



# A user-centered approach for the design and implementation of KDD-based DSS: A case study in the healthcare domain

Mounir Ben Ayed, Hela Ltifi, Christophe Kolski, Adel M. Alimi

## ► To cite this version:

Mounir Ben Ayed, Hela Ltifi, Christophe Kolski, Adel 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*, 2010, 50 (1), pp.64-78. 10.1016/j.dss.2010.07.003 . hal-03285454

**HAL Id: hal-03285454**

**<https://uphf.hal.science/hal-03285454>**

Submitted on 21 Dec 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A User-centered Approach for the Design and Implementation of KDD-based DSS: A case Study in the Healthcare Domain

Mounir Ben Ayed <sup>(\*,\*\*)</sup> - Hela Ltifi <sup>(\*,\*\*\*)</sup> - Christophe Kolski <sup>(\*\*\*)</sup> – Adel M. Alimi <sup>(\*)</sup>

<sup>(\*)</sup> REGIM: REsearch Group on Intelligent Machines,

National School of Engineers (ENIS) of Sfax, BP 1173, Sfax, 3038, Tunisia

[hela\\_ltifi@ieee.org](mailto:hela_ltifi@ieee.org), [mounir.benayed@ieee.org](mailto:mounir.benayed@ieee.org), [adel.alimi@ieee.org](mailto:adel.alimi@ieee.org)

<sup>(\*\*)</sup> Faculty of Sciences of Sfax, Computer science and Communications Department

Route Sokra Km 3.5 - BP 1171 – 3000, Sfax, Tunisia

<sup>(\*\*\*)</sup> LAMIH - UMR CNRS 8530, Université de Valenciennes and Hainaut-Cambrésis,

Le Mont Houy, F-59313 Valenciennes cedex 9, France

[Christophe.Kolski@univ-valenciennes.fr](mailto:Christophe.Kolski@univ-valenciennes.fr)

## ABSTRACT:

In this article we propose an approach for a decision support system (DSS) based on Knowledge Discovery from Databases (KDD). In such system, user must be involved throughout the decision-making process. In consequence we propose the integration of a Human-Computer Interaction (HCI) model into the development of DSS process based on KDD. The approach we propose is based on two systems development methods—the Unified Process (UP) from Software Engineering and the U Model from HCI. In this article, we describe our combined approach (UP/U) and the way we used it to develop a DSS in a medical field.

**Key-words:** Decision support system; Knowledge Discovery from Databases; Data Mining, Human-Computer Interaction; Software Engineering; Unified Process; U Model.

## 1. Introduction

This article stems from our research on the user-centered design of Interactive Decision Support Systems (DSS) based on a process of Knowledge Discovery from Data (KDD). The evolution of data processing and the increasing complexity of the problems involved have generated needs to more and more technological solutions for decision process strategies. DSS deal with problems based on the available knowledge. Some of this knowledge can be extracted using a decisional support tool called Data Mining [13, 17, 31], which is in fact part of the KDD process (see section 2). Many techniques can be used to mine data and extract new information. This new information needs to be interpreted and evaluated in order to propose it as a valid element of decision support. Data mining tools are usually difficult to exploit because most of the end users are expert neither in computer science nor in statistics. It is also difficult to develop a KDD system that responds exactly to the expert (or end user) needs. Those difficulties can be overcome by the implication of the user all along the DSS development. No approach is proposed in the literature for the KDD-based DSS development. So, in this work, we wanted to explore the following questions: (1) “which approach should be followed to build a KDD-based DSS?” (2) “Is it possible to take advantage of the existing approaches or methodologies?”

Human-Computer cooperation is essential to the decision support process. From this perspective, Human-Computer Interaction (HCI) is a fundamental aspect for interactive DSS, because the design of such systems heavily relies on a user-centered approach. Given this context, in order to accurately design a DSS, we believe that it is necessary to combine and integrate methods from Software Engineering (SE) and HCI. There already are a number of software development models available in

SE for improving software reliability, evolvability, reusability, and portability. However, these models are limited when the system is highly interactive because they do not take the user into account explicitly and systematically [26].

In this paper we propose an approach to develop a KDD-based DSS. This approach is based on two systems development methods—the Unified Process (UP) from SE and the U Model from HCI.

We developed and validated our approach with the help of physicians from the Intensive Care Unit (ICU) of Habib Bourguiba Teaching Hospital of Sfax, Tunisia. Indeed, medicine, and more precisely the ICU, is one of the fields where there is a great mass of data daily collected and where experts have to take important decisions. These physicians need a DSS based on KDD to help them predicting and preventing nosocomial infections, also called hospital infections. These infections are caught by some patients, a few days after their entrance to the hospital. Some of these patients may die because of nosocomial infections.

This article continues with a short review of the literature about the two fundamental concepts: DSS and KDD (section 2). Then, it presents a short survey of the most widespread development processes in the fields of SE and HCI in order to identify the key elements needed to implement a more complete process that supports DSS design and development based on a KDD process (section 3). It then summarizes this more complete approach (section 4) and provides a case study of its implementation in a medical environment (section 5). In section 6 we give a discussion about our approach, and the research and practice implications of this work. This article ends with our conclusions and the perspectives for future research (section 7).

## **2. DSS and HCI: Key concepts for KDD**

### **2.1. Decision-making process, DSS and HCI**

Simon [53] distinguishes three phases in the decision-making process [24, 39, 43, 57]: (1) the search for information, which includes the identification and the statement of the problem to deal with; (2) the design process, which includes generating, developing and analyzing the different possible series of actions (solutions); and (3) the choice of a solution, which involves researching, evaluating and selecting the most appropriate solution identified during the previous phase.

Beynon et al. [6] have shown that there are fundamental problems for the DSS process described above: (1) initially the expert doesn't know how to identify or formulate a problem; (2) the expert doesn't always have an abstract or systematic method for finding solutions, only recognizing solutions that have been seen or explored; (3) the expert may have no explicit heuristic for evaluating and selecting solutions. Beynon et al. [6] highlighted the DSS's need for synergy between human processing and "automatic" processing.

In the early eighties, several User Interface Management Systems (UIMS) [44] and many toolkits [40] were developed, as extensions of the graphics libraries. The fundamental principles of UIMSs are the re-use of interface code and the separation of the functional part of an application from its interface [40].

Fisher [14] has stated that the success or failure of a DSS depends more on its communication capabilities and its interface for human-computer interaction than on its processing speed and problem solving capabilities. We tend to agree with this assessment. We consider that interactive DSS are systems that allow one user (decision-maker) or a group of users to identify, explore and solve problems through human-computer dialogue, with the communication between the various human actors (e.g., decision-makers, consultants, experts) involved in a DSS [57].

The concept of interactivity in a DSS highlights the essential role of humans. This role is not passive and is facilitated by the integrative quality of the various system components and by the HCI, which takes on the supporting role of decision-maker's assistant [43]. We believe that system developers

must design cooperative DSS systems that permit the distribution of competencies between the user and the computer.

## 2.2. KDD: an interactive and iterative process

The concept of decisional computer science [58] refers to decision-making support, or in other words, to the exploitation of data to facilitate decision-making. There exist several decisional tools, such as: On-Line Analytical Processing (OLAP) [12, 56] for multidimensional analysis and Data Mining (DM) for exploring new knowledge in large quantities of data [13, 17]. DM aims to identify the *possible correlations* between data elements in a significant volume of data in order to highlight "hidden" trends. With DM the system often takes the initiative, discovering the associations among data by itself. To a certain extent, this makes it possible "to predict the future, according to the past".

Originally, DM referred to the most important stage of a multi-stage interactive and iterative process known as KDD (Knowledge Discovery from Databases) [31]; results can be refined by repeating the various stages several times under the control of an analyst (Figure 1). The various stages of the KDD process are: (1) problem formulation (identifying the objectives, defining the targets and checking the needs); (2) data retrieval (identifying the information and the sources, checking their quality and accessibility); (3) data selection (choosing the data related to the requested analysis); (4) data cleansing (detecting and correcting the inaccuracies and/or errors in the data); (5) data transformation (formatting the data in preparation for the "mining" operation); (6) data mining (applying one or more techniques—such as neuronal networks, Bayesiens networks or decision trees—to extract interesting patterns); (7) result evaluation (estimating the quality of the patterns extracted, interpreting the patterns' meaning); (8) knowledge integration (putting the pattern or its results into the company's information system).

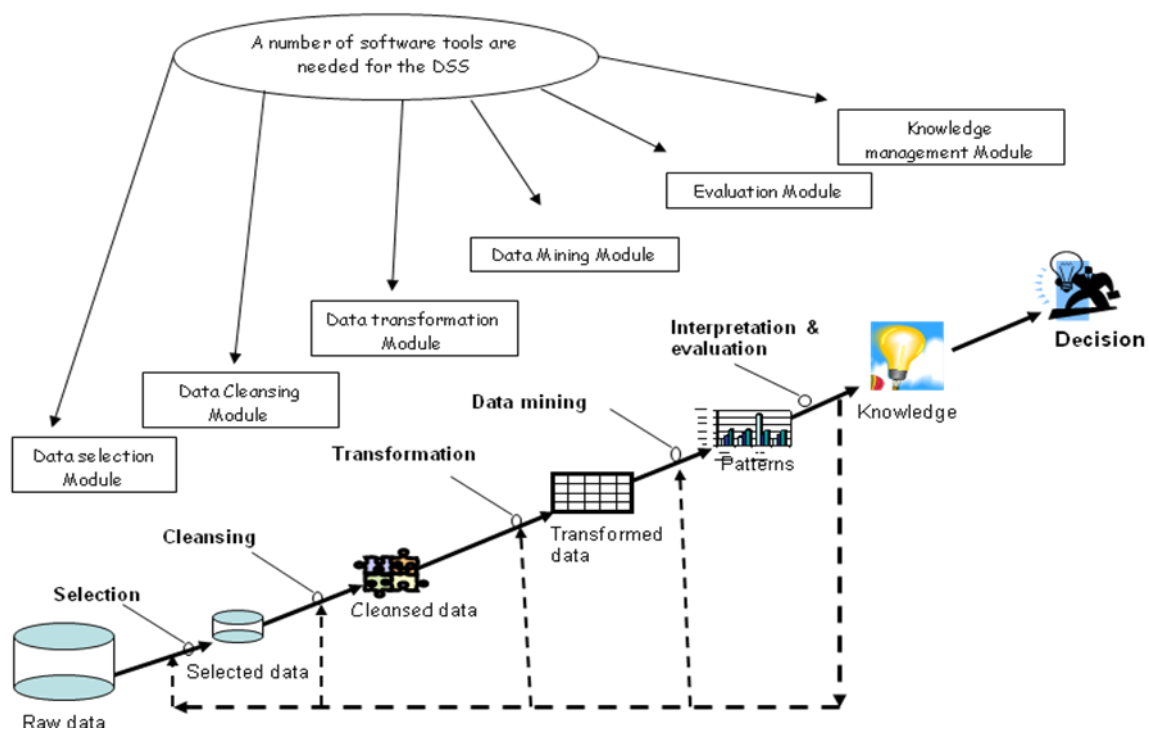


Figure 1: KDD tools

According on the above stages, a KDD-based DSS could require up to six modules for its implementation: (1) one to select data, (2) one to cleanse data, (3) one to transform them, (4) one to mine them, (5) one to evaluate and interpret the patterns extracted, and finally (6) one to manage the extracted knowledge. Some modules may contain a number of different applications. For example, the

data mining (DM) module could contain several applications, each one using a different technique to achieve different objectives. Other modules may be combined together. Some researchers have proposed combining the data selection, cleansing and transformation modules into a single module called "data preparation" [17, 41]. This is the direction that we have taken in our work (see section 5).

### 2.3. Integrating HCI in a KDD process

One of the challenges of KDD-based DSS is detecting the strategies for solving a decision problem through a process that has a DM stage. This process includes various important stages, such as analyzing the decision-makers' need, determining the various preparatory activities, handling the relevant data and visualizing the results. The end-users' acceptance or rejection of the decision support tool depends on these stages [31]. For this reason, appropriate HCI activities should be able to guide users throughout the stages of the KDD process; it is also important to adapt these HCI activities as much as possible to each decision-maker [28]. The next section describes some representative systems development models in the fields of the SE and the HCI.

## 3. Development approaches

Given the stakes involved in most DSS problems and the quantity of raw data used in KDD, it is essential to have the most complete model possible of the methodology. In order to select an appropriate global approach for our context, a critical study of some traditional development cycles was necessary [34]. This section provides a representative rather than exhaustive examination of the available systems development models in the fields of the SE and the HCI.

### 3.1. The principal SE development models

SE development processes or models describe the logical and temporal order of the software production stages. Several models have been proposed. The main ones are discussed briefly below. The Unified Process, which is based on a specific methodological process, is also examined.

The **waterfall model** [48] identifies the main steps in the sequential production and maintenance of an application. It allows returns to the immediately preceding stage but it does not take HCI into account, even when the system is highly interactive [25]. The **V model** [37] structures the process steps in two sub-processes: (1) a descending sub-process for specification and design and (2) an ascending sub-process for validation and testing. But, it does not introduce HCI considerations; it is mainly intended to evaluate the technical aspects of the system concerned. The **spiral model** [7] is based on a succession of 4-phase cycles, each of which is based on a systematic analysis of the risks related to the alternatives in the decision-making process during the project steps. It introduces the concepts of prototyping and refining. Nevertheless, like the other models, it does not take the HCI into considerations.

The **agile models** [30] for software development appeared in the early 1990s. These models do not have such systematic processes as the traditional models. Their objective is quick software design based on heavy customer involvement and significant reactivity to his requests. These models target customer satisfaction rather than satisfying the terms of the contract. They include, for example, Adaptive Software Development (ASD), Dynamic Systems Development Method (DSDM), Extreme Programming (XP) [23] and Rapid Application Development (RAD). 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 developers [11]

Jacobson and his colleagues, the fathers of the **Unified Process** (UP), do not cite any of the methods mentioned above in their reference book [21], nor do they use the term "agile". They thus apparently do not consider UP being an "agile" model. Nevertheless, we feel that its potential is very interesting.

Indeed, UP consists of a set of generic principles that can be adapted to specific projects [21, 29]. It 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. UP is (1) controlled by the use cases, representing the functional needs of the system, (2) centered on system architecture, which provides the structure used as framework for the work carried out during the iterations, and (3) iterative and incremental, with the aim of reducing complexity by controlling it, by breaking up a data-processing project into sub-projects that each represent one iteration. These iterations indicate the steps in the sequence of activities, while the increments correspond to the stages of product development. The aspects of the model being analyzed and designed are based on UML [21, 30, 50].

UP has 4 phases: (1) *initialization*, during which the extent of the project is defined through use cases and feasibility studies; (2) *development*, during which the needs are defined and the architecture specified; (3) *construction*, during which the software is built by means of several iterations and various system versions; and (4) *transition*, during which the system is delivered to the end-users and put into service and these end-users are trained and provided with tech support [21, 29] (Figure 2).

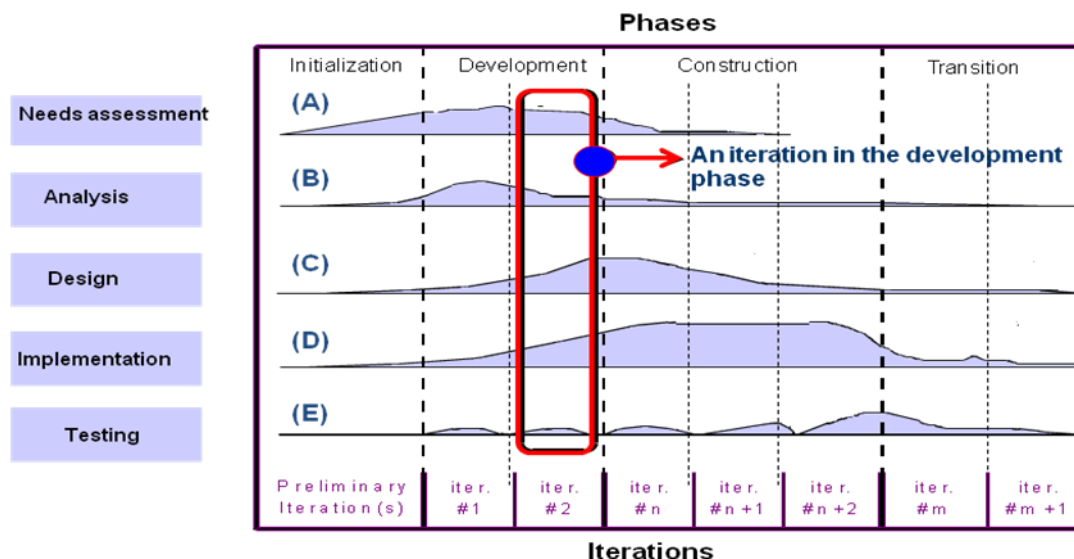


Figure 2: Unified Process [21]

The advantage of UP is that it : (i) allows the costs to be limited to the strict expenses (workload) related to an iteration, (ii) makes it possible to limit the risks of delaying the installation of the application to be developed, (iii) permits potential problems to be identified in the first stages of development. This short-term planning is due to the fact that user needs and the corresponding requirements cannot be defined completely in advance. The system architecture provides the structure that is used as a framework.

Lemieux and Desmarais [32] showed that the Rational Unified Process (RUP), a commercial version of the UP, is not user-centered according to the standard ISO 13407 [20]. This standard specifies the rules to be followed to adapt a software development process for user-centered design. They also note that the introduction of the use cases is not sufficient to make a design process user-centered.

SE is designed to produce quality systems. However, most 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. Nonetheless, though the user is involved throughout the

project life cycle, no formal explanation shows this involvement. The UP diagram (Figure 2) does not mention the user anywhere.

The general tendency in software development is towards iterative processes (e.g., spiral, UP) but also towards integrating the re-use concept. Even when 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. HCI design and evaluation principles are not part of these generic processes [27, 32]. From the perspective of interactive system development, user characteristics must be clearly expressed. The SE development models have been judged to take the user insufficiently into account, about fifteen years ago, the traditional models began being enriched with HCI principles. This enrichment is discussed in the following section.

### 3.2. The principal HCI-enriched development models

HCI-enriched development models are user-centered; their principal concern is to highlight the fundamental methodological aspects of the system, such as the modeling of human tasks, the iterative development based on prototypes and the evaluation of the interactive system. The best-known models are mentioned below (For a more complete review, see the article by Kolski et al. [27]).

The **star model** [19] places the evaluation at the center of the complete process, which permits interactions and iterations between each stage. This model is very flexible; it does not impose any order for the stages of the process. The star model offers a participative design process that targets the detection of usability problems [34]. This model's weakness is that the tasks are validated by only one prototype [10].

In the **HCI-enriched V model** proposed by Balbo [5], the steps of development process are divided into two zones: the HCI zone and the software zone. The first is characterized by the priority given to the ergonomic aspects; the second by the importance given to the software implementation techniques.

The **U model** [1, 3, 33, 38] considers the steps which do not exist in traditional SE models. The U model has two phases (Figure 3): (1) a descending phase for specification and human-machine systems design, which leads to its implementation; and (2) an ascending phase for the evaluation of the global system. Validation consists of comparing the model of the theoretical tasks specified in the descending phase with the model of the real tasks highlighted in ascending phase, according to the original principles suggested by Abed et al. [2, 3]. The result of this comparison either validates the human-machine system or highlights its deficiencies. In case of deficiencies, the result of the comparison has to be gradually fine-tuned, particularly on the level of HCI and the support tools. The final model resulting from the assessment allows the users' specific behaviors to be generalized under particular work conditions, and can be reused in situations dealing with similar systems [34].

The U model can be adapted according to the characteristics of the application. For example, Lepreux et al. [33] adapted this model for particular types of DSS. This model is centered on HCI, but it does not clearly present the iterative and incremental development of the interactive system to be created.



of these traditional models are still insufficient for designing and evaluating the DSS that interests us because the designer does not have to cover all the steps of the traditional development process. For this reason, we propose a new design approach that combines an SE process with an HCI model. Since our approach reconsiders the **UP** in the light of the **U** model, we call it **UP/U**. We think this new approach is more appropriate to our DSS context.

#### 4.1. Justification of the adopted methodology

The critical study of the development processes in the fields of SE and HCI, presented in section 3, allowed us to highlight the advantages and disadvantages of the various models. Interactive systems design as practiced in HCI and the development processes as performed in SE are generally executed separately and thus lack coherency. In order for the two fields to work together, it is essential to discover the commonalities of SE and HCI practice. To accomplish this, we compare the UP from SE and the U model from HCI. In our opinion, these two standards are complementary. Table 1 presents the elements that, in our opinion, justify this choice.

Table 1: The complementarity of the Unified Process and the U model

THE UNIFIED PROCESS <i>Methodology from SE</i>	THE U MODEL <i>Methodology from HCI</i>
<b>Defining characteristic:</b> Generic: this process can be adapted to a large class of software systems, various fields of application, various ability (or qualification) levels and various project sizes.	<b>Defining characteristic:</b> Specific to interactive applications: this model has its roots in the various traditional HCI-enriched models
<b>Basic principles:</b> Iterative and incremental; based on use cases, and thus incidentally on user needs; centered on the system architecture rather than the system user	<b>Basic principles:</b> User centered, meaning that human factors are taken into account by the development team; comparison of the theoretical and real tasks in order to validate and fine-tune the system
<b>Relevance:</b> More than the advantages offered by this methodology (e.g., adaptability, control of projects complexity), the strong point is <b>the interactivity</b> of its process. <u>Advantages:</u> - Minimizes the costs, in terms of risk, to the cost of only one iteration. - Thanks to the iterative nature of the development, user needs appear more clearly with each successive iteration, making it easier to adapt to the user's evolving needs.	<b>Relevance:</b> The methodology's <b>interactivity</b> is fundamental from the perspective of DSS design. In addition, the analysis of the system users and their tasks is recommended. <u>Advantages:</u> - A set of tasks is associated to each user objective. This allows certain interactive system functions to be defined. - The designer becomes more aware of the importance of the human factor in HCI design and evaluation.
<b>Life Cycle:</b> - UP iterates a series of cycles that constitute the construction of a system generation. All cycles end with the delivery of a product version to the customers. - Each UP cycle has 4 phases: initialization, development, construction and transition. Each of these phases is in turn divided into iterations. Each iteration includes five activities: needs assessment, analysis, design, implementation and testing.	<b>Life Cycle:</b> The U model has 2 phases: - A descending phase that includes the specification and design of human-machine systems. - An ascending phase that includes the evaluation of the global system. In turn, the first phase can be divided into 4 stages: needs assessment, analysis, design and implementation; the second phase can be divided into the activities of evaluation and comparison.

As the table shows, both methodologies have elements that are relevant for the development of a DSS based on a KDD process. The development approach that we propose must make it possible to take the user into account (the contribution of the U model) by emphasizing the use of prototypes, the explicit positioning of the activities of the development process actors and the analysis of the activity (the contribution of UP).

To model the DSS that interests us, we used the Unified Modeling Language (UML) [29, 49]. Indeed, UP (Figure 2) offers a generic methodological framework based on UML [21]. Given that UML does not impose a specific working method, it can be used transparently with any other software development process, making it possible to use this language with the U model (Figure 3), which does not define an obligatory modeling language.

Our objective is to design a DSS that makes it possible for decision-makers to interact with the system to get information to support their decision-making process. In the problem identification stage of the KDD process, the different main objectives of the future system are determined and defined. In the pre-treatment stages, specific databases are built, the data is cleansed, any missing data is processed, data attributes or elements are selected and then transformed that it can be used by a DM algorithm. These stages are crucial for the retrieval of information relevant to the decision-making process [57]. DM can then be performed to provide knowledge in the form of models that must be validated. Post-processing is also necessary to make these models understandable to humans and/or computers [57]. Information visualization can help the user (decider) to get and understand information efficiently and implicate him/her in the data mining process thanks to his/her perception possibilities [35]. For this reason, several possible solutions to the studied problem need to be developed based on the knowledge discovered through KDD. Figure 4 shows how a DSS can be based on a KDD process.

As pointed out above the U-model must be adapted in such way that it can be used in our approach.

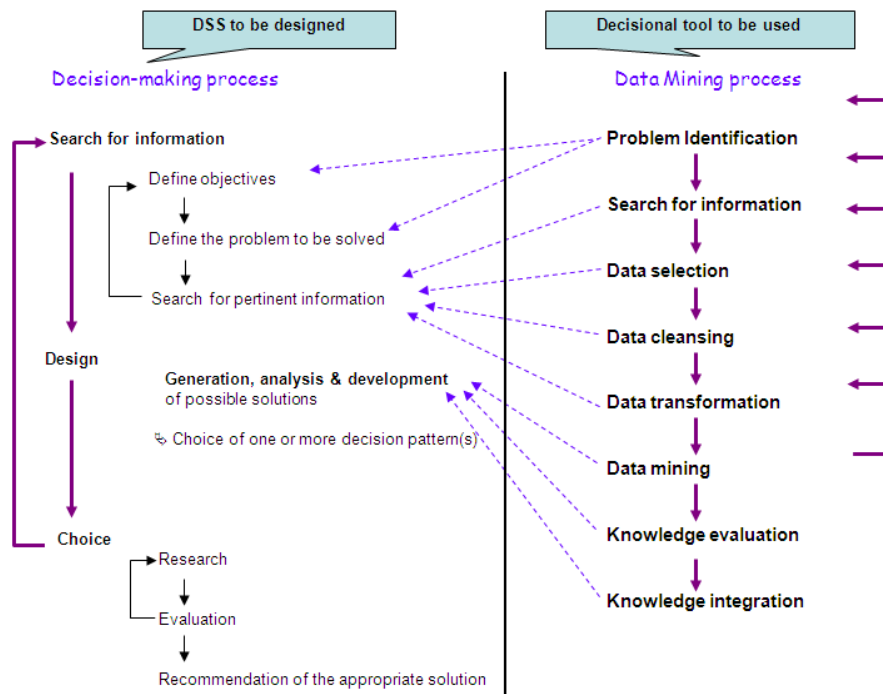


Figure 4: DSS based on KDD

## 4.2. Adaptation of the U model

In this paragraph we will explain the reasons for the adaptation of the U model.

Indeed, several elements of this model could be better adapted to the DSS context. Thus, when the existing system is studied, it is important for the decision-maker (the potential user) to describe his/her functional needs and to evaluate and validate the preliminary interface models to show how he/she wants to see HCI executed in the upcoming application. All this information can be used "to model" the decision-maker (e.g., characteristics, preferences, strategies) [46]. In addition, since task definition and task allocation in a KDD-based DSS are very important, these activities must be singled out for attention. Furthermore, the original U model does not clearly present the order of the activities (needs assessments, analysis, design, implementation and testing); so, we proposed an order for these activities needs to be established as the U model is to be used with UP. Moreover, since the steps of U model are more concerned with the specification and evaluation of HCI than with the applications of the interactive system studied, this aspect also needs to be reinforced for the KDD-based DSS. Finally, we think that the "general knowledge capitalization model" step could be simplified by getting feedback from the "user model" step. Based on this list of elements, we propose an adapted U model for DSS design and development (Figure 5). Its description is indicated in the following paragraph.

**Descending phase:** the beginning of the descending phase starts with two essential steps that take place simultaneously: (1) the analysis of the application domain, including defining the system objectives, which allows the first functional and structural description of the system to be developed; and (2) the development of the first interface prototypes (models) for the DSS in question, which, by giving future users an idea of the possible solutions, allows them to be implicated as early as possible in the project life cycle. These two steps provide a structural framework for future activities and technical solutions. Once these steps accomplished, it is possible to draw up a list of the tasks (manual, automatic or interactive) needed for the future operations of the DSS. These tasks must be defined and formalized. These steps can take place cyclically, as suggested by the arrows in the figure of the adapted U model (Figure 5). Then, the specified tasks must be modeled [2]. Three principal task categories are customary in SE and HCI: (1) tasks involving the user only, called manual tasks, (2) tasks involving the application only, called automatic or system tasks, and (3) tasks involving different levels of collaboration between the user and the system, called interactive tasks. The probable behaviors of the various user types serve as the basis of these representations, taking the form of a general user model. The limitations and physical and cognitive resources of the various types of users are incorporated into the model [46, 59].

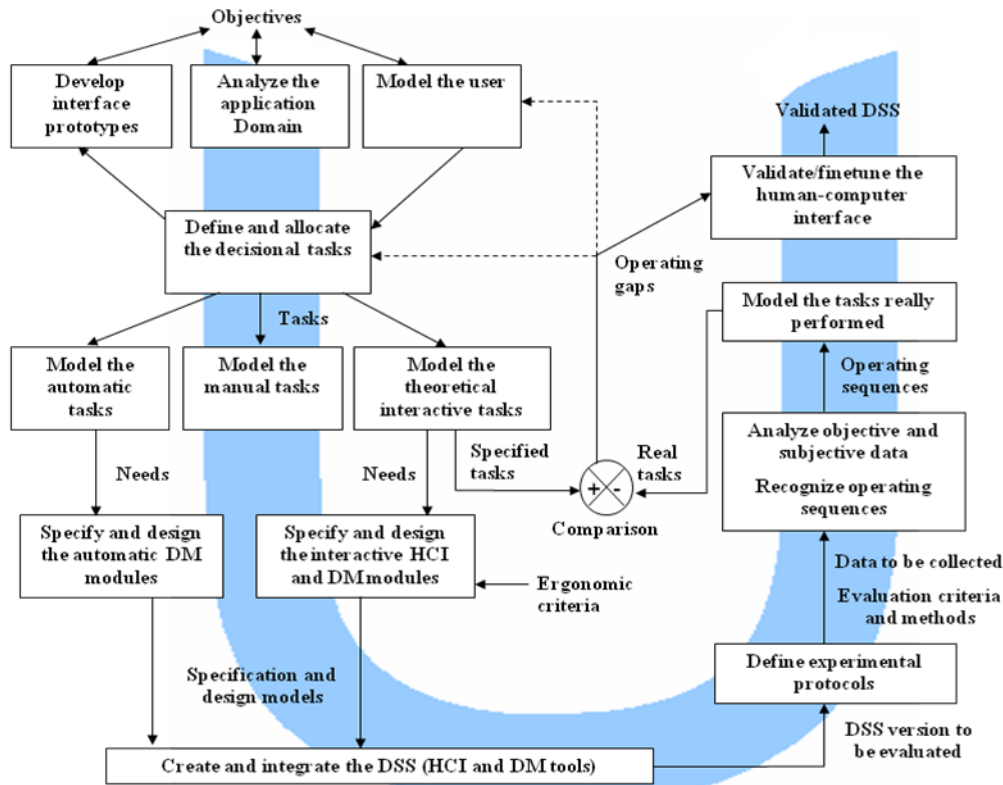


Figure 5: Adapted U model

Depending on the specificities of the domain, the various decision support tools must be analyzed in order to determine those that are the most appropriate for the system to be designed. The last step of the descending phase leads to the implementation of the DSS module or one of its versions.

**Ascending phase:** the evaluation of a human-machine system consists of testing whether or not users can accomplish their tasks using the provided interface. Two properties are usually explored for such evaluations: usefulness and usability [42, 51]. A number of methods can be used to evaluate interactive systems [42, 60]: interviews, questionnaires, observations, electronic informers, trace analysis. The ascending phase generally concerns the tasks shown on the right-hand side concentrating on: (1) user behavior when interacting with the system (e.g., the time required to accomplish a task, the accuracy of the results, the number and types of errors, user opinions, etc.), and (2) the differences between the system objectives and the final results.

The ascending phase starts with the definition of the experimental protocols (i.e., subjects, progress, situations and tasks concerned, data to be collected, etc.) [3]. Once collected, the data is processed according to the operational principles established in the descending phase in order to correlate the data with the observed human activity for the whole decision-making process shown in Figure 4. The operating sequences are highlighted, which allows the real activities (tasks) to be gradually reconstituted. The tasks really accomplished can be different from the tasks initially specified by the designer in the descending phase. One of the fundamental principles of the U model concerns the direct comparison between the real and the specified tasks. The results of this comparison lead either to validate the entire system or to identify the areas that need to be improved. Depending on the extent of the improvements needed, implementing these improvements may require returning to the various steps of the descending phase, which is perfectly coherent with the iterative nature of the UP.

#### 4.3. The proposed user-centered approach (UP/U)

Since it is intended to allow KDD-based DSS to be designed, our approach puts HCI in a central position, redefining the user's role allowing him/her to intervene at any time in the KDD process

(Figure 6). This approach is based on the UP principle of iterative and incremental development, which allows each task accomplished to be evaluated as soon as the first iterations of the development process have been completed.

Our approach executes several complete UP iterations, from the initialization phase to the transition phase [36]. However, since the five activities of the original UP process (needs assessment, analysis, design, implementation and testing) do not model the system users or system-user interaction, our adapted U model is applied at each iteration level. Our approach thus 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) [46], "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 components 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.

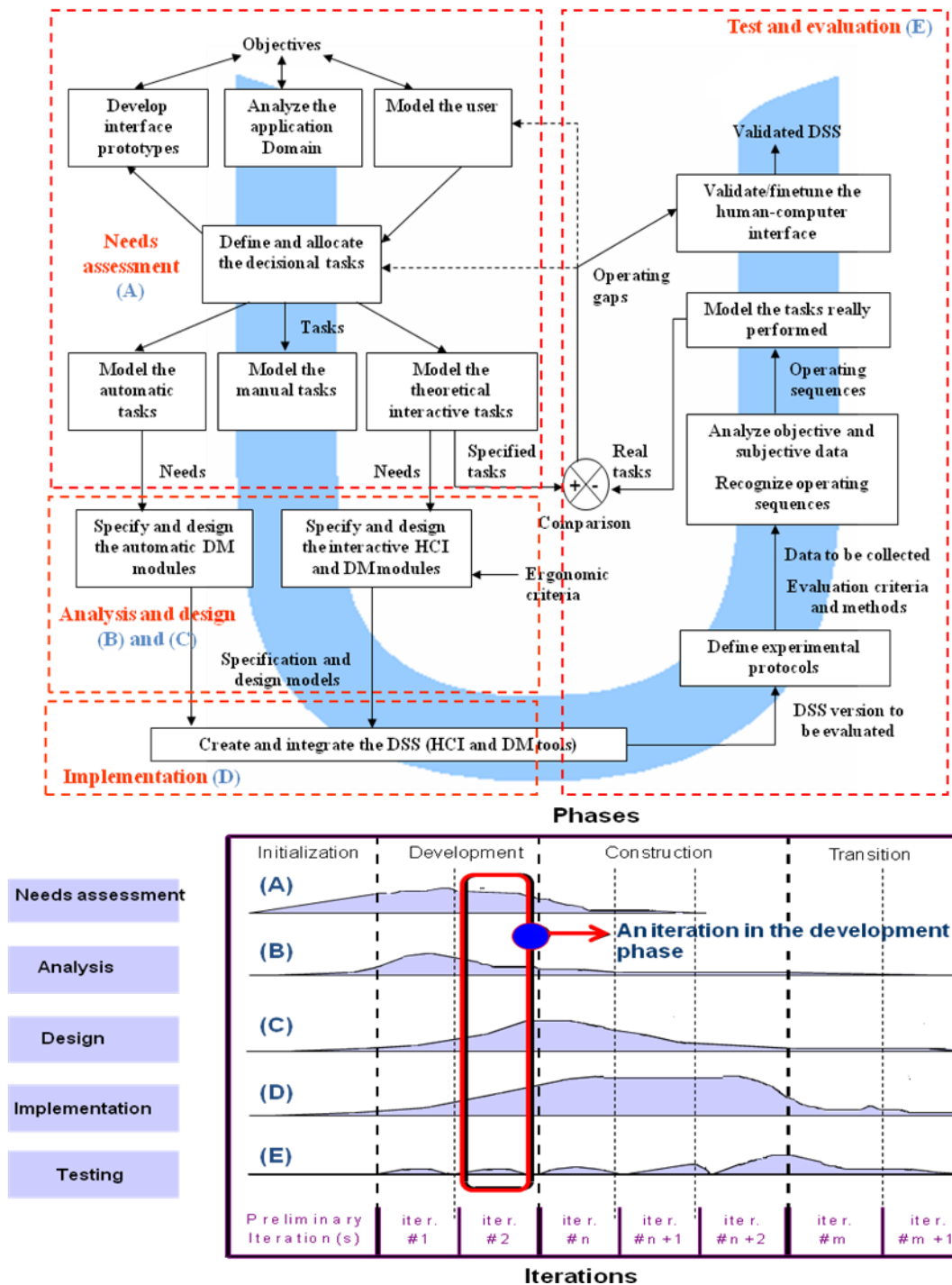


Figure 6: The UP/U approach

## 5. Healthcare case study

In the previous section, we presented our proposal for a user-centered approach. This approach was applied to a concrete case in the field of healthcare. The DSS discussed below is based on a KDD process in which the data mining stage uses Case-based Reasoning [45]. This DSS is being used and evaluated in the intensive care unit (ICU) of the Habib Bourguiba Teaching Hospital (HBTH) in Sfax, Tunisia. The DSS was designed to help the physicians, who are the current users of the system, to

understand, predict and prevent nosocomial infections (NI). In this section, we first describe the context. Then we explain how the proposed user-centered approach was implemented in this context.

### 5.1. Context

Our project is part of a much wider project aimed at fighting NI. They are infections contracted by patients during their hospital stay. It is a major public health problem. An infection is typically regarded as nosocomial if it appears 48 hours or more after hospital admission [16]. NI can be located in any organ; the most commonly affected organs are lungs, kidneys and the heart. In ICU, the problem of NI is far more alarming because the patients who are hospitalized in that unit are more fragile. It is such an important problem that some European countries have created groups to fight NI [54]. A few studies have proposed information-processing systems based on data mining techniques to monitor NI [8, 9]. Some of these studies have shown the effectiveness of these systems and their capacity to produce useful rules. But, as described in the articles, these systems appear to be difficult for physicians to use.

A study on the prevalence of NI in the HBTH [22] showed that 17.9 % of the 280 patients hospitalized in the entire hospital were victims of one NI between April 17th 2002 (midnight) and April 18th 2002 (midnight). The ICU of the HBTH has 22 beds in 11 compartments. The patients admitted to ICU are known as "critical" because they require much care and constant monitoring. These patients are often connected to machines (e.g., artificial respirator, electrocardiogram, electrical syringe) and/or have catheters (e.g., venous catheters, urinary probes, thoracic drains). Extremely fragile, these patients are sensitive to every new germ that enters their bodies, which makes them more likely to develop NI. When an infection (nosocomial or not) appears, a sample is sent to the laboratory to get an anti-biogram. Depending on the result of the anti-biogram, anti-biotherapy is prescribed. The problem with anti-biotherapy is that a germ can be sensitive to an antibiotic one week and resistant a few weeks or a month later. In addition, this sensitivity can be different from one patient to another.

Several doctoral dissertations and studies have been published on this subject by the team of ICU physicians at HBTH [15, 18]. Previously, to conduct these studies, doctors collected their data manually on paper forms and then entered and stored the data in an Excel® file, which was analyzed by statistics software such as SPSS®. However, such tools are only capable of producing the "classic" statistics (e.g., percentages, averages, variance analyses). The physicians wanted go further in their analysis in order to extract information and knowledge that would allow them to better understand, predict and prevent the appearance of NI. We got involved because data mining can provide them with that knowledge.

### 5.2. KDD process modules

Figure 1 shows the six modules that it would, in theory, be necessary to develop to create a DSS based on a complete KDD process. In reality, depending on the specific details of the context, the number of modules can vary. To develop our DSS, since we did not have initial data sources, we had to create one especially for the project. Consequently, the modules "data selection", "data pre-treatment" and "data transformation" could be combined to form one module "data storage and acquisition". As a result of this fusion, only four modules were necessary: "Data acquisition and storage", "Data mining", "Evaluation" and "Knowledge management".

This development project should last several more years. At the time that this article was written, our user-centered, iterative, incremental process had resulted in: (1) the creation of a database, (2) the development of a data acquisition application and (3) the development of two data mining applications. The evaluation module is now under development [4]. The knowledge management module is expected to be brought under development soon. The various phases of our approach are described in more detail in the following section.

### 5.3. Implementation

In this section, we first describe the activities undertaken to create the first interactive KDD module in order to show how our approach can be implemented. Next, we provide a general explanation for one of the 2 data mining modules, illustrated by a screen shot of the interface of this data mining tool.

#### 5.3.a. Implementation process of "storage and data acquisition" module

In this section, the progress of our approach towards the creation of the "Data acquisition and storage" module is described in 4 tables, one for each phase (initialization, development, construction and transition). The activities related to UP are presented on the lines with a grey background, and those related to the U model appear on the lines with a white background.

The **initialization phase** activities are presented in Table 2. During this phase, which has only one iteration, we mainly studied the context and defined the objectives through regular meetings with the doctor dealing with the problem of nosocomial infections (NI), who played the double role of decision-maker and user. Sometimes his associates attended the meetings. The tasks were defined and allocated with his total cooperation. During this phase, we also studied the NI that constituted the context of the project. Only a first draft of the database set was concerned by the implementation. During this phase, no applications or interfaces were proposed to the user.

Table 2: Initialization phase for the "storage and data acquisition" module

Initialization	
Activities	Iteration 1 (only one iteration in this phase)
UP: Needs assessment	Need for data to be used by a data mining tool. A database needed to be implemented containing data useful for studying NI. Data should be recorded after the patient leaves the ICU. The persons in charge placed a computer with limited capacities at our disposal for certain period of time
U: Analysis of the application domain	Study of Nosocomial Infections (definition, causes, risks...)
U: First interface prototypes. Description of the system's functional and structural needs and task definition.	Structural and functional description was deduced after consultation with the future users and the general architecture of HCI prototypes was outlined. The doctors proposed windows with tabs.
U: User design	A preliminary user model (only one type for the moment) was created. This user was an expert in the NI field, as witnessed by his many publications and his supervision of doctoral dissertations about the fight against NI. He also had good computer skills, spending more than 3 hours per day in front of his PC.
U: Definition and allocation of tasks to be executed	The first description of manual tasks (Cards containing the data necessary to the study are filled out by the internal doctors at the patients' bedside), automatic tasks (calculation of the patient's age and death risk parameter) and interactive tasks (choice of the titles of the data entry zones) was done using the MAD task model [50].
UP: Analysis and design	A first analysis consisted of studying the way in which studies are now carried out (manually). To facilitate the design process, a description of the data useful for the study and an outline of a Entity/Association model were generated.
U: Specification and design of automatic modules	A first specification of calculation modules (age, duration of stay, risk of death) and of the transactions with the DBMS was carried out.
U: Specification and design of interactive modules	A first specification of interactive modules was done using UML sequence diagrams and the MAD task model.
UP: Implementation of the application	The implementation concerned only the database. The DBMS Oracle ® and its tools (SQLplus...) were used.
U: HCI implementation	<i>N/A in this iteration (no application intended for the user has been developed yet)</i>
UP: Technical tests	<i>N/A in this iteration</i>
U: User-centered Tests	<i>N/A in this iteration</i>

The iterations of the **development phase** (Table 3) are often the most numerous and most charged. During this phase, the user expressed new needs. The first tests of the preliminary versions were very useful for the remainder of the development process. The interest the user showed for our approach allowed him to clearly express the evolutions of his objectives. As a result of his evolving objectives, we developed several versions, one for each of the three iterations.

Table 3: Development phase for the "storage and data acquisition" module

Development			
Activities	Iteration 1	Iteration 2	Iteration 3
UP Needs assessment	At the beginning, we were asked to store only the patient diagnoses. It was necessary to refine this information to indicate the type of pathology associated to the diagnosis. This would make it possible to carry out analyses on a particular type of pathology and also to compare the appearance of NI according to various pathology types.	Following the first tests, the users wanted other functions (patient searches, data modification and/or suppression). Moreover, it was not always possible for us to work at the hospital, so we needed to transfer the data to the laboratory. With Oracle, data transfers were difficult and complex. For this reason, we migrated towards the DBMS SQL Server.	As the data are recorded after the patients leave ICU, it is useless to record the date each time. A button for "next Day" and for "previous Day" would facilitate the data entry. Further more, filling out the forms created an additional workload for the interns. To compensate for this increased workload, our interlocutor asked us to offer interns full patient reports to reduce their workload by allowing them to avoid writing exit reports
U: Analysis of the application domain	Thorough study of NI, especially with regard to the infectious examinations and the antibiotic regulations.	Further study of international and national researches about NI.	<i>N/A in this iteration.</i>
U: First interface prototypes. Description of the system's functional and structural needs and task definition.	No change on the level of the initial architecture of the proposed HCI.	Addition of the possibility of searching for a patient in the database and of updating some data .	Modifications of the HCI: check boxes or radio buttons to facilitate the data entry, to avoid errors and to make the data coherent, especially the values on which certain statistical tests are based. Addition of a "Report" button for launching a procedure recapitulating the patient's history of hospitalization
U: User design	The user understood UP principles very well. He found it advantageous to be able to express other needs and to propose modifications over the course of the project. He was, in fact, an active participant in the approach.	<i>N/A in this iteration</i>	<i>N/A in this iteration</i>

Table 3: Development phase for the "storage and data acquisition" module (*continued*)

Development			
Activities	Iteration 1	Iteration 2	Iteration 3
U: Definition and allocation of tasks to be executed	The thorough analysis of the NI domain allowed us to refine the definition of the tasks to be executed by the user/decision-maker); the MAD task model was thus adapted to reflect this finetuning.	Addition of tasks: display of the patients concerned when the user types a name, an index (administrative code) or a birth date; double clicking on the line corresponding to a patient results in a display of his/her contact information.	Addition of a selection task for the "Next Day" or "Previous Day", a simplification of a task that previously required 11 mouse clicks. The entry of the date with the keyboard is still possible.
UP: analysis and design	A first use case diagram was set up, followed by other UML diagrams.	Modification of the diagrams to reflect the additional tasks.	Modification of the diagrams according to that
U: Specification and design of the automatic modules	Use of UML diagrams for specifying and designing the application	Addition of the necessary procedures to display the patient data corresponding to the selection criteria.	<i>N/A in this iteration</i>
U: Specification and design of the interactive modules	The sequence diagrams, the models obtained with the MAD task model in the initialization phase, as well as the first paper drafts produced with the users, enabled us to specify and design the HCI.	Modification of the diagrams to reflect the additional tasks.	<i>N/A in this iteration</i>
UP: Implementation of the application	Implementation of the first data entry application	Implementation of a new DB using SQL Server and a new data entry application using C#.net	Implementation of a new version for entering daily information
U: Implementation of the HCI	Implementation of the data entry HCI.	Where it is possible, the data entry zones were replaced by radio buttons or check boxes.	Modification of the data entry HCI by adding the buttons: "Next Day" and "Previous Day".
UP: Technical tests	Tests of different types of transactions in the DBMS with SQL as well as in Oracle ® with Java application	Tests of the transactions with the new DBMS SQL Server (in place of Oracle).	Tests of the new version.
U: User-centered Tests	Execution of user tests. Results: description of the usability problems related to data entry, detection of other needs for the doctors.	Execution of user tests. Results: tasks accomplished as expected, though with some HCI overlap.	Execution of user tests related to the dates. Results: simplifications of the data entry procedures

The **construction phase** (Table 4) is principally made up of "Implementation" and "Testing" activities. Two iterations were sufficient to obtain an almost final version. The tests of the technical and functional aspects were performed with the expert user (the doctor).

Table 4: Construction phase for the "storage and data acquisition" module

<b>Construction</b>		
<b>Activities</b>	<b>Iteration 1</b>	<b>Iteration 2</b>
UP Needs Assessment	<i>N/A in this iteration</i>	<i>N/A in this iteration</i>
U: Analysis of the appl. domain	<i>N/A in this iteration</i>	<i>N/A in this iteration</i>
U: First interface prototypes..	<i>N/A in this iteration</i>	<i>N/A in this iteration</i>
U: User design	<i>N/A in this iteration</i>	<i>N/A in this iteration</i>
U: Def. and alloc. of tasks to execute	<i>N/A in this iteration</i>	<i>N/A in this iteration</i>
UP: Analysis and design	<i>N/A in this iteration</i>	<i>N/A in this iteration</i>
U: Specification and design of the automatic modules	<i>N/A in this iteration</i>	<i>N/A in this iteration</i>
U: Specification and design of the interactive modules	<i>N/A in this iteration</i>	<i>N/A in this iteration</i>
UP: Implementation of the application	Complete Implementation of the application.	The last bugs linked to the application were corrected.
U: Implementation of the HCI	Implementation of the various HCI, taking all the user's remarks into account, especially concerning the addition of explanatory icons associated to the various buttons (+, Magnifying glass, Stop...).	The last bugs connected to the latest user tests were corrected.
UP: Technical tests	Tests of the application in laboratory with fictitious data.	Tests of the final version.
U: User-centered Tests	Execution of user tests comparing the specified tasks and the tasks really carried out by the user. Result: overall success, some remaining usability problems were highlighted.	Execution of user tests verifying the interface proposed had the characteristics of a quality interface (Coherence, error prevention...) and a final validation.

During the **transition phase** (Table 5), we corrected some residual defects. The application now being used, with the data recorded in the database being exploited by both of DM modules (see 5.3.b).

Table 5: Transition phase for the "storage and data acquisition" module

<b>Transition</b>	
<b>Activities</b>	<b>Iteration 1 (only one iteration)</b>
UP Needs assessment	<i>N/A in this iteration</i>
U: Analysis of the application domain	<i>N/A in this iteration</i>
U: First interface prototypes.	<i>N/A in this phase</i>
U: User design	<i>N/A in this iteration</i>
U: Definition and allocation of tasks to be executed	<i>N/A in this iteration</i>
UP: Analysis and design	<i>N/A in this iteration</i>
U: Specification and design of the automatic modules	<i>N/A in this iteration</i>
U: Specification and design of the interactive modules	<i>N/A in this iteration</i>
UP: Implementation of the application	<i>N/A in this iteration</i>
U: Implementation of the HCI	Some slight improvements to the HCI were made (texts displays).
UP: Technical tests	Test of the last version carried out with ICU patient data transcribed on the provided forms: no error detected.
U: User-centered Tests	The user expressed his satisfaction and acceptance of the final module "data acquisition and storage".

Figure 7 provides an idea of the significant changes made to the HCI over the course of the project. Figure 7(a) shows the HCI of the very first version, and Figure 7(b) shows the HCI of an advanced version of the data acquisition application.

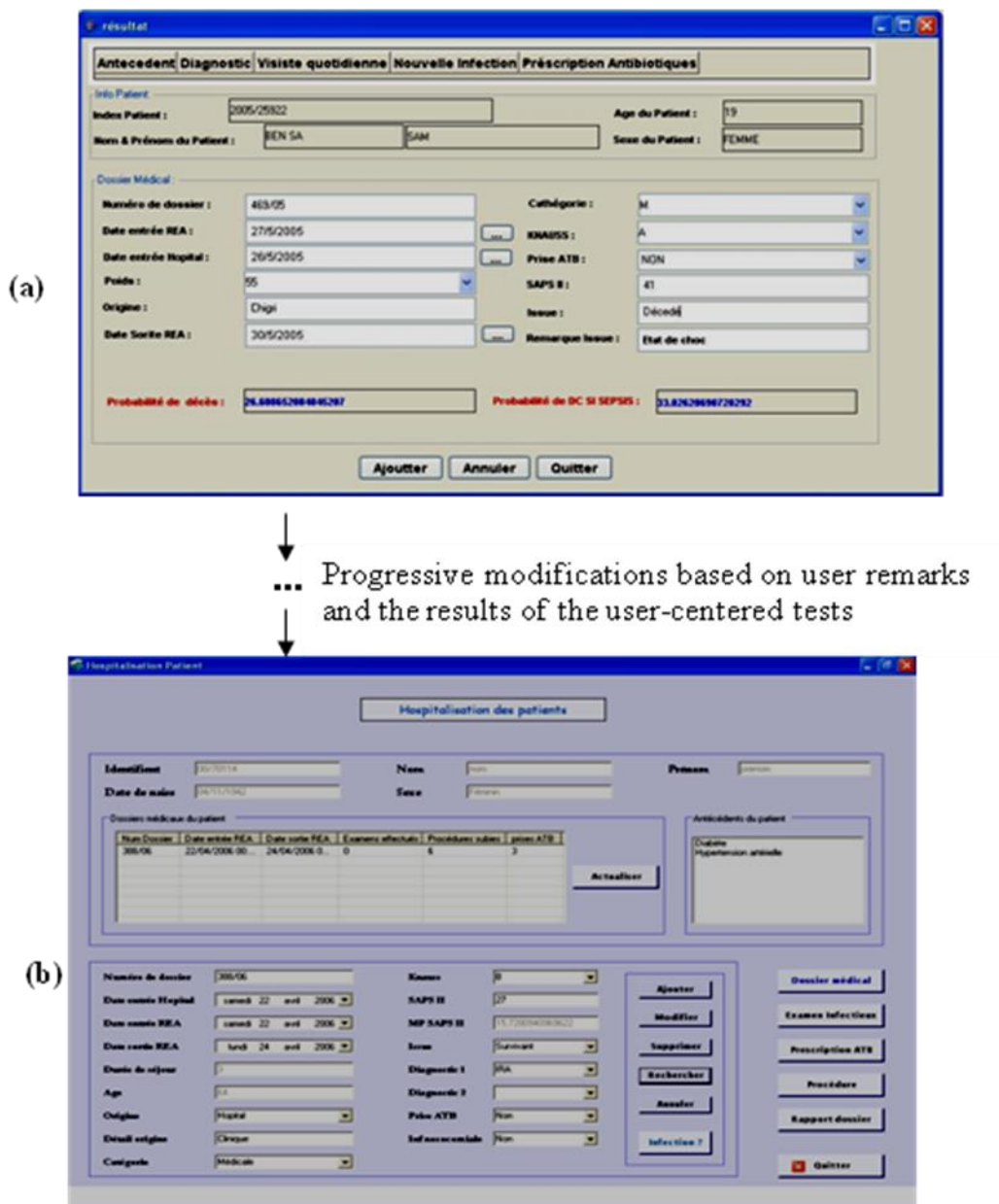


Figure 7. Screen shot of the patient data HCI: (a) from the development phase (iteration 1); (b) from the construction phase (iteration 2)

### 5.3.b. The data mining module

Once the module of "data acquisition and Storage" was realized, we started developing the "data mining" module. The initial objective of this module was to help the doctor predict the probability of NI appearance for a patient entering ICU. We developed two applications for data mining based on the needs expressed by the expert users: (1) an application for predicting the appearance of an NI, based on the KNN technique [45] and (2) an application for calculating, displaying and printing out

the status of NI outbreaks in ICU (Figure 8). As for the previous module, this one was designed and implemented according to the user-centered UP/U approach.<sup>1</sup>

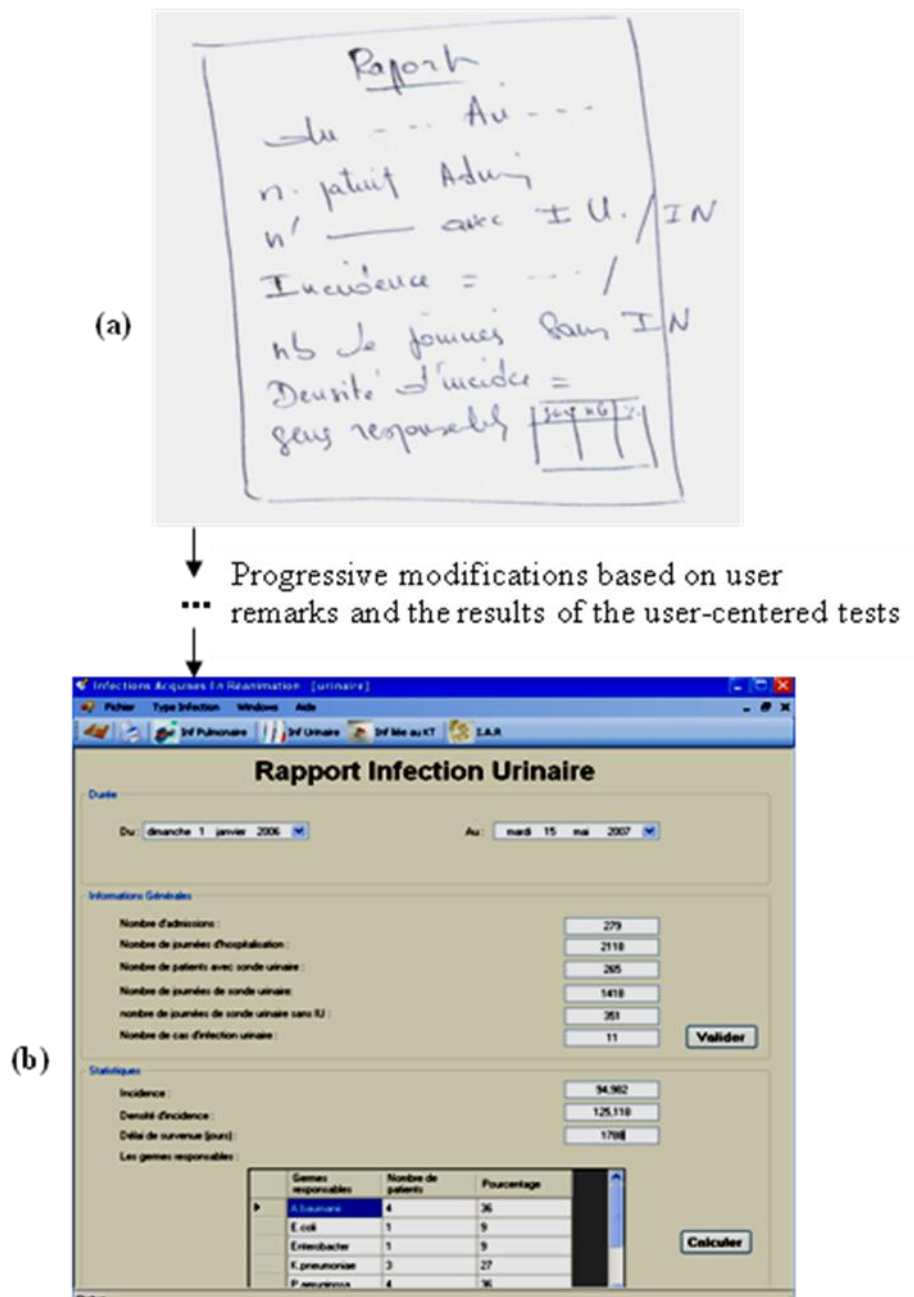


Figure 8: Extracts of the HCI of the second data mining tool: (a) from the initialization phase (iteration 1); (b) from the construction phase (iteration 1)

While the prediction module was being implemented, the needs evolved significantly: the original objective was to predict a possible NI outbreak when the patient was admitted or the day after. This objective was extended to allow the prediction to be calculated every day throughout the hospital stay due to the fact that patients see their state worsening or improving from the perspective of the risk of catching a NI during their stay in the ICU. The second "data mining" application involves statistical calculations. The objective was to make statistical studies of NI for each organ. As for all the DSS modules, the doctors showed a great interest for this type of studies. Figure 8(a) shows a handwritten

<sup>1</sup> By lack of place, it was not possible to detail each phase like in §5.3.a, but this description is available.

description (paper draft) of the way the doctor wanted to obtain the results; Figure 8(b) shows the HCI created by taking his needs into account.

## 6. Discussion

In this work we showed that KDD is a process that helps experts to get knowledge that supports them to choose the “right” decision. DSS and KDD converge in the fact that DSS can be based on KDD process. We have shown in section 4.1 and figure 4 that a DSS can be based on KDD. A DSS is naturally interactive [6]; and according to Fayyad and al. [13], a KDD process should be interactive too. But when we studied the way they have to be set up, according to the authors, we noticed that they propose stages in the process development (see section 2.1 and 2.2). The stages, indicated for both DSS process and KDD process, start by the definition of the problem and the objectives; and end by the choice of the decision, for the DSS process, and the use of the extracted knowledge to help choosing decisions, in the KDD process. As far as we know, no one has proposed any approach to develop a KDD-based DSS.

The motivation of this work is to propose an approach where the user is the principal actor all along the development period. So our approach must be a model that guides the developer to build up the system by respecting the interactivity and the iterativity qualities. In the literature we can find many models either for SE development, (such as the waterfall model [48], the V model [37], the spiral model [7] and the UP) or enriched under the HCI angle (such as the star model [19], the HCI-enriched V model [5] and the U model [1, 3]). But none of these models is adapted to develop a KDD-based DSS.

By studying the SE model we found that UP is the most appropriate one for iterative and interactive process; but as it is not user centered, it has to be associated with a HCI model. Among the HCI-enriched development models, the U model is the one that imposes to the development team to take the human actor into consideration and it permits to make assessments cyclically; the tasks are then validated for each system prototype. Thus, as U model is generic to develop user centered applications, it has to be adapted to the DSS context. The adaptation is also made to allow the U model to be used in association with the UP model.

As a result, to build a KDD-based DSS, we proposed a new user centered approach (based on the use of the U model) by underlining the use of prototypes, the explicit positioning of the activities of the development process actors (lying on the UP). Our approach is then called UP/U. In the following paragraphs, we will highlight the main implications for research and practice of our work.

The most important research contribution of this work is the proposed approach (UP/U) to develop a KDD-based DSS. UP/U is a generic approach that helps data miners to develop systems by involving the end user all along the KDD-based DSS process. Usually, Data Miners think about the efficiency of their algorithm, more than about the ease of use for the end user. Our approach incites the developer to build up a user centered DSS and to evaluate, with him/her, the tasks he/she achieves from the beginning to the end of the development process. A DSS which is difficult to use is generally dropped out by the user [31].

On addition, in many cases, the user either couldn't think of all his/her needs or discover some needs during the project development. Then developer has to be very often near by the end user (or users) to: (1) detect those new needs and integrate them into the system, (2) implicate him/her all along the development process (so at the end he/she find himself/herself familiar with the application), (3) guarantee that the system evaluation (of utility and usability) is made from the beginning of the project. To make this choice we had to accomplish a critical study survey about the development models in the fields of SE and HCI fields. This survey allows us to note that no model associates both advantages of those of SE and HCI models. The survey conclusion led us to choose the Unified Process and the U model to build up our UP/U approach. Another important implication is the adaptation of the U model to our context. Thereby, we made it possible for the potential user to describe his/her functional needs and to evaluate and validate the different interfaces.

As a practical implication, we can note the development of a KDD-based DSS using our UP/U approach in a real context which is the fight against nosocomial infections. During this project we were able to involve physicians throughout the development of the KDD based DSS. Indeed, as we have detailed in Tables 2 to 5, physicians were involved from the beginning to the end of the development process. The palpable implication for practice of this study is the final product set up in the ICU of HBTH of Sfax. Indeed, nowadays the KDD-based DSS is set in that ICU and it is used to explain and prevent nosocomial infections. Each patient entering the ICU, has his/her data collected and used to predict and prevent a possible nosocomial infection.

## 7. Conclusion

This article is the result of work on a user-centered approach for designing DSS based on KDD. For the last several decades, companies have stored a significant amount of information electronically. Company's information systems are designed to keep track of events reliably and with integrity. They automate more business processes, particularly to decision support. To accomplish this automation, the KDD process is used as decisional tool that makes it possible to explore databases to discover previously unknown knowledge that is potentially useful for the decision-making. Since KDD-based DSS are highly interactive, designers of such systems must rely on elements from two separate fields, Software Engineering (SE) and Human-Computer Interaction (HCI).

After studying the models from both the SE and HCI fields, we proposed a user-centered approach, called UP/U, based on the Unified Process from the field of the SE and the U model from the field of the HCI. Using these two complementary methods allowed us to take advantage of the positive elements in both fields to develop Decision Support Systems (DSS). To validate our approach, we designed and constructed a KDD-based DSS for a hospital Intensive Care Unit (ICU). This system aims to facilitate the fight against nosocomial infections contracted in such units. Throughout the entire process (needs assessment, task definition, identification of typical user characteristics, modelling and prototyping, evaluation and validation), we were in constant contact with the physicians, the system's end users. The final DSS will have four modules: the first two modules have already been developed using the UP/U approach, the third is still under development [4] and the fourth one remains at the project stage. An ongoing study relating to the module "data mining", using dynamic Bayesian networks is underway [55].

In the short term, we plan to continue developing the two remaining modules (evaluation and knowledge management). In the long term, we expect to validate our UP/U approach on other types of KDD-based DSS, using other data mining techniques, such as association rules and Bayesian networks. The information produced by these techniques has a very complex form, especially for a user who is not a specialist in the field of the artificial intelligence. We hope to be able to facilitate the user's comprehension of this complex information. We also hope to propose a specific evaluation methodology for DM-based DSS, taking as our starting point the evaluation criteria, methods and techniques used in the fields of HCI and visual data mining [35].

## 8. Acknowledgments

The first, second and fourth authors would like to acknowledge the financial support provided by grants from the General Department of Scientific Research and Technological Renovation (DGRST), Tunisia, under the ARUB program 01/UR/11/02. Thanks also to Dr. H. Kallel and all the ICU staff of Habib Bourguiba Teaching Hospital for their interest to the project and all the time spent to help us design, use and evaluate our system. The third author would like to acknowledge the Nord-Pas de Calais region, the French national government and the FEDER program for their financial support of the TAC MIAOU and EUCUE projects, in which were studied models and methods that proved to be very useful within the framework of this study.

## 9. Bibliography

- [1] M. Abed, Contribution à la modélisation de la tâche par outils de spécification exploitant les mouvements oculaires: application à la conception et à l'évaluation des interfaces homme-machine, PhD Thesis, University of Valenciennes, France, 1990.
- [2] M. Abed, J.M. Bernard, J.C. Angué, Task analysis and modelization by using SADT and Petri Networks, in: Proceedings Tenth European Annual Conference on Human Decision Making and Manual Control, Liege, Belgium, 1991.
- [3] M. Abed, J.C. Angué, A new method for conception, realisation and evaluation of man-machine, in: Proceedings of the IEEE International Conference on SMC, 1994.
- [4] R. Bahloul, M. Ben Ayed, A. M. Alimi, Vers une méthode d'évaluation de Système Interactif d'Aide à la Décision basé sur le processus d'ECD, in: Atelier : Évaluation des méthodes d'Extraction de Connaissances dans les Données, 10ème Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances, EGC 2010, Hammamet, Tunisia, 26-29 janvier 2010.
- [5] S. Balbo, Evaluation ergonomique des interfaces utilisateur: un pas vers l'automatisation, PhD Thesis, University of Grenoble, France, 1994.
- [6] M. Beynon, S. Rasmequan, S. Russ, A new paradigm for computer-based decision support, *Decision Support Systems* 33 (2002), 127-142.
- [7] B. Boehm, A Spiral Model of Software Development and Enhancement, *Computer* 21, (May 1988) 61-72.
- [8] S.E. Brossette, A.P. Sprague, J.M. Hardin K.B. Waites W.T. Jones, S.A. Moser, Association Rules and Data Mining in Hospital Infection Control and Public Health Surveillance, *Journal of the American Medical Informatics Association*, 5 (4) (1998) 373-381.
- [9] S.E. Brossette, A.P. Sprague, W.T. Jones, S.A. Moser, A data mining system for infection control surveillance, *Methods of Information in Medicine* 39(4-5) (2000) 303-310.
- [10] G. Calvary, Proactivité et réactivité: de l'assignation à la complémentarité en conception et évaluation d'interfaces Homme-Machine, PhD Thesis, Grenoble I, 1998.
- [11] F.K.Y. Chan, J.Y.L. Thong, Acceptance of agile methodologies: A critical review and conceptual framework, *Decision Support Systems* 46 (2009) 803-814.
- [12] S. Chaudhuri, U. Dayal, An overview of data warehousing and OLAP technology, *ACM SIGMOD Record*, 261(997) 65-74.
- [13] U. Fayyad, S.G. Djorgovski, N.Weir, *Advances in Knowledge Discovery and Data Mining* AAAI/MIT Press, 1996.
- [14] G. Fischer, *Human-Computer Interaction Software: Lessons Learned, Challenges Ahead*, *IEEE Software* 6 (1) (1989) 44-52.
- [15] 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.
- [16] J.S. Garner, W.R. Jarvis, T.G. Emori, T.C. Hogan, J.M. Hugues, CDC definitions for nosocomial infections, *Am. J. of Inf. Contr.*, 16 (3) (1988) 128-140.
- [17] D. Hand, H. Mannila, P. Smyth, *Principles of Data Mining* MIT Press, Cambridge, 2001.
- [18] L. Hergafi, Présentation et validation d'un nouveau système pour la surveillance de l'infection acquise en réanimation, PhD Thesis, Faculty of Medicine of Sfax, Tunisia, 2006.
- [19] D. Hix, H.R. Hartson, *Developing user interface: ensuring usability through product & process*, Wiley, New York, 1993.
- [20] ISO 13407, *Human-centred design processes for interactive systems*, The International Organization For Standardization, 1999.
- [21] I. Jacobson, G. Booch, J. Rumbaugh, *The Unified Software Development Process* Addison Wesley Longman, 1999.
- [22] H. Kallel, M. Bouaziz, H. Ksibi, H. Chelly, C. Ben Hmida, A. Chaari, N. Rekik, M. Bouaziz, Prevalence of hospital-acquired infection in a Tunisian Hospital, *Journal of Hospital Infection* 59 (2005) 343-347.
- [23] B. Kent, *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 2000.

- [24] M. Klein, L.B. Methlie, *Expert Systems: A Decision Support Approach with applications in management and finance*, Addison-Wesley, 1990.
- [25] C. Kolski, *Interfaces Homme-Machine, application aux systèmes industriels complexes*, Hermes, Paris, 1997.
- [26] C. Kolski, A call for answers around the proposition of an HCI-enriched model, *ACM SIGSOFT Software Engineering Notes* 3 (1998) 93-96.
- [27] C. Kolski, H. Ezzedine, M. Abed, Développement du logiciel: des cycles classiques aux cycles enrichis sous l'angle des IHM, in: *Analyse et conception de l'IHM, Interaction Homme-Machine pour les SI*, Hermes, Paris, 2001, pp. 145-174.
- [28] M. Lajnef, M. Ben Ayed, C. Kolski, Convergence possible des processus du data mining et de conception-évaluation d'IHM: adaptation du modèle en U, in: *Proceedings of IHM 2005, International Conference Proceedings Series*, ACM Press, Toulouse, pp. 243-246.
- [29] C. Larman, *Applying UML and Patterns, An Introduction to Object-Oriented Analysis and Design and the Unified Process*, third ed., Prentice Hall, 2004.
- [30] C. Larman, *Agile and Iterative Development: a Manager's Guide*, Addison-Wesley, Pearson Education, 2007.
- [31] R. Lefébure, G. Venturini, *Data Mining: Gestion de la relation client, Personnalisation des sites Web*, Eyrolles, Paris, 2001.
- [32] F. Lemieux, M.C. Desmarais, RUP et conception centrée sur l'utilisateur: une étude de cas, in: *Proceedings ERGO-IA'2006 Conference*, 11-13 octobre 2006, Biarritz, pp. 237-244.
- [33] S. Lepreux, M. Abed, C. Kolski, A human-centred methodology applied to decision support system design and evaluation in a railway network context, *Cognition Technology and Work* 5 (2003) 248-271.
- [34] S. Lepreux, *Approche de Développement centré décideur et à l'aide de patrons de Systèmes Interactifs d'Aide à la Décision*, PhD Thesis, University of Valenciennes, France, 2005.
- [35] H. Ltifi, M. Ben Ayed, S. Lepreux, A. M. Alimi, Survey of Information Visualization Techniques for Exploitation in KDD, in: *7th ACS/IEEE International Conference on Computer Systems and Applications, AICCSA-2009*, May 10-13, Rabat, Morocco, pp. 218-225.
- [36] H. Ltifi, M. Ben Ayed, C. Kolski, A. M. Alimi, HCI-enriched approach for DSS development: the UP/U approach, in: *14th IEEE Symposium on Computers and Communications, ISCC 2009*, July 5-8, Sousse, Tunisia, pp. 895-900.
- [37] J. McDermid, K. Ripkin, *Life cycle support in the ADA environment*, Cambridge University Press, Cambridge, 1984.
- [38] P. Millot, S. Debernard, Men machines cooperative organizations: Methodological and Practical Attempts in air traffic control, in: *Proceedings IEEE SMC*, Le Touquet, France, October 1993.
- [39] H. Mintzberg, D. Raisinghani, A. Theoret, The structure of "unstructured" decision processes, *Administrative Science Quarterly*, 1976.
- [40] B.A. Myers, User Interface Tools: Introduction and Survey, *IEEE Software*, 6 (1989) 15-23.
- [41] D. Nakache, *Data warehouse et data mining*, CNAM, Lille, accessible at: <http://home.nordnet.fr/~dnakache/valeurc>, 1998.
- [42] J. Nielsen, *Usability Engineering*, Academic Press, Boston, 1993.
- [43] D.J. Power, *Decision support systems: concepts and resources for managers*, Westport, Quorum Books, 2002.
- [44] S. Rathnam, M.V. Mannino, Tools for building the human-computer interface of a decision support system, *Decision Support Systems* 13 (1995) 35-59.
- [45] C. Riesbeck, R. Shank, *Inside Case-Based Reasoning*, Lawrence Erlbaum, 1989.
- [46] J.M. Robert, Que faut-il savoir sur l'utilisateur pour réaliser des interfaces de qualité?, in: G. Boy (Ed.), *Ingénierie cognitive, IHM et cognition*, Hermes, Paris, 2003, pp. 249-283.
- [47] M. Rosson, J.M. Carroll, Scenario-based design, in: J.A. Jacko, A. Sears (eds.), *The human-computer interaction handbook*, Lawrence Erlbaum Associates, London, 2003, pp. 1032-1050.
- [48] W. Royce, *Managing the development of large software systems: Concepts and techniques*, WESCON, technical papers, 1970.

- [49] J.R. Ruault, UML and interactive systems, another step forward, in: C. Kolski, J. Vanderdonckt (eds.), *Computer-aided Design of User Interface III*, Kluwer Academic Publishers, Dordrecht, 2002, pp. 243-256.
- [50] J. Rumbaugh, I. Jacobson, G. Booch, *The Unified Modeling Language Reference Manual*, Addison-Wesley Professional, 1999.
- [51] D.L. Scapin, J.M.C. Bastien, Ergonomic criteria for evaluating the ergonomic quality of interactive systems, *Behaviour and Information Technology* 16 (1997) 220-231.
- [52] D. L. Scapin, C. Pierret-Golbreich, Towards a method for task description, in: *Proceedings of Conference on Working With Display Units*, L. Berlinguet, D. Berthelette (eds.), Amsterdam, The Netherlands: Elsevier, 1990, pp. 371-380.
- [53] H.A. Simon, *The new science of management decision*, Prentice Hall, Harper and Row, 1960.
- [54] C. Suetens, A. Savey, J. Labeeuw, I. Morales I, Le projet HELICS-ICU: vers une surveillance européenne des infections nosocomiales dans les unités de soins intensifs, *Euro Surveill.*, 7(9) (2002) 127-128.
- [55] G. Trabelsi, M. Ben Ayed, M.A. Alimi, Système d'extraction des connaissances à partir des données temporelles basé sur les Réseaux Bayésiens Dynamiques, in: *10ème Conférence Internationale Francophone sur l'Extraction et la Gestion des Connaissances, EGC 2010*, Hammamet, Tunisia, 26-29 janvier 2010, *Revue des Nouvelles Technologies de l'Information RNTI-E-19*, 2010, pp 241-246.
- [56] M.C. Tremblay, R. Fuller, D. Berndt, J. Studnicki, Doing more with more information: Changing healthcare planning with OLAP tools, *Decision Support Systems* 43 (2007) 1305–1320.
- [57] E. Turban, *Decision Support and Expert Systems*, Macmillan, New York, 1993.
- [58] E. Turban, J.E. Aronson, T. Liang, R. Sharda, *Decision Support and Business Intelligence Systems*, Pearson Education, 2008.
- [59] G.C. Van der Veer, M.D.C. Puerta Melguizo, Mental models, in: J.A. Jacko, A. Sears (eds.), *The human-computer interaction handbook*, Lawrence Erlbaum Associates, London, 2003, pp. 52-80.
- [60] J.R. Wilson, E.N. Corlett (eds.), *Evaluation of human works: a practical ergonomics methodology*, second edition, Taylor & Francis, 1996.