization ISO*, 9241, 1998.

[205] Z. Sun. *Handbook of Research on Demand-Driven Web Services: Theory, Technologies, and Applications: Theory, Technologies, and Applications.* Advances in Web Technologies and Engineering:. IGI Global, 2014.

[206] J. Tague-Sutcliffe. Measuring the informativeness of a retrieval process. In *Proceedings of the 15th annual International ACM SIGIR Conference on Research and development in information retrieval*, pages 23–36. ACM, 1992.

[207] A. Trabelsi and H. Ezzeddine. Evaluation of an information assistance system based on an agent-based architecture in transportation domain: first results. *International Journal of Computers Communications & Control*, 8(2):320–333, 2013.

[208] G. Trabelsi, M. B. Ayed, and A. M. Alimi. Système d'extraction des connaissances à partir des données temporelles basé sur les réseaux bayésiens dynamiques. In *Extraction et gestion des connaissances (EGC'2010), Actes, 26 au 29 janvier 2010, Hammamet, Tunisie*, pages 241–246, 2010.

[209] C. D. Tran, H. Ezzedine, and C. Kolski. *A Generic and Configurable Electronic Informer to Assist the Evaluation of Agent-Based Interactive Systems*, pages 251–263. Springer London, London, 2009.

[210] C. D. Tran, H. Ezzedine, and C. Kolski. Eiseval, a generic reconfigurable environment for evaluating agent-based interactive systems. *International Journal of Human-Computer Studies*, 71(6):725–761, 2013.

[211] E. Turban. *Decision Support and Expert Systems: Management Support Systems*, volume 10. 1986.

[212] E. Turban. *Decision Support and Expert Systems: Management Support Systems.* MacMillan Series in Information Systems. Macmillan, 1990.

[213] B. Vadher, D. Patterson, and M. Leaning. Evaluation of a decision support system for initiation and control of oral anticoagulation in a randomised trial. *BMJ*, 314(7089):1252–1257, 1997.

[214] B. Van de Walle and M. Turoff. Decision support for emergency situations. *Information Systems and e-Business Management*, 6(3):295–316, 2008.

[215] S. Wagner. Quality planning. In *Software Product Quality Control*, pages 91–110. Springer Berlin Heidelberg, 2013.

[216] S. J. Waterson. Webquilt: A visual analysis tool for understanding web usability clickstream data. *Masters of Science, Plan II Report. University of California, Berkeley*, 2002.

[217] G. I. Webb. Efficient search for association rules. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 99–107, New York, NY, USA, 2000. ACM.

[218] W. Wen, Y. H. Chen, and H. H. Pao. A mobile knowledge management decision support system for automatically conducting an electronic business. *Know.-Based Syst.*, 21(7):540–550, Oct. 2008.

[219] J. Winter and K. Rönkkö. Spi success factors within product usability evaluation. *Journal of Systems and Software*, 83(11):2059–2072, 2010.

[220] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[221] B. Wong and R. Jeffery. *Cognitive Structures of Software Evaluation: A Means-End Chain Analysis of Quality*, pages 6–26. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.

[222] T. Wu, Y. Chen, and J. Han. *Knowledge Discovery in Databases: PKDD 2007: 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17-21, 2007. Proceedings*, chapter Association Mining in Large Databases: A Re-examination of Its Measures, pages 621–628. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[223] Y. WU, Y. TIAN, and W. ZHOU. The development of a mobile decision support system. *Journal of Interconnection Networks*, 02(03):379–390, 2001.

[224] J. Xu and M. Quaddus. Examining a model of knowledge management systems adoption and diffusion: A partial least square approach. *Knowledge-Based Systems*, 27:18 – 28, 2012.

[225] W. Z. Y. Wu, Y. Tian. The design of a mobile decision support system. In *International Conference on Algorithms and Architectures for Parallel Processing: ICA3PP*, pages 178–189. World Scientific, 2000.

[226] S. Zaim, N. Bayyurt, M. Tarim, H. Zaim, and Y. Guc. System dynamics modeling of a knowledge management process: A case study in turkish airlines. *Procedia - Social and Behavioral Sciences*, 99:545 – 552, 2013. The Proceedings of 9th International Strategic Management Conference.

[227] M. Zen and J. Vanderdonckt. Towards an evaluation of graphical user interfaces aesthetics based on metrics. In *2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS)*, pages 1–12. IEEE, 2014.

[228] L. S. Zettlemoyer, R. St Amant, and M. S. Dulberg. Ibots: Agent control through the user interface. In *Proceedings of the 4th international conference on Intelligent user interfaces*, pages 31–37. ACM, 1998.

[229] M. Zheng, S. Cheng, and Q. Xu. Context-based mobile user interface. *Journal of Computer and Communications*, 4(09):1–9, 2016.

[230] F. Zhou, B. Yang, L. Li, and Z. Chen. Overview of the new types of intelligent decision support system. In *3rd International Conference on Innovative Computing Information and Control, 2008. ICICIC '08*, pages 267–267, June 2008.

[231] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, and J. Tian. Mining mobile user preferences for personalized context-aware recommendation. *ACM Transactions on Intelligent Systems and Technology*, 5(4):58:1–58:27, Dec. 2014.

A. Appendix A

As evoked in section 2.3.7, several assessment processes have been proposed in the literature such as Ivory and Hearst [98]. According to these authors, usability evaluation is a process that entails some of 12 activities, depending on the method used [98]. This section discusses each of these activities.

1. **Specify Usability Evaluation Goals**: Usability evaluation (UE) is applicable at all stages of a user interface (UI) life cycle (e.g., design, implementation, and re-design). At these various stages, different UE goals are relevant. Below is a list of typical UE goals:

    - Specify UI requirements
    - Evaluate design alternatives
    - Identify specific usability problems
    - Improve UI performance

    The evaluator must clearly specify the goals of the usability evaluation at the outset of the study. These goals influence other aspects of UI assessment, such as the UI components to evaluate and appropriate evaluation methods.

2. **Determine UI Aspects to Evaluate:** Some UIs can be extremely large and complex, and an evaluation of all aspects may not be economically feasible. Hence, the evaluator must determine specific UI aspects to evaluate. These aspects must be consistent with the goals of the usability evaluation.

3. **Identify Target Users:** An interface may be intended for a large user community, but it is important to determine user characteristics most relevant for the study and for the UI aspects in particular. If users are employed during the study, they need to be as representative of the larger user community as possible.

4. **Select Usability Metrics:** Usability metrics are a crucial component of the usability evaluation. The goal in selecting these metrics is to choose a minimal number of metrics that reveal the maximum amount of usability detail for the UI under study. ISO Standard 9241 (1999) recommends using effectiveness, efficiency, and satisfaction measures as described below.

    - Effectiveness is the accuracy and completeness with which users achieve specified goals. Example metrics include: percentage of goals achieved, functions learned, and errors corrected successfully.

    - Efficiency assesses the resources expended in relation to the accuracy and completeness with which users achieve goals. Example metrics include: the time to complete a task, learning time, and time spent correcting errors.

    - Satisfaction reflects users' freedom from discomfort and positive attitudes about use of an interface. Example metrics include: ratings for satisfaction, ease of learning, and error handling.

    Metrics discussed above are quantitative in nature. Non-quantitative metrics could include, for example, specific heuristic violations identified during a usability inspection.

5. **Select Evaluation Method(s):** Choosing one or more usability evaluation methods is an important step of the UE process.

6. **Select Tasks:** Tasks are the most crucial part of the usability evaluation [157]. They must be appropriate for the UI aspects under study, the target users, and the evaluation method. Other constraints may affect the selection of tasks, such as cost and time limits during usability testing sessions, for instance.

7. **Design Experiments:** After completing the previously discussed activities, the evaluator may need to design experiments for collecting usability data. In particular, the evaluator needs to decide on the number of participants (evaluators and users), the evaluation procedure (this is largely dictated by the UE method) as well as on the environment and system setup. The nature of experiments depends on the evaluation method. Experiments may entail: completing tasks in a controlled manner (usability testing); responding to specific questions (inquiry); or comparing alternative designs (analytical modeling and simulation). It is also recommended that the evaluator conduct pilot runs during this phase [157], especially if user involvement is required.

8. **Capture Usability Data:** During this phase, the evaluator employs the UE method to record previously specified usability metrics. For some methods, such as usability testing and inspection, the evaluator may also record specific usability problems encountered during evaluation.

9. **Analyze and Interpret Data:** The primary goal of usability data analysis is to summarize the results in a manner that informs interpretation. This summarization may entail statistical techniques based on the goals of the UE. It may also entail creating a list of specific usability problems found along with their severity. Actually interpreting the results of the study is a key part of the evaluation. It entails using the analysis of usability data to draw conclusions as dictated by the evaluation goals. For example, it may mean concluding that one design is better than another or whether usability requirements have been met.

10. **Critique UI to Suggest Improvements:** Ideally, analysis and interpretation of usability data illustrate aws in the UI design as well

as ways to possibly improve the design. Subsequent analysis may be required to verify that suggested improvements actually improve interface usability.

11. **Iterate Process:** Analysis and interpretation of usability data may illustrate the need to repeat the UE process. This iteration may be warranted due to the identification of other UI aspects that need evaluation or changes to the UI. Hence, UE may consist of several cycles through this process. This is as expected when an evaluator follows usability engineering or iterative design processes [157].

12. **Present Results:** The final step of the usability evaluation process is to communicate the results and interpretation of these results to the stakeholders. Ideally, the evaluator presents the results such that they can be easily understood (e.g., using graphs and providing severity ratings) and acted upon.

B. Appendix B

In this appendix, we present the script evoked in section 4.3.2. It allows our developed system CEVASM to capture the data from the mobile application to be evaluated. The script is presented in Figure B.1. These data would be

```
public class AsyncSendButton extends AsyncTask<String, Integer, Double> {
    @Override
    protected Double doInBackground(String... params) {
        postData(params[0]);
        return null;}
    protected void onPostExecute(Double result){ Log.i("Sent", "OK"); }
    public void postData(String button) {
        // Create a new HttpClient and Post Header
        HttpClient httpclient = new DefaultHttpClient();
        HttpGet Get = new HttpGet("http://IPadress/Emna/test_echo.php?button="
        +button.toLowerCase());
        try
        {   // Execute HTTP Post Request
            HttpResponse response = httpclient.execute(Get);
        }
        catch (ClientProtocolException e) {  Log.e("",e.getMessage());  }
        catch (IOException e) { Log.e("",e.getMessage()); }
    }    }
```

Figure B.1: The script allowing CEVASM to receive data from the mobile application

further treated to be stored in the evaluation database. During the use of the evaluated system by the evaluator, the proposed system allows the capture of the following items, already presented in chapter 3 (section 3.5). These data are received thanks to functions added in the MDSS (see chapter 4, section 4.3.2).

C. Appendix C

As evoqued in section 5.3, to develop our MDSSM/KDD, we applied the approach proposed by Ltifi et al. [136] because it is the most recent an appropriate approach for DSS/KDD process. In this appendix, we present this approach and its application for the development of our MDSSM/KDD. Then we conduct the general architecture of the developed MDSS/KDD process.

The UP/U approach for the design and development of the MDSSM/KDD

The UP/U approach presentation

Since it is intended to allow DSS/KDD to be designed, UP/U approach, proposed by Ltifi et al., puts HCI in a central position, redefining the user's role allowing him/her to intervene at any time in the KDD process [32]. The UP/U is based on the UP principle. It executes, for every KDD module, several complete UP iterations following these four phases [136]:

1. The phase of inception: in which the main cases of use are identified.
2. The phase of elaboration: in which the analysis and the design of most of the features and HCI of the module are approached.
3. The construction phase: in which the design and the realization of the module are finished.
4. The phase of transition: this one is dedicated to the test of the features and the HCI of the module.

We note that we have used UP/U process since the five activities of the original UP process (needs assessment, analysis, design, implementation and testing) do not model the users of the DSS or the system-user interaction. UP/U approach incorporates the continual presence and constant participation of the user throughout the project. Each activity of the adapted U model is divided into sub-activities that model the HCI of the DSS in question. Each of these activities is presented in detail below:

- **Needs assessment** This activity allows the user's functional needs and the non-functional technical needs to be defined. At each UP phase (initialization, development, construction and transition) user-centered activities are carried out. Therefore, to the original UP activity level, we have added the actions "model user" (e.g., the decision-maker), "define and allocate the decisional functions" and "model the automatic, manual and interactive tasks".
- **Analysis** This activity allows the customer needs and requirements to be understood. This understanding leads to defining the specifications in order to choose the design solution. An analysis model provides a complete needs specification based on the use cases and structures these needs in a form (e.g., in a scenario form [47]) that facilitates the comprehension, the preparation, the modification and the maintenance of the future system.
- **Design** This activity provides a more accurate understanding of the constraints related to the programming language, the use of compo-

nents and the operating system. It also determines the architecture of the automatic and interactive modules.

- **Implementation** This activity is the result of the design. Its main objectives are planning the integration of the components and producing the classes and providing the source code. This activity includes also the interfaces implementation according to the defined specifications.

- **Testing** This UP activity allows the results to be verified. It must be carried out at the same time as the activities suggested for the U model, notably tests with the users and the comparison of the tasks initially specified by the designer and the tasks really accomplished by the users.

The UP/U approach application for the development of a MDSSM/KDD

Ltifi *et al.* have applied their UP/U approach in the medical field. In fact, we have the same goal which is predicting nosocomial infections in the same intensive care unit of Habib Bourguiba hospital-Sfax, Tunisia. Consequently, the activities *Needs assessment*, *Analysis*, and *Design* were already achieved by Ltifi *et al.*. However, The *Implementation* was performed on an immobile platform. So, before carrying out our evaluation for MDSSM/KDD, we had to implement such system that can be used through mobile devices. Then, once the system is implemented, we can apply our approach for the test of all the KDD modules.

The implementation of these modules requires the follow up of the diagram presented in Figure C.1. Table C.1 details the implementation activity that consists in coding the functional parts, based on the algorithms (in the design activity), and the user interfaces. All the code components are then assembled and integrated in a subsystem in order to build a prototype at the end of the iteration.

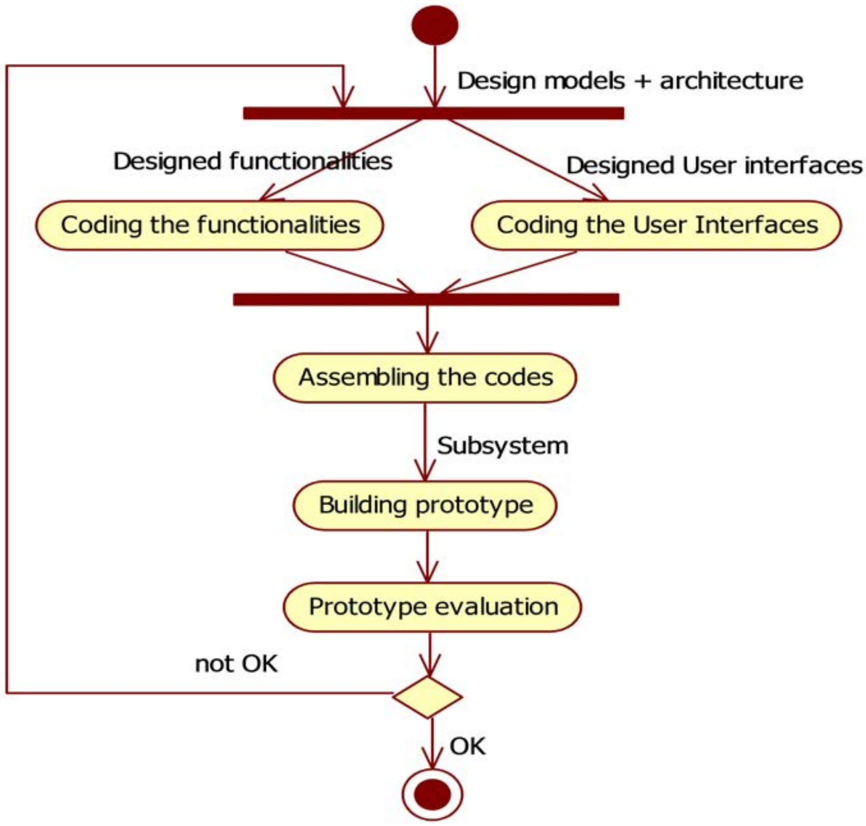

Figure C.1: The implementation activity used for each module of the DSS/KDD process, according to [136]

General architecture of the developed MDSS/KDD process

In this section, we present the architecture of the developed MDSS/KDD process. It is composed of two parts: hardware architecture and software architecture.

1. The hardware configuration that was used for the development of our application is the following:

    - Server: it is the component which includes the algorithms of our application as well as the database.

Table C.1: The implementation of the MDSSM/KDD modules.

Module	**Specificities**
Data acquisition and storage module	(1) Implementing the data acquisition user interfaces, and the software packages. (2) Assembling the code components for the user interfaces and the software packages in order to build the prototype.
Data-mining module	(1) Coding the data-mining user interfaces. (2) Assembling the code components for the user interfaces and the software packages (in the application server) in order to build the prototype.
Knowledge management module	(1) Implementing the user interfaces and the software packages for the prediction, possible solution generation and decision-making submodules. (2) Assembling the software packages for the prediction and the code components for the user interfaces to build a prototype.

- Mobile device: it is the most important component. The end user can use these devices to take advantage the functionalites of the different prototypes.
- Computer: this is the component that retrieves the web page that was created to extract the rules of association from the browser (For data mining and knowledge management modules).

2. To develop our application, we take advantage the following elements:

   - Java EE: this tool includes the application server[1] JBoss, the

[1] An application server is a software framework that provides both facilities to create web applications and a server environment to run them.

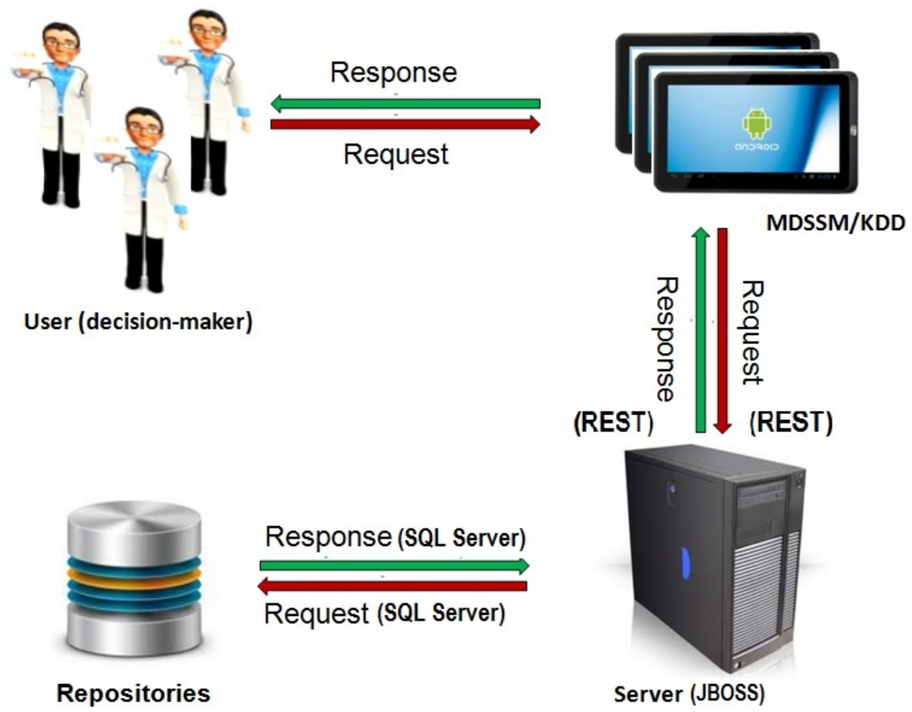

Figure C.2: General architecture of the developed MDSSM/KDD process

web server REST[2], servlets[3] and the JSP[4].

- Java for Android: the platform that we used to develop the prototypes that corresponds to the KDD process.
- Microsoft SQLSERVER: database management system that includes the different repositories.

[2]Representational state transfer (REST) web services are one way of providing interoperability between computer systems on the Internet

[3]A Java servlet is a Java program that extends the capabilities of a server. Although servlets can respond to any types of requests, they most commonly implement applications hosted on Web servers. Such Web servlets are the Java counterpart to other dynamic Web content technologies such as PHP

[4]Servlets can be generated automatically from Java Server Pages (JSP) by the JavaServer Pages compiler. The difference between servlets and JSP is that servlets typically embed HTML inside Java code, while JSPs embed Java code in HTML.

Tunisian republic
Ministry of Higher Education and Scientific Research
University of Sfax
National Engineering school of Sfax

University of Valenciennes and Hainaut-Cambresis

Doctoral Schools:
Science and Technology
&
Sciences for the Engineer

Doctoral thesis

2016 – 2017

An approach for the evaluation of Mobile Decision support Systems based on a Knowledge Discovery from Data process: Application in the medical field

Emna BEN AYED

الخلاصة: في هذا العمل، نحن مهتمون بتقييم نظم دعم القرار المحمول، التي تقوم على عملية اكتشاف المعرفة من البيانات. نقترح إلحاق وحدة دعم و تقييم لكل وحدة نمطية تكون عملية اكتشاف المعرفة من البيانات استناداً إلى نماذج الجودة .تسمح وحدات التقييم المقترحة بتقييم نوعية استخدام كل وحدة تؤلف عملية اكتشاف المعرفة من البيانات و معايير أخرى تعكس استراتيجيات كل وحدة نمطية . هدفنا الرئيسي هو مساعدة القائمين على الكشف بغية تحسين نوعية جميع الوحدات التي تشكل عملية اكتشاف المعرفة من البيانات .وقد أخذنا أيضا في الاعتبار تغير سياق الاستخدام. وبالإضافة إلى ذلك، اقترحنا نظام تقييم يراقب ويقيس جميع المعايير المقترحة : سيفاسم. وأخيراً، طبقنا هذا النهج لتقييم الوحدات النمطية من نظام دعم القرار المحمول لمكافحة عدوى المستشفيات، في مستشفى الحبيب بورقيبة بصفاقس .ونحن نتابع عملية التقييم استناداً لمعيار ISO/IEC 25040. والهدف أن تكون قادرة على التحقق من صحة أداة التقييم سيفاسم ونتيجة لذلك، النهج المقترح.

Résumé : Dans ce travail, on s'intéresse aux Systèmes d'Aide à la Décision Mobiles qui sont basés sur le processus d'Extraction des Connaissances à partir des Données (SADM/ECD). Nous contribuons non seulement à l'évaluation de ces systèmes, mais aussi à l'évaluation dans le processus d'ECD lui-même. L'approche proposée définit un module de support d'évaluation pour chaque module composant le processus d'ECD en se basant sur des modèles de qualité. Ces modules évaluent non seulement la qualité d'utilisation de chaque module logiciel composant le processus d'ECD, mais aussi d'autres critères qui reflètent les objectifs de chaque module de l'ECD. Notre objectif est d'aider les évaluateurs à détecter des défauts le plus tôt possible pour améliorer la qualité de tous les modules qui constituent un SADM/ECD. Nous avons aussi pris en compte le changement de contexte d'utilisation en raison de la mobilité. De plus, nous avons proposé un système d'aide à l'évaluation, nommé CEVASM : Système d'aide à l'évaluation basée sur le contexte pour les SADM, qui contrôle et mesure tous les facteurs de qualité proposés. Finalement, l'approche que nous proposons est appliquée pour l'évaluation des modules d'un SADM/ECD pour la lutte contre les infections nosocomiales à l'hôpital Habib Bourguiba de Sfax, Tunisie. Lors de l'évaluation, nous nous sommes basés sur le processus d'évaluation ISO/IEC 25040. L'objectif est de pouvoir valider, *a priori*, l'outil d'évaluation réalisé (CEVASM) et par conséquent, l'approche proposée.

Abstract: In this work, we are interested in Mobile Decision support systems (MDSS), which are based on the Knowledge Discovery from Data process (MDSS/KDD). Our work is dealing with the evaluation of these systems, but also to the evaluation in the KDD process itself. The proposed approach appends an evaluation support module for each software module composing the KDD process based on quality models. The proposed evaluation support modules allow to evaluate not only the quality in use of each module composing the KDD process, but also other criteria that reflect the objectives of each KDD module. Our main goal is to help evaluators to detect defects as early as possible in order to enhance the quality of all the modules that constitute a MDSS/KDD. We have also presented a context-based method that takes into account the change of context of use due to mobility. In addition, we have proposed an evaluation support system that monitors and measures all the proposed criteria. Furthermore, we present the implementation of the proposed approach. These developments concern mainly the proposed evaluation tool: CEVASM: Context-based EVAluation support System for MDSS. Finally, the proposed approach is applied for the evaluation of the modules of a MDSS/KDD for the fight against nosocomial infections, in Habib Bourguiba hospital in Sfax, Tunisia. For every module in KDD, we are interested with the phase of evaluation. We follow the evaluation process based on the ISO/IEC 25040 standard. The objective is to be able to validate, *a priori*, the realized evaluation tool (CEVASM) and consequently, the proposed approach.

المفاتيح: نظم دعم القرار ,عملية اكتشاف المعرفة من البيانات ,تقييم ,تغير سياق الاستخدام, المحمول.

Mots clés : système d'aide à la décision, extraction des connaissances à partir des données, évaluation, contexte d'utilisation, mobilité.

Key-words: decision support system, knowledge discovery from data, evaluation, context of use, mobility.

<u>Résumé du mémoire de thèse de</u>

Emna Ben Ayed

En vue de l'Obtention du

Doctorat en Informatique

à

<u>l'école nationale d'ingénieurs de Sfax</u>

<u>et</u>

<u>l'université de valenciennes et et du hainaut cambrésis</u>

Titre de la thèse :

«An approach for the evaluation of mobile decision support system based on the knowledge discovery from data process: Application in the medical field»

Le domaine de l'Aide à la décision est particulièrement vaste, source de nombreuses propositions, aussi bien dans le milieu académique qu'industriel. Au milieu des années quatre-vingt sont apparus les outils d'aide à la décision fournissant à un décideur ou une équipe de décideurs des indicateurs et des analyses. Ces outils permettent de faciliter l'accès aux données en ouvrant la possibilité à des analyses plus complètes. La tendance actuelle est d'aller vers les outils permettant le passage de l'information à la connaissance. On parle de l'Extraction de Connaissances à partir des Données (ECD), outil de manipulation et d'exploitation de données. L'ECD est apparu pour explorer la gigantesque quantité de données et d'en extraire des nouvelles connaissances utiles, aidant à prendre des décisions à propos de divers sujets qui touchent le domaine dans lequel on travaille. On s'intéresse donc à des Systèmes Interactifs d'Aide à la Décision (SIAD) basés sur l'ECD

Cette thèse se situe dans le domaine des systèmes d'information et plus précisément des systèmes d'aide à la décision. La thèse traite des systèmes mobiles sous l'angle de l'évaluation. Elle vise à assister les experts à évaluer les systèmes d'aide à la décision basés sur le processus d'extraction des connaissances à partir des données (SIADM/ECD).

Aujourd'hui, les progrès des technologies mobiles et la large disponibilité des appareils mobiles personnels créent une nouvelle classe de Systèmes d'aide à la décision (SIAD) connus sous le nom SIAD mobiles, qui offrent aux utilisateurs la possibilité de prendre des décisions appropriées à tout moment et n'importe où, via leurs appareils mobiles. Ainsi, les développeurs tentent de développer des SIAD dans plusieurs domaines d'application. Ces systèmes devraient permettre aux utilisateurs de gérer facilement la base de connaissances et permettre la prise de décision rapide et efficace.

Bien que ces dernières années ont vu un intérêt accru au sein de la communauté et dans la recherche associée à l'évaluation de ces systèmes, cette dernière n'a pas été un point principal. La phase d'évaluation a toujours été négligée. Peu de chercheurs ont souligné cette lacune et ont défini les critères qui doivent être optimisés pour obtenir une meilleure qualité de SIAD. Cependant, ces travaux restent limités au SIAD non mobiles.

La recherche dans le domaine des systèmes d'aide à la décision débouche sur l'apparition de nouvelles technologies et de concepts qui concernent l'analyse des données et la découverte de connaissances, qui sont nécessaires pour le processus de prise de décision. En particulier, le processus d'extraction des connaissances à partir des données (ECD) est le processus le plus connu et le plus utilisé pour le développement de SIAD. Nous nous intéressons uniquement à une technologie particulière utilisée pour la prise de décision : le SIAD mobile basé sur processus d'ECD.

Le problème traité est l'insuffisante prise en compte du contexte d'usage dans les systèmes d'aide à la décision alors que paradoxalement l'informatique devient ubiquitaire permettant une interaction en tout lieu, à tout instant, via des dispositifs de plus en plus variés. La thèse défend l'importance d'une évaluation en contexte tissée dans le processus de décision (data acquisition, data mining, knowledge management). Elle milite pour une ingénierie outillée de l'évaluation : conception de l'évaluation, exécution de l'évaluation puis conclusion statistique sur la base d'indicateurs observés.

Ce travail contribue non seulement à l'évaluation de ces systèmes, mais aussi à l'évaluation dans le processus d'ECD lui-même. L'approche proposée définit un module de support d'évaluation pour chaque module composant le processus d'ECD, en se basant sur des modèles de qualité. Ces modules évaluent la qualité d'utilisation de chaque module logiciel composant le processus d'ECD, et d'autres critères qui reflètent les objectifs de chaque module de l'ECD. L'objectif est d'aider les évaluateurs à détecter des défauts pour améliorer la qualité de tous les modules qui constituent un

SIADM/ECD. En raison de la mobilité, l’approche prend en compte le changement de contexte d'utilisation. De plus, un système d’aide à l’évaluation a été proposé. Il est nommé CEVASM (Système d’aide à l’évaluation basée sur le contexte pour les SIADM). Son rôle est de mesurer tous les facteurs de qualité proposés. Finalement, l'approche est appliquée pour l'évaluation des modules d'un SIADM/ECD pour la lutte contre les infections nosocomiales à l'hôpital Habib Bourguiba de Sfax, Tunisie.

****

Introduction générale

Dans l'**Introduction générale** de 5 pages, le contexte, la motivation, la problématique et les objectifs sont introduits :

Le processus ECD, proposé par Fayyad en 1996, est le procédé le plus connu et le plus utilisé pour le développement du SIAD. Nous ne sommes intéressés que par une technologie particulière utilisée pour la prise de décision : SIADM / ECD. La principale préoccupation de cette recherche est l'évaluation de ces systèmes. L'approche que nous proposons ajoute un module de support d'évaluation pour chaque module composant le processus ECD basé sur des modèles de qualité.

Les modules d'aide à l'évaluation proposés permettent d'évaluer non seulement la qualité d'utilisation de chaque module composant le processus ECD, mais aussi d'autres critères qui reflètent les objectifs de chaque module ECD. Notre objectif principal est d'aider les évaluateurs à détecter les défauts le plus tôt possible afin d'améliorer la qualité de tous les modules qui constituent un SIADM / ECD.

Dans cette introduction, la structure du mémoire est également dévoilée. Il se compose de cinq chapitres et se finalise par une conclusion générale, des perspectives et une liste des références et des publications.

Chapitre 1

Le **premier chapitre** de la thèse s'intitule « ***Etat de l'art : les systèmes d'aide à la décision et le processus d'extraction des connaissances à partir des données*** » (27 pages). Il vise à présenter les concepts de base.

- ✓ Au début le SIAD aussi bien que sa composition et son évolution sont présentés. Puis, le SIADM, la nouvelle génération de SIAD, est présenté. Ensuite, trois exemples de processus d'extraction des connaissances à partir des données sont détaillés, avec un accent sur un processus particulier qui représente un lien entre des systèmes d'aide à la décision et le processus ECD. Un large éventail d'applications, y compris des systèmes d'aide à la décision, supportant les appareils mobiles et les attentes des utilisateurs s'améliorent progressivement.

Depuis les travaux de Scott Morton (1971), le domaine des SIAD n'a cessé d'évoluer. Le concept de système d'aide à la décision est extrêmement vaste et ses définitions dépendent du point de vue de chaque auteur. Un tel système peut prendre de nombreuses formes et peut s'utiliser de diverses manières. D'une manière générale, on peut le définir comme étant "un système informatique qui facilite le processus de prise de décision".

D'autres définitions des SIAD existent dans la littérature, ces diverses définitions portent soit sur le type de problème de décision, soit sur les fonctions du système, soit sur sa constitution ou encore sur le processus de développement. Nous reprenons ici la définition de Turban (1993), qui porte à la fois sur les fonctions et les composants du système : "Un SIAD est un système d'information interactif, flexible, adaptable et spécifiquement développé pour aider à la résolution d'un problème de décision en améliorant la prise de décision. Il utilise des données, fournit une interface utilisateur simple et autorise l'utilisateur à développer ses propres idées ou points de vue. Il peut utiliser des modèles – soit standards, soit spécifiques -, supporter les différentes phases de la prise de décision et inclure une base de connaissances".

De nos jours, le SIADM est souvent utilisé dans des environnements changeants, mais ne s'adapte pas très bien à ces changements. Bien que l'éloignement du modèle de bureau apporte une variété de nouvelles situations dans lesquelles une application peut être utilisée, les périphériques informatiques connaissent rarement leurs environnements environnants. Ainsi, l'information dans les environnements physiques et opérationnels des appareils mobiles crée un contexte d'interaction entre les utilisateurs et les périphériques. Comme l'avenir de la mobilité est inhérent au concept de contexte, le SIADM doit fournir un support en fonction du contexte d'utilisation. En effet, la mobilité offre la possibilité de prendre conscience de l'individu et de ses interactions avec son environnement en constante évolution.

Le contexte est une notion utilisée depuis longtemps dans des conceptions d'applications interactives. Les dictionnaires le définissent, par exemple, comme étant un "ensemble d'informations dans lequel se situe" quelque chose, ou encore en tant qu' "ensemble qui entoure". Ces définitions restent cependant relativement abstraites face aux nombreuses utilisations de ce terme dans les diverses disciplines scientifiques ou littéraires - ce qui rend donc la formalisation du contexte relativement difficile, ou spécifique à un domaine particulier. Les recherches en Interaction Homme-Machine, par exemple, ont précisé ces définitions apportant divers éléments permettant de le qualifier de manière plus approfondie. C'est le cas par exemple de Calvary et al. en 2004, qui proposeront de définir le contexte d'usage selon le triplet <utilisateur, plateforme, environnement>.

C'est sur cette proposition que nous baserons nos recherches en proposant d'intégrer et de définir la notion de contexte pour les tables interactives.

Ainsi, un modèle de contexte est nécessaire pour définir et stocker des informations contextuelles. Bien que les architectures existantes apportent des avantages à leurs utilisateurs, certaines lacunes entravent la croissance et la mise en œuvre de cette technologie dans des environnements réels. La nécessité de systèmes mobiles de haute qualité devient plus nécessaire. Ainsi, d'autres projets de recherche et de développement peuvent couvrir ces lacunes et permettre la mise en place de systèmes plus répandus. Par la suite, l'évaluation de ces systèmes semble être très nécessaire.

<u>Chapitre 2</u>

Le **deuxième chapitre** intitulé « ***L'évaluation des systèmes interactifs*** » (31 pages) présente un état de l'art concernant l'évaluation de ces systèmes, puisque les SIAD sont souvent fortement interactifs.

Après plus d'une vingtaine d'années de recherche et de pratique, il est indéniable que l'évaluation des systèmes interactifs est une activité essentielle pour produire des systèmes de haute qualité. L'évaluation est le processus qui consiste à estimer ou à justifier la valeur des systèmes évalués. Il présente l'un des plus grands intérêts de la communauté de l'interaction homme-machine (HCI). Par conséquent, un système interactif de haute qualité offre non seulement le succès dans l'industrie mais aussi la satisfaction de l'utilisateur final. De nombreux concepts, méthodes et outils d'évaluation ont été proposés par la communauté de HCI afin de valider un système déjà construit ou en cours de conception ou de développement et d'améliorer son efficacité.

Il y a un nombre croissant et de nombreux types de systèmes interactifs qui sont développés pour les utilisateurs finaux. Parmi ces systèmes, nous trouvons les Systèmes d'information (IS). L'incorporation d'utilisateurs dans l'évaluation de l'IS a été identifiée comme une préoccupation importante pour les chercheurs IS. Cependant, l'évaluation a toujours été un problème en ce qui concerne les systèmes d'aide à la décision qui ont été introduits dans la littérature en informatique, particulièrement concernant les systèmes d'information.

Dans ce chapitre, nous nous sommes focalisés sur les **approches centrées sur l'utilisateur.** Ces approches sont basées sur des techniques d'observation de l'utilisateur réel (utilisateurs finaux) et de recueil des données de l'interaction (questionnaire, interview, verbalisation, etc.) afin d'analyser les traces de l'activité des utilisateurs. Ces approches permettent de détecter les problèmes réels que rencontre l'utilisateur lorsqu'il réalise sa tâche avec le système. Les résultats portent sur l'interface et le système mais ils n'offrent pas les moyens de corriger les erreurs.

Parmi ces approches centrées sur l'utilisateur on peut citer les approches empiriques de diagnostic d'usage (utilisables lorsque l'IHM est réalisée totalement ou partiellement), les approches centrées sur l'estimation de la charge de travail et les approches basées sur les tests de conception (interviennent tout au long du cycle de développement de l'IHM).

Le problème c'est qu'il n'y a aucun précédent à suivre pour l'évaluation de SIADM. Dans ce travail, nous ne sommes intéressés que par une technologie particulière qui s'appuie sur les systèmes mobiles de soutien à la décision (SIADM) utilisés et mis en œuvre dans plusieurs domaines. Leur importance potentielle pour soutenir l'accès en temps opportun aux informations critiques implique de réduire les erreurs et d'améliorer l'accès à toutes les informations préalablement centralisées. Cependant, certaines lacunes entravent la croissance et la mise en œuvre de cette technologie dans un environnement réel. En fait, de nombreuses exigences telles que la mobilité et la sensibilisation au contexte doivent être respectées.

Ainsi, d'autres projets de recherche et de développement peuvent couvrir ces lacunes et permettre la mise en place de systèmes plus répandus. Grâce à ce chapitre, nous présentons les travaux les plus connus dans le domaine de l'évaluation des systèmes interactifs. Dans la première section, nous présentons les processus de développement les plus connus dans l'ingénierie logicielle, en mettant l'accent sur l'évaluation dans ces processus. Dans la deuxième section, nous établissons un état de l'art sur l'évaluation dans le domaine de l'interaction homme-ordinateur, y compris les méthodes, les techniques et les processus d'évaluation existants.

- ✓ Ce chapitre commence par une présentation des travaux antérieurs concernant l'évaluation dans les processus de développement dans le domaine du génie logiciel aussi bien que dans le domaine de l'interaction homme-machine. Ensuite, certaines théories et normes (standards) de mesure de qualité sont introduites telles que les normes ISO 9126, ISO 25010 et ISO 25020. Finalement, un état de l'art de l'évaluation de systèmes d'aide à la décision est effectué. Ce chapitre est clôturé par une synthèse et une conclusion

Chapitre 3

Dans le **troisième chapitre** (31 pages), intitulé « ***Contributions quant à l'évaluation des systèmes d'aide à la décision mobiles basés sur le processus d'extraction des connaissances à partir des données*** », la contribution quant au processus ECD est présentée.

Vu que peu de progrès ont été réalisés pour améliorer les méthodes d'évaluation des systèmes d'aide à la décision (SIAD) dans le processus d'extraction des connaissances à partir des données (ECD), peu d'efforts ont été déployés sur les aspects de mesure pour évaluer la qualité de ces systèmes.

Principalement, notre contribution consiste à ajouter un module de support d'évaluation dans chaque module composant le processus ECD. Ainsi, nous pouvons établir un système d'aide à l'évaluation qui aide l'évaluateur (s) à avoir une idée globale de la qualité du système évalué. L'évaluation n'est plus un module centralisé dans le processus ECD précédemment (dans plusieurs travaux de recherche) concernés uniquement par l'évaluation des modèles extraits. L'évaluation a été intégrée au module Data mining (fouille de données). Par conséquent, la sortie de ce module serait à l'origine intéressante. À notre avis, le module d'acquisition et de stockage de données (le premier module dans le processus ECD), ainsi que le module de gestion des connaissances (dernier module dans le processus ECD), devraient également intégrer un module de support d'évaluation.

Notre proposition permet une évaluation plus améliorée de tous les modules du processus (l'acquisition de données et le stockage, extraction des connaissances et gestion des connaissances). Par conséquent, toute fonction doit être évaluée et au plus tôt, y compris la collecte des données. Le principe de l'instrumentation est également pertinent : il permet un passage à l'échelle, une objectivation du diagnostic, une réutilisation et probablement une évolution.

- ✓ Dans la deuxième partie de ce chapitre, une méthode à base d’un modèle de contexte est proposée. Elle prend en compte le changement de contexte d'utilisation en raison de la mobilité. Dans la troisième section, un système de support d'évaluation est conçu. Il permet la mesure de tous les critères détaillés dans la première partie de ce chapitre.

Chapitre 4

Dans le **quatrième chapitre** (21 pages), intitulé « ***CEVASM : Un système d'aide à l'évaluation basé sur le contexte pour l'évaluation des systèmes d'aide à la décision basés sur le processus d'extraction des connaissances à partir des données*** », la mise en œuvre de l'approche est introduite, ainsi que les développements réalisés pour la mise en place de l'approche proposée. Ces développements concernent principalement l'outil d'évaluation appelée (CEVASM). Ce dernier offre une synthèse des mesures obtenues lors de l'évaluation.

CEVASM est une application Web qui a été développée à l'aide du langage de programmation PHP 5. Il permet de mesurer non seulement des critères d'évaluation subjectifs, mais aussi objectifs se référant à l'ensemble du processus ECD. L'objectif du développement du CEVASM est d'aider les évaluateurs à évaluer la qualité d'un processus ECD.

Dans la prochaine section, nous détaillons la mise en œuvre de notre système de soutien à l'évaluation (CEVASM) composé de plusieurs composants qui sont :

- Composant d'évaluation de la qualité d'utilisation, qui est exécuté lorsque nous sommes confrontés à tous les modules d'acquisition de données, d'exploration de données, de gestion des connaissances.
- Composant d'évaluation de la qualité des données, qui est exécuté lorsque nous sommes confrontés au module d'acquisition de données.
- Composant d'évaluation d'intérêt, qui est exécuté lorsque nous sommes confrontés au module Data mining.
- Composant d'évaluation de performance du classificateur, qui est exécuté lorsque nous sommes confrontés au module de gestion du savoir.

- Le composant d'évaluation de la qualité de la promotion, qui est exécuté lorsque nous sommes également confrontés au module de gestion du savoir.
- Composant d'évaluation global, qui est exécuté après l'évaluation de tous les modules ECD. Il concerne l'ensemble du processus ECD.

✓ Ce chapitre contient trois sections y compris la présentation du système développé et ses principaux objectifs. Le processus d'évaluation ainsi que les développements sont détaillés. Ce chapitre est clôturé par une conclusion concernant les travaux réalisés.

Chapitre 5

Dans le **cinquième chapitre** (33 pages), l'approche proposée est appliquée pour l'évaluation des modules d'un SIADM/ECD pour la lutte contre les infections nosocomiales, en suivant le processus d'évaluation défini dans le chapitre 4. L'objectif de ce chapitre est de valider, *a priori* l'évaluation réalisée par l'outil (CEVASM) et, par conséquent, l'approche proposée.

En effet, ce chapitre démontre la faisabilité technique d'un environnement logiciel par le développement de l'outil CEVASM. Il rapporte une série d'évaluations (sur 3 prototypes) menées en grande partie sur le terrain avec 33 sujets de plusieurs profils, incluant des professionnels du domaine de la santé. Notre approche est appliquée à un cas concret dans le domaine médical pour le développement d'un SIAD visant la lutte contre les infections nosocomiales. Ce système a été développé en collaboration avec l'équipe du service de réanimation du Centre Hospitalo-universitaire Habib Bourguiba à Sfax, Tunisie. Il a pour but d'aider les médecins à prévoir l'apparition des infections nosocomiales pour les patients en réanimation.

Les infections nosocomiales (IN) représentent un des problèmes majeurs de la santé publique. Ce sont des infections contractées dans un établissement de soins. Une infection est considérée comme telle lorsqu'elle était absente au moment de l'admission du patient. Lorsque l'état infectieux du patient à l'admission est inconnu, l'infection est classiquement considérée comme nosocomiale si elle apparaît après un délai de 48 heures d'hospitalisation. Ces infections peuvent être directement liées aux soins ou simplement survenir lors de l'hospitalisation indépendamment de tout acte médical.

En réalité, les SIADM sont utilisés dans différentes situations, environnements, dans des conditions différentes et, par différents profils d'utilisateurs. Lorsque le contexte de l'utilisation change en raison d'un ou de plusieurs facteurs, la performance du SIADM peut aussi changer; mais il existe des aspects fonctionnels indépendants du contexte. Pour cette raison, nous considérons que les critères qui ne sont pas conscients du contexte peuvent être suffisamment évalués sous une seule instance de contexte d'utilisation. Néanmoins, l'évaluation des critères contextuels devrait être sensible à l'évolution du contexte. Cette variété est principalement due à trois causes, à savoir l'utilisateur, la technologie et l'environnement.

Nous avons réalisé cette étude afin de rechercher des critères contextuels qui sont principalement sensibles aux caractéristiques caractérisant le contexte d'utilisation (le profil utilisateur, la plate-forme et l'environnement). Nous avons spécifié la fonction de sensibilisation de chaque critère et nous avons proposé un ensemble de critères complémentaires qui peuvent contribuer à l'évaluation du processus SIADM / ECD. Comme dernière étape de cette étude, il faudrait classer les critères utilisés pour l'évaluation d'un SIADM / ECD. La plupart de ces critères sont sensibles au contexte dans lequel le prototype est utilisé.

Ensuite, nous pouvons conclure que les facteurs de qualité qui sont purement liés aux dépôts (la base de données, base des modèles, la base de connaissances) ne sont pas sensibles au contexte d'utilisation.

- ✓ Dans ce chapitre le contexte général du travail est présenté, ainsi que le SIADM/ECD à évaluer à l’aide du système de support d’évaluation. Puis, une discussion qui concerne les risques de la validité de la proposition est effectuée. Cette discussion ouvre plusieurs perspectives de recherche dans le but d’améliorer et étendre l’approche proposée. Ces perspectives sont citées dans la dernière partie de ce chapitre.

Conclusion générale

La « **Conclusion générale** » est présentée en 7 pages, elle résume le contenu de la thèse, les travaux et l’évaluation menés visant à valider les contributions théoriques. Les limites des propositions présentées sont également montrées.

Sur la base de ces limites, nous présentons ensuite les principales perspectives de notre recherche, qui concernent nos composants du système d’aide à l'évaluation :

- Nous avons l'intention de généraliser l'approche pour l'évaluation de différents types de SIADM (utilisés non seulement dans les cas de prédiction) tout en considérant différents contextes d'utilisation.
- Afin d'améliorer l'acquisition de données contextuelles et de faciliter la tâche de l'évaluateur, il serait utile d'utiliser des capteurs qui capturent les données contextuelles automatiquement, facilement et rapidement. Grâce aux capteurs, nous pouvons collecter beaucoup plus de données contextuelles qui peuvent nous aider à détecter les défauts d'un SIADM.

****

Dans ce travail, on s'intéresse aux Systèmes d'Aide à la Décision Mobiles qui sont basés sur le processus d'Extraction des Connaissances à partir des Données (SIADM/ECD). Nous contribuons non seulement à l'évaluation de ces systèmes, mais aussi à l'évaluation dans le processus d'ECD lui-même. L'approche proposée définit un module de support d'évaluation pour chaque module composant le processus d'ECD en se basant sur des modèles de qualité. Ces modules évaluent non seulement la qualité d'utilisation de chaque module logiciel composant le processus d'ECD, mais aussi d'autres critères qui reflètent les objectifs de chaque module de l'ECD. Notre objectif est d'aider les évaluateurs à détecter des défauts le plus tôt possible pour améliorer la qualité de tous les modules qui constituent un SIADM/ECD. Nous avons aussi pris en compte le changement de contexte d'utilisation en raison de la mobilité. De plus, nous avons proposé un système d'aide à l'évaluation, nommé CEVASM : Système d'aide à l'évaluation basée sur le contexte pour les SIADM, qui contrôle et mesure tous les facteurs de qualité proposés. Finalement, l'approche que nous proposons est appliquée pour l'évaluation des modules d'un SIADM/ECD pour la lutte contre les infections nosocomiales à l'hôpital Habib Bourguiba de Sfax, Tunisie. Lors de l'évaluation, nous nous sommes basés sur le processus d'évaluation ISO/IEC 25040. L'objectif est de pouvoir valider, *a priori*, l'outil d'évaluation réalisé (CEVASM) et par conséquent, l'approche proposée.