A Knowledge-Based Software Life-Cycle Framework for the Incorporation of Multicriteria Analysis in Intelligent User Interfaces

Katerina Kabassi and Maria Virvou, *Member*, *IEEE*

Abstract—Decision-making theories aiming at solving decision problems that involve multiple criteria have often been incorporated in knowledge-based systems for the improvement of these systems' reasoning process. However, multicriteria analysis has not been used adequately in intelligent user interfaces, even though user-computer interaction is, by nature, multicriteria-based. The actual process of incorporating multicriteria analysis into an intelligent user interface is neither clearly defined nor adequately described in the literature. It involves many experimental studies throughout the software life-cycle. Moreover, each multicriteria decision-making theory requires different kinds of experiments for the criteria to be determined and then for the proper respective weight of each criterion to be specified. In our research, we address the complex issue of developing intelligent user interfaces that are based on multicriteria decision-making theories. In particular, we present and discuss a software life-cycle framework that is appropriate for the development of such user interfaces. The life-cycle framework is called MBIUI. Given the fact, that very little has been reported in the literature about the required experimental studies, their participants and the appropriate life-cycle phase during which the experimental studies should take place, MBIUI provides useful insight for future developments of intelligent user interfaces that incorporate multicriteria theories. One significant advantage of MBIUI is that it provides a unifying life-cycle framework that may be used for the application of many different multicriteria decision-making theories. In the paper, we discuss the incorporation features of four distinct multicriteria theories: TOPSIS, SAW, MAUT, and DEA. Furthermore, we give detailed specifications of the experiments that should take place and reveal their similarities and differences with respect to the theories.

Index Terms—Decision support, knowledge engineering methodologies, software engineering process, user interfaces.

1 Introduction

The use of computers has rendered the performance of many tasks more efficient than it used to be. However, it has also introduced new kinds of problems which are mainly due to the complexity of user interfaces. Such problems are addressed by Intelligent User Interfaces (IUIs) that aim at improving human computer interaction.

Common approaches proposed in the literature for incorporating intelligence in user interfaces include probabilistic reasoning through Bayesian Networks, machine-learning algorithms, such as neural networks and Case-Based Reasoning. All of these techniques try to model the user's reasoning process and have proven to be rather effective. However, a user interface that provides intelligent advice should also be able to reproduce human advisors' reasoning. For this purpose, decision-making theories seem very promising. Indeed, decision-making theories have been used for selecting the best information source when a user submits a query [1], modeling user preferences in recommender systems [2], selecting the best route in mobile guides [3] or individualizing e-commerce Web pages [4], [5].

The decision-making theories that seem to be more appropriate for computer problems are the multicriteria ones. This is due to the fact that computer problems usually involve several objectives and criteria. Decision-making theories provide precise mathematical methods for combining criteria in order to make decisions. However, they do not define the criteria. Therefore, the first step for applying any decision-making theory is to conduct an empirical study for selecting the criteria that are usually taken into account by a human decision maker. Furthermore, many decision-making theories require other additional empirical studies for applying the mathematical model proposed. For example, additional empirical studies may be needed for the estimation of the weights of the criteria, etc. Thus, it is evident that the adaptation and application of decision-making theories into intelligent software requires the conduction of empirical studies at the early stages of the development process. Consequently, the design of these experiments affects the efficiency of the resulting IUIs. Indeed, if the experiments are not carefully designed and implemented, then there is a possibility that useful pieces of knowledge are missed out and the application of the decision-making theory fails in the end.

Despite the importance of the development process of IUIs that are based on multicriteria theories, very little information is reported in the relevant literature about it. As a matter of fact, this is a problem that concerns the IUI

- *K. Kabassi is with the Technological Educational Institute of the Ionian Islands, 2 Kalvou Sq., 29100 Zakynthos and the University of Piraeus, 80 Karaoli & Dimitriou St., 18534 Piraeus, Greece. E-mail: kkabassi@unipi.gr.*
- *M. Virvou is with the University of Piraeus, 80 Karaoli & Dimitriou St., 18534 Piraeus, Greece. E-mail: mvirvou@unipi.gr.*

Manuscript received 1 Oct. 2005; revised 24 Mar. 2005; accepted 31 Mar. 2006; published online 19 July 2006.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-0452-1005.

literature in general. Delisle and Moulin [6], after an exhaustive review of the relevant literature, have come to the conclusion that there is a shortage of guidelines available for the development of IUI applications. In the specific case of IUIs that involve multicriteria theories, there are many development steps that are required for their effective application. These steps are neither trivial nor adequately described in the relevant literature. This is probably one important reason why decision-making theories have not been widely used in IUIs, despite the fact that their application seems very promising. Thus, it is quite important to define and present a life-cycle framework for the incorporation of a multicriteria theory in an IUI. Given the fact that there are many multicriteria theories, it is also important to highlight the similarities and differences of these theories in terms of the process of their possible incorporation into IUIs. Therefore, the main focus of the present paper is on how a decision-making theory can be incorporated in an IUI.

In view of the above, our research has focused on creating a generic software life-cycle framework of how a decision-making theory can be applied effectively in an intelligent user interface. The resulting framework is called MBIUI (Multicriteria-Based Intelligent User Interface) life-cycle framework and involves the description of a software life-cycle that gives detailed information and guidelines about the experiments that need to be conducted, the design of the software, the selection of the right decision-making theory, and the evaluation of the IUI that incorporates a decision-making theory.

As an example of use of the MBIUI framework, we have developed MBIFM, which is an IUI that bases its reasoning on the decision-making theory called TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) [7]. However, in this paper, in the context of the MBIUI framework, we discuss issues about the possible application of other multicriteria theories such as SAW (Simple Additive Weighting) [7], [8], MAUT (Multi-Attribute Utility Theory) [9], and DEA (Data Envelopment Analysis) [10].

The main body of the paper is organized as follows: Section 2 describes related work in software life-cycle processes and IUIs. Moreover, we give a very brief description of the four decision-making theories that are discussed in the paper. Section 3 presents an overview of MBIUI framework. MBIUI is further analyzed in the subsequent sections, where we give examples of its use for the development of MBIFM using TOPSIS.

2 Related Work

2.1 Software Life-Cycle Processes

As systems become more complex, their development and maintenance is becoming a major challenge [11]. This is particularly the case for software that incorporates intelligence. Indeed, intelligent systems are quite complex and they have to be developed based on software engineering approaches that are quite generic and do not specialize on the particular difficulties of the intelligent approach that is to be used.

This problem has given rise to research that is oriented toward bridging the gaps between software engineering approaches and the development of special purpose intelligent systems. For example, Del Socorro Bernardos [12] proposes a general framework that helps develop a natural language generation (NLG) project from the conception of the need to the retirement of the product. This framework is based on the IEEE standards 1074-1997 [13]. ADELFE [14] is another methodology that is specifically devised for software engineering of adaptive multiagent systems. ADELFE is based on the Rational Unified Process (RUP) [15] and its objective is not to add another methodology but to work on some aspects not already considered by existing methodologies, such as complex environment, dynamic software adaptation.

Similarly, to the research projects mentioned above, our research presents a knowledge-based software engineering framework for a special category of intelligent systems. In our case, this category concerns IUIs that incorporate multicriteria theories. The resulting framework of our research is called MBIUI. Like ADELFE, MBIUI is based on the RUP.

RUP is an object-oriented process that advocates multiple iterations of the software development process. It divides the development cycle in four consecutive phases: the inception, the elaboration, the construction, and the transition phase. Each phase is divided into four procedural steps, namely, requirements capture, analysis and design, implementation, and testing. The phases are sequential in time but the procedural steps are not.

RUP is clearly documented and easily used due to its clarity as has been pointed out in [11], where RUP has been compared and contrasted to other software engineering processes, such as Catalysis [16] and OPEN [17]. Additionally, to its clarity, RUP is an object-oriented process; thus, it is appropriate for the development of graphical user interfaces such as the one described in our research. Moreover, one important advantage of RUP is the highly iterative nature of the development process. For the above reasons, RUP has been selected as the basis for the MBIUI life-cycle framework. However, RUP does not provide specific guidelines about what sort of experiments or prototypes are needed and when. Such information concerning IUIs that are based on multicriteria decision making is given by the MBIUI life-cycle framework, which is described in this paper.

2.2 Intelligent User Interfaces

Several researchers have dealt with the issue of designing and developing IUIs using theories from various research areas. In addition to other theories, decision-making theories have also been used as reasoning methods in IUIs. One common application area of decision-making theories in IUIs is e-commerce. The theory that has been used most extensively in this area is MAUT (e.g., [2], [4], [5], [18]). All of the above mentioned projects have focused on adapting a theory in order to improve the user interface by dynamically selecting the most appropriate e-commerce product to be recommended to a user. However, there is a shortage of reports on experimental studies that are needed throughout the life-cycle of such IUIs that base their reasoning on multicriteria theories.

If a user interface incorporates intelligence, the complexity of the development of the system increases dramatically. This is even more the case when multicriteria theories are meant to be incorporated into the reasoning of the system. Interesting research on this subject has been conducted by Bohnenberger et al. [3] who have presented the studies that are needed for developing a decision-theoretic location-aware shopping guide. However, these studies mainly focus on the evaluation and the possible improvements of the system that incorporates decision theoretic planning and not the whole life-cycle. Therefore, it is our goal to present the life-cycle of an IUI that incorporates a decision-making theory and address the problems that are related to such incorporation.

As a test-bed for the corpus of our research, we have used a file manipulation environment like Microsoft Windows Explorer. The aim of our research is to generate automatic assistance in cases where users have made mistakes with respect to their hypothesised intentions. Another help system, which is similar to the domain of our research but different in its basic rationale is Tip Wizard that has been introduced by Microsoft. Tip Wizard is quite well known to users of Windows who remember the animated agent of the paper-clip. However, this paper-clip was often criticized as annoying [19]. Tip Wizard's main objective is to recommend new commands to users. This is done based on alternative commands' equivalence to the less efficient command sequence that a user may be using in order to perform a task. In contrast, in our research, the objective of the intelligent help system is to intervene only when this is considered really necessary for helping the user achieve his/her goals without errors. In our approach, we do not consider giving comments to users on the actual way they choose to accomplish their goals as long as this way is not leading to a failure. In this way, we keep the interventions to a minimum to avoid annoyance of users. Therefore, in our approach, if the help system suspects that an action would not have the desired results for the user, it generates alternative actions that would achieve these hypothesized goals. Interventions are made only in cases when the multicriteria decision-making theory has ranked very highly the alternative to be proposed to the user. Furthermore, another advantage of our approach, in contrast to the one adopted by Microsoft, is that our system takes into account information about the user's goals, usual errors, and misconceptions and, therefore, makes interaction adaptive to each individual user. An example of the IUI's operation is presented below.

A user in his attempt to organize his file store moves the contents of two folders into a third one, named "Programs." Then, the user accidentally selects the folder "Programs" and issues the command delete. However, this action seems contradictory to the user's goals as the particular folder has just acquired new contents that may be useful for the particular user. Therefore, the system generates alternative actions and applies a decision-making model in order to select the one that seems more likely to have been intended by the user. As a result, the system proposes to the user to delete another folder that has a similar name: "Program" and is empty. The folders "Programs" and "Program" have similar names and are neighboring in the graphical representation of the file store, thus the user may have mistakenly selected "Programs" instead of "Program." In the reasoning of the system, a factor that was also taken into consideration for the selection of the command to be proposed to the user was the fact that from the history of the user's actions the user had been characterized as prone to accidental slips. The actual selection of the "best" alternative command to be proposed to the user is based on a multicriteria theory.

Our research described in the present paper is based on previous research of ours on IUIs [20], [21], [22]. In particular, in [21], we argued that experimental studies were needed for the life-cycle of an IUI irrespective of the incorporation of a multicriteria theory. In that paper, we described the experimental studies that were needed for the development of an IUI prototype system, which was called IFM. IFM's reasoning was primarily based on a cognitive theory and did not incorporate any decision-making theory at all [20]. As a major improvement of our research, the cognitive theory was combined with the multicriteria decision-making theory, which is called SAW. The combination of SAW with the cognitive theory is described in [22].

However, the incorporation of a decision-making theory into IFM revealed a greater role that decision-making theories can play into the reasoning of IUIs in terms of the way their specifications are formed, their functionality, and maintenance. At the same time, important questions were raised: 1) What sort of experiments are needed for the efficient incorporation of a multicriteria theory into an IUI? 2) How should these experiments be set up? 3) How do these experiments differ depending on the particular multicriteria theory that is used? These questions are very crucial for the incorporation of a multicriteria theory into an IUI. This is due to the fact that this incorporation is a process that involves many stages and experimental studies that are dedicated to the purpose of incorporating a multicriteria theory. For example, the application of a decision-making theory in an IUI requires conducting experiments so that decision-making information may be acquired from the human experts. Therefore, a designer has to address issues like setting up the experiments. Such issues, once determined, can be reused in other similar systems.

The above questions motivated extensive further research that led to a brand new version of the IUI that is now called MBIFM (Multicriteria-Based Intelligent File Manipulator) and is based extensively on the decision-making theory which is called TOPSIS [7]. The multicriteria theory has been used for the whole reasoning of the system and, thus, the cognitive theory is not used any more. The life-cycle framework that was devised for the development of MBIFM is called MBIUI and is described in the present paper. Similarly to [21], MBIUI is based on an adaptation of RUP, which is an object-oriented software life-cycle model. However, unlike [21], MBIUI is dedicated to a software life-cycle model for the development of IUIs that incorporate a decision-making theory and the main emphasis of the research described in the present paper has been put on the adaptation of decision-making theories in IUIs.

2.3 Decision-Making Theories

According to Triantaphyllou and Mann [23], there are three steps in utilizing a decision-making technique that involves numerical analysis of alternatives: 1) determining the relevant attributes and alternatives, 2) attaching numerical

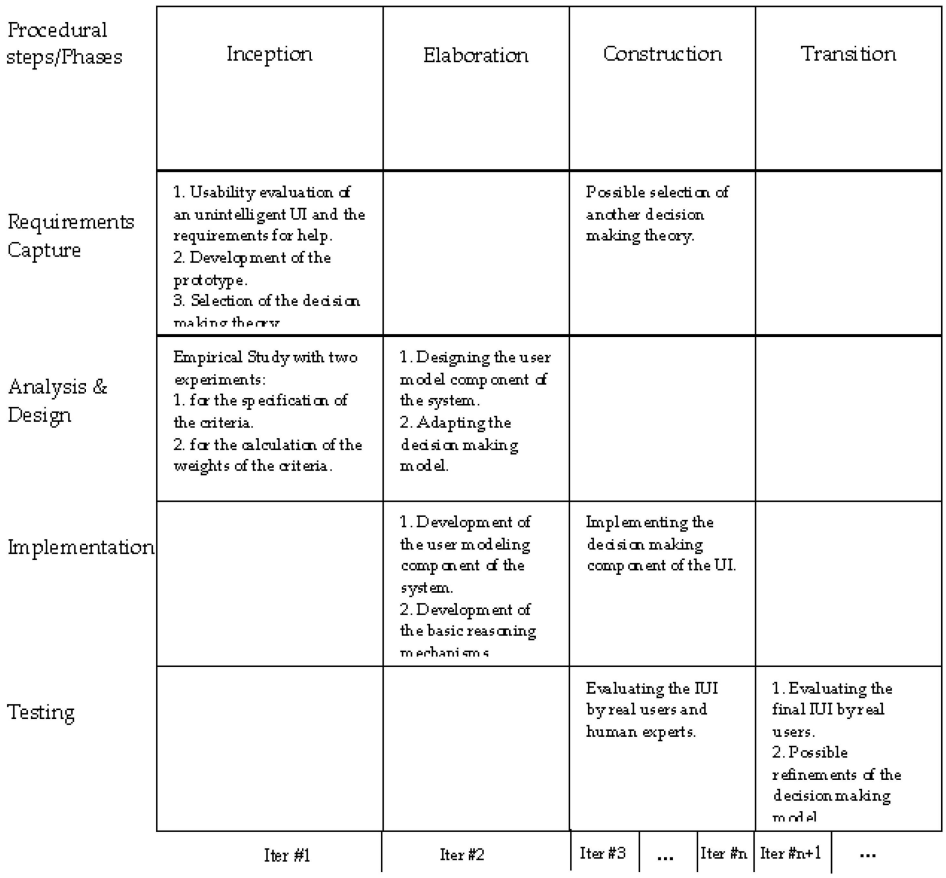

Fig. 1. The RUP as adapted for the MBIUI life-cycle framework.

measures to the relative importance of the attributes and to the impacts of the alternatives on these attributes, and 3) processing the numerical values to determine a ranking of each alternative.

The determination of the relevant attributes and their relative importance is made at the early stages of the software life-cycle and is performed by the developer or is based on an empirical study which may involve experts in the domain. However, decision-making techniques mainly focus on Step 3. There are many decision-making techniques and they have similarities and dissimilarities. In the following paragraphs, we present briefly and discuss issues about four decision-making techniques, namely, SAW, MAUT, DEA, and TOPSIS.

In typical decision-making methods such as the SAW [7], [8], the alternative actions are ranked by the values of a multiattribute function that is calculated for each alternative as a linear combination of the values of the n attributes. The multiattribute function used in SAW is also used in MAUT. However, the main difference of the particular theory with SAW is that the two theories use different experiments for the calculation of the weights of the criteria.

SAW and MAUT presuppose that the weights of the criteria are calculated in the early phases of the theory's application and do not change over time. However, the weights of the criteria are calculated dynamically in other decision-making theories such as the Data Envelopment Analysis (DEA) [10]. DEA is a nonparametric linear programming approach to evaluate the relative efficiency of decision making units (DMUs) that use multiple inputs to produce multiple outputs. Unlike SAW, MAUT, and DEA, TOPSIS calculates the relative Euclidean distance of the alternative from a fictitious ideal alternative. The alternative closest to that ideal alternative and furthest from the negative-ideal alternative is chosen best. More specifically, the steps that are needed in order to implement the technique are:

1. Scale the values of the n attributes to make them comparable.
2. Calculate weighted ratings.
3. Determine positive-ideal and negative-ideal solutions.
4. Calculate the separation measure from the positive-ideal and negative-ideal alternative.
5. Calculate similarity indexes.

3 MBIUI Life-Cycle Framework

MBIUI life-cycle framework is based on RUP. As already mentioned, RUP gives a framework of a software life-cycle that is based on iterations. However, RUP does neither specify what sort of requirements analysis has to take place nor what kind of prototype has to be produced during each phase or procedural step. Such specifications are provided by our MBIUI framework concerning IUIs that are based on multicriteria theories.

The MBIUI framework is illustrated in Fig. 1. In this figure, we have maintained the phases and procedural steps of RUP. Based on this, we have specified what kind of prototype has to be constructed in each iteration and what kind of experiment has to be conducted. Therefore, Fig. 1 represents our solution to this problem.

According to MBIUI framework, during the inception phase, the requirements capture is conducted. During requirements capture, a nonintelligent version of the user interface has to be evaluated. The usability evaluation of nonintelligent version of the user interface has to be conducted in order to identify the usability problems of such user interfaces. The usability problems of nonintelligent user interfaces that can be identified can serve as a basis for the requirements specification of the intelligent version. Furthermore, during the requirements capture, a prototype of the IUI should be developed. This includes the specification of the way that the generation of intelligent advice to be proposed to the user takes place in the system. However, the first prototype cannot include a user model or the adaptation of a multicriteria decision-making theory, which requires further experiments. At this point, the multicriteria decision-making theory that seems most promising for the particular application has to be selected. This decision may be revised in the procedural step of requirements capture in the phase of construction.

According to MBIUI, in the inception phase, during analysis, two different experiments are conducted in order to select the criteria that are used in the reasoning process of the human advisors as well as their weights of importance. The experiments should be carefully designed since the kind of participants as well as the methods selected could eventually affect the whole design of the IUI.

In MBIUI life-cycle framework, the two experiments involve human experts in the domain being reviewed. These experts should comment on the protocols collected during the requirements capture of the inception phase. In particular, the human experts should define the criteria that they would use if they had to give advice to the users of the protocols. The criteria that are proposed by the majority of the human experts are selected. When the final set of criteria is formed, another experiment is conducted in which the human experts that participated in the first experiment are asked about the weight of importance of each criterion in their reasoning process. The setting of the second experiment depends on which decision-making theory has been selected.

The information collected during the two experiments of the empirical study is further used during the design phase of the system, where the decision-making theory that has been selected is applied to the user interface. More specifically, in the elaboration phase, during design, the user modeling component of the system is designed and the decision-making model is adapted for the purposes of the particular domain. Kass and Finin [24] define the user model as the knowledge source of a system that contains hypotheses concerning the user that may be important in terms of the interactive behaviour of the system.

In the elaboration phase, during implementation, the user modeling component of the system as well as the basic decision-making mechanisms are developed. As a result, a new version of the IUI is developed which incorporates fully the multicriteria decision-making theory.

In the construction phase, during testing, the IUI that incorporates the multicriteria decision-making theory is evaluated. The evaluation of IUIs is very important for their accuracy, efficiency, and usefulness. Indeed, as McTear [25] points out, the relationship between theory and practice is particularly important in Intelligent Interface Technology as the ultimate proof of concept that the interface actually works and that it is acceptable to users. Similarly, Chin [26] points out that empirical evaluations are needed to determine which users are helped or hindered by user-adapted interaction in user modeling systems. He adds that the key to good empirical evaluation is the proper design and execution of the experiments. However, he notes that empirical evaluations are not so common in the user modeling literature.

In MBIUI, evaluation is considered important for two reasons: 1) the effectiveness of the particular decision-making theory that has been used has to be evaluated and 2) the effectiveness of the IUI in general has to be evaluated. In case the version of the IUI that incorporates a particular decision-making theory does not render satisfactory evaluation results with respect to real users and human experts, then the designers have to return to requirements capture, select an alternative decision-making model, and a new iteration of the life cycle takes place. In the transition phase, during testing, the decision-making model that has been finally selected is evaluated and possible refinements of that model may take place, if this is considered necessary.

4 Requirements Capture

In the inception phase, during the procedural step of requirements capture a usability evaluation of a nonintelligent version of the user interface should be conducted so that the usability problems that the IUI has to address may be revealed.

In the case of MBIFM, a usability evaluation of a standard file manipulation program (Windows 98/NT Explorer) was conducted. This evaluation aimed at identifying usability problems of standard file manipulation programs so that these problems were addressed in the design of MBIFM. For this reason, we conducted an experiment, which involved both users and human advisors. One of the main aims of the empirical study was to categorize as many users' plans as possible and to identify the most frequent errors that expert and novice users may make while interacting with a standard explorer. In this way, we could identify limitations of standard file manipulation programs and find the main requirements for help. Indeed, this experiment revealed that the users had difficulties in copying or moving objects, in realizing the structure of the file store, in achieving their deletion goals successfully, etc.

The requirements identified by the above mentioned process were used for designing and developing a primary executable release of MBIFM. MBIFM uses the multicriteria theory TOPSIS. The particular theory has been chosen to serve as a test bed for our research because during the

knowledge acquisition, it does not require the setting of questions that would be difficult for experts to answer accurately. Indeed, if the questions made to the human experts are very difficult for them to answer, then the knowledge extracted may not be correct. Furthermore, calculating the relative distance from a fictitious ideal alternative seems more promising than just taking a linear combination of the criteria.

5 Analysis

Decision-making theories provide precise mathematical methods for combining criteria in order to make decisions but they do not define the criteria. Therefore, in order to specify the criteria that human experts (decision makers) take into account while providing individualized help and advice, an empirical study is needed during the early stages of analysis. More specifically, in the inception phase, during analysis, two different experiments were conducted for the specification of the criteria as well as their weights of the importance. The first experiment for the specification of the criteria is the same for the application of all four multicriteria decision-making theories that are examined in this paper.

The second experiment may vary according to the theory that has been selected to be adapted to the IUI. For example, multicriteria decision-making theories, such as SAW and TOPSIS, require an experiment like the one that will be described as an example below. However, there are some other decision-making theories, such as MAUT that have a predefined way for the calculation of the weights. Such theories usually describe the nature of the experiment by giving information about the kind of questions that should be made to the human experts (decision makers), how their answers should be used for the calculation of the weights of the criteria, etc. (for more information about the application of MAUT in an IUI, the reader can refer to [27]). Finally, some theories, such as DEA [10], do not require any experiment for the calculation of weights because the weights are dynamically calculated during the decision-making process.

5.1 First Experiment: Specifying the Criteria

The specification of the criteria through the first experiment is required for the application of any kind of multicriteria decision-making theory. The first experiment should involve a satisfactory number of human experts, who represent the decision makers. Therefore, in the experiment conducted for the application of the multicriteria theory in the IUI of our domain, 16 human experts were selected in order to participate in the empirical study. All the human experts possessed a first and/or higher degree in computer science and had teaching experience related to the use of file manipulation programs. The empirical study consisted of two main phases. In the first phase, all human experts were asked about the criteria that they used to take into account when providing individualized advice. From the criteria that appeared in the first experiment, only those proposed by the majority of the human experts were selected.

The first experiment revealed that the criteria that human experts take into account when providing advice in computer applications of our domain are the following:

- Frequency of an error (f): The value of this criterion shows how often a user makes a particular error. Some users tend to entangle similar objects, other users entangle neighboring objects in the graphical representation of the file store, and others mix up commands. As the frequency of an error increases, the possibility that the user has repeated this kind of error increases, as well.
- Percentage of the wrong executions of a command in the number of total executions of the particular command (e): The higher the number of wrong executions of a command, the more likely for the user to have failed in the execution of the command once again.
- Degree of similarity of an alternative action with the actual action issued by the user (s): Similar commands or objects of the file store are likely to have been confused by the user. Therefore, the similarity of the object and the command selected with the object and the command proposed by the system is rather important in order to locate the user's real intention.
- Degree of difficulty of a command (d): It has been observed that some commands are not easily comprehensible by the user. Therefore, the higher the degree of difficulty of a command, the more likely for the user to have made a mistake in this command.
- Degree of relevance to the user's goals (g): An alternative action may be proposed to a user if it confirms the user's goals or if it does not influence them. The actions that complete or continue an already declared and pending goal have higher degree of relevance to the user's goals than other actions.

These criteria were proposed among other criteria which were less popular by the majority of human experts that participated in the experiment.

5.2 Second Experiment: Determining the Weights of Importance of the Criteria

The second experiment for determining the weights of importance of the criteria is required for the application of theories that have predefined weights of importance of the criteria. Such theories are SAW, MAUT, and TOPSIS. However, other theories, such as DEA, allow the dynamic calculation of weights of criteria while the system is running. The dynamic calculation of the weights in DEA is performed based on the available alternatives. The kind of experiment that is needed for the adaptation of SAW and TOPSIS is the same and is described below as an example. In contrast, the experiment needed for the application of MAUT is different and is defined by the theory itself.

For the application of either SAW or TOPSIS in our domain, the settings of the second experiment are the following: The 16 human experts that participated in the first experiment participated in the second experiment as

well. They were asked to rank the five criteria that resulted from the first experiment with respect to how important these criteria were in their reasoning process. However, in TOPSIS, similarly to SAW, decision makers should give a weight of importance to each criterion but the theory does not predefine the scale used for the weights of the criteria. Several researchers have used different scale rating. For example, Zhu and Buchmann [28] use a scale from 1 (least desirable) to 9 (most desirable) for six different criteria. Then, all the weights given to a particular criterion are summed up and divided by the sum of the weights of all criteria. In this way, the sum of the weights of the criteria will be 1.

In view of this empirical study, a scale from 1 to 5 is proposed for rating the criteria. More specifically, every one of the human experts was asked to assign one score of the set of scores (1, 2, 3, 4, and 5) to each one of the five criteria and not the same one to two different criteria. The sum of scores of the elements of the set of scores was 15 (1 + 2 + 3 + 4 + 5 = 15). For example, a human expert could assign the score 5 to the degree of relevance to the user's goals (criterion g), the score 4 to the frequency of an error (criterion f), the score 3 to the percentage of the wrong executions of a command in the number of total executions of the particular command (criterion e), the score 2 to criterion s (degree of similarity of an alternative action with the actual action issued by the user), and the score 1 to the criterion d (degree of difficulty of a command).

As soon as the scores of all the human experts were collected, they were used to calculate the weights of the criteria. The scores assigned to each criterion by each human expert were summed up and then divided by the sum of scores of all criteria (16×15 human experts = 240). In this way, the sum of all the weights would be equal to 1.

As a result, the weight for the degree of similarity (s) is $w_s = \frac{75}{240} = 0.31$, the weight for the frequency of an error (f) is $w_f = \frac{39}{240} = 0.16$, the weight for percentage of the wrong executions of a command in the number of total executions of the particular command (e) is $w_e = \frac{37}{240} = 0.15$, the weight for the degree of difficulty of a command (d) is $w_d = \frac{27}{240} = 0.11$, and the weight for the degree of relevance the user's goals (g) is $w_g = \frac{62}{240} = 0.26$.

This process revealed that the most important criterion that human experts had used when they evaluated candidate alternative actions to suggest to a user was the similarity of the alternative action generated by the system to the actual action of the user. The majority of them thought that similarity was important because users usually tend to tangle up actions or objects that are very similar and most users assigned to it the score 5. The weight of the particular criterion that represents its relative importance as this has been estimated by the empirical study is 0.31.

The second most important criterion was considered to be the degree of relevance to the user's goals. For example, an action that achieves or leads to a user's goal is considered to be more likely to have been intended by that user than another command that is irrelevant to the users' goals. The weight assigned to this criterion was 0.26, which was quite close to the most important criterion. The frequency of an error was also considered to be rather important. For example, some users are more prone to accidental slips while others make mistakes due to lack of knowledge. The weight assigned to this criterion was 0.16. A quite similar weight was also given to the criterion e, which corresponds to the percentage of the wrong executions of a command in the number of total executions of the particular command. Its weight was estimated to be 0.15. Finally, human experts thought that the degree of difficulty of a command was also to be considered. Therefore, the weight of the degree of difficulty of a command was estimated by the above mentioned procedure to be 0.11.

6 Design

In MBIUI, the design of the running application is mainly concerned with the design of the user model and is divided into two major parts with respect to the application of a multicriteria decision-making theory: 1) design decisions about how the values of the criteria are estimated based on the information of the user model and 2) the design of the embedment of the actual multicriteria theory that has been selected into the system. The first part is independent of the multicriteria theory that has been used and is the same for all of them.

MBIFM collects information about its users implicitly, based on their behavior while interacting with the system. Furthermore, the system maintains one user model for every user that interacts with the system. The user model consists of the system's hypotheses about the user's goals as well as information that concerns the user's level of knowledge of the domain, his/her common errors, the correct and wrong executions of a command, etc. This information is further used by the system in order to calculate the values of the criteria that are user dependent.

The value of the criterion f that refers to the frequency of an error is calculated by dividing the times a particular user has made an error by his/her total errors. The value of the criterion e that represents the percentage of the wrong executions of a command in the number of total executions of the particular command is calculated by dividing the times the user has made an error in the execution of a command by the total number of the command's execution. The degree of relevance to the user's goals is estimated by taking into account the information about the user's goals that is stored in the individual short-term user model. If the alternative action that is evaluated results in the achievement of a goal or a subgoal, then the value of the particular criterion is 1; otherwise, its value is 0.5.

The values of the other two criteria, s and d, are calculated by taking into account the information that is stored in the knowledge representation component of the system. For example, the degree of difficulty of each command is given a prefixed value that is maintained constant for all users. The degree of similarity, on the other hand, is dynamically calculated based on the user's file store. More specifically, the value of criterion s that represents the similarity of an alternative action to the actual action issued by the user depends on the similarity of the commands and the similarity of the objects that are involved in these two actions. The similarity between two commands of the hierarchy of user actions is precalculated.

The value is estimated by taking into account the result of the commands, their relative distance in the user actions hierarchy of actions that the user is allowed to perform by the GUI and, finally, the actions' relative geographical position in the graphical user interface. For example, two commands that have a very similar result, such as "cut" and "copy" commands, have a great degree of similarity. In the case of objects, the similarity is dynamically calculated. The value of the similarity of two objects is partially based on the resemblance of their names (for example, directories "Project" and "Projects"), but it is also based on the relative distance of objects in the graphical representation of the file store. For example, if two files are neighboring in the graphical representation of the file store, the user may have selected the one instead of the other.

In the inception phase, during design, as soon as decisions about the calculation of the values of the criteria have been made, then the decision-making theory may be adapted to the system. As already mentioned, the theory that has been selected to be adapted to MBIFM is TOPSIS. According to this theory, the first step involves scaling the values of the five attributes to make them comparable. The values are calculated in a way so that they would be in the interval [0, 1] and, therefore, are comparable.

Then, the system should calculate the weighted ratings of the criteria using the weights (w_s, w_g, w_f, w_e, and w_d) that have been estimated during the empirical study. The weighted values for the criteria are calculated by the formulas: $v_{sj} = w_s \cdot s_j$, $v_{gj} = w_g \cdot g_j$, $v_{fj} = w_f \cdot f_j$, $v_{ej} = w_e \cdot e_j$, and $v_{dj} = w_d \cdot d_j$.

TOPSIS is based on the concept that "the chosen alternative should have the shortest distance from a positive-ideal solution and the longest distance from a negative-ideal solution." Therefore, as a next step, the system should determine the Positive-Ideal and the Negative-Ideal alternative actions taking into account the weighted values of the criteria that were calculated above. The Positive-Ideal alternative action is the composite of all best criteria ratings attainable and is denoted: $A^* = \{v_s^*, v_g^*, v_f^*, v_e^*, v_d^*\}$, where $v_s^*, v_g^*, v_f^*, v_e^*, v_d^*$ are best weighted values of the criteria among all alternatives. The Negative-Ideal solution is the composite of all worst attribute ratings attainable: $A^- = \{v_s^-, v_g^-, v_f^-, v_e^-, v_d^-\}$, where $v_s^-, v_g^-, v_f^-, v_e^-, v_d^-$ are the worst weighted values for the criteria among all alternatives.

For every alternative action, MBIFM calculates the Euclidean distance from the Positive-Ideal and Negative-Ideal alternative. For the j alternative, the Euclidean distance from the Positive-Ideal and the Negative-Ideal alternatives are given by (1) and (2), respectively:

$$S_j^* = \sqrt{(v_{sj}-v_s^*)^2+(v_{gj}-v_g^*)^2+(v_{fj}-v_f^*)^2+(v_{ej}-v_e^*)^2+(v_{dj}-v_d^*)^2}, \quad (1)$$

$$S_j^- = \sqrt{(v_{sj}-v_s^-)^2+(v_{gj}-v_g^-)^2+(v_{fj}-v_f^-)^2+(v_{ej}-v_e^-)^2+(v_{dj}-v_d^-)^2}. \quad (2)$$

Finally, the similarity to the positive-ideal solution for the alternative action j, is given by the formula:

$$C_j^* = \frac{S_j^-}{S_j^* + S_j^-} \text{ with } 0 \le C_j^* \le 1. \quad (3)$$

Therefore, MBIFM selects the alternative action that has the highest value of C_j^*, which means that is closer to the positive ideal solution.

7 Implementation

During implementation, in the elaboration phase, the user modeling component of the system as well as the goal recognition mechanism of the system and the advice generation component are developed. The first component is responsible for the identification of the users' goals, whereas the second is used in order to generate alternative actions in case of an error. The selection of the best one is done by the decision-making component, which is developed in the construction phase, during the implementation of the system.

In this section, we present how TOPSIS is implemented in an IUI for the purposes of intelligent help. First, we present an overall description of the system's operation and then present a simple example of how TOPSIS is used for the selection of the best advice.

7.1 Overall Description of the System

MBIFM is a graphical user interface for file manipulation that provides intelligent help to its users. MBIFM monitors users' actions and reasons about them. In case it diagnoses a problematic situation, it provides spontaneous advice. When MBIFM generates advice, it actually generates alternative actions, other than the one issued, which was problematic. In this respect, MBIFM tries to find out what the error of the user has been and what his/her real intention was.

In order to make hypotheses about each user's possible intentions, the system uses a limited goal recognition mechanism [22]. Using this mechanism, MBIFM evaluates each user's action with respect to its relevance to the user's hypothesised goals. As a result of this evaluation, each action is categorized in one of four categories, namely, expected, neutral, suspect, and erroneous. Depending on the category, where it is categorized, the action is processed further by MBIFM or not.

In particular, if an action is compatible with the system's hypotheses about the user's intentions, it is categorized as expected. If it is neither expected nor contradictory to the user's hypothesized goals, it is categorized as neutral. In the cases of expected and neutral actions, the system executes the user's action normally without further notice. If an action contradicts the system's hypotheses about the user's intentions, it is categorized as suspect. Finally, if an action is wrong with respect to the user interface formalities, it is categorized as erroneous. In the cases of suspect and erroneous, the system tries to generate an action other than the one issued that would fit better in the context of the user's hypothesised intentions.

The same reasoning mechanism is also used to evaluate each alternative action that is generated by the system and only the actions that are compatible with the user's goals (expected or neutral) are selected. If the system does not manage to find even one alternative action that is compatible to the user's goals, the initial action of the user is executed normally, and the user never realizes that the

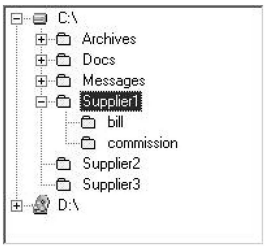

Fig. 2. The user's initial file store state.

system was alerted by his/her action. However, in most cases, this process results in the generation of many alternative actions. At this stage, an important role is revealed for a decision-making theory. Therefore, MBIFM uses TOPSIS in order to find the alternative action that seems more likely to have been intended by the user. For this purpose, MBIFM first calculates the values of the criteria for each alternative action for the particular user and then the weighted values of the criteria. The weighted values of the criteria are used for the determination of the positive-ideal and the negative-ideal alternative actions. Finally, the system chooses the alternative that has the shortest distance from a positive-ideal solution and the longest distance from a negative-ideal solution by calculating the similarity index. The alternative action with the highest similarity index is finally selected and proposed to the user. However, the user is not obligated to follow MBIFM's advice. She/he can execute her/his initial action or generate a new one. The efficiency of this method could only be tested if the system is evaluated by real end users. The evaluation setting and results are described in more details in Section 8.

7.2 An Example of MBIFM's Operation

In this section, we give an example of a user's interaction with MBIFM and how TOPSIS is used to select the alternative action that is going to be presented to the user. The user of the example is a male novice user that uses a file manipulation program for a fifth time. The initial file store state of the user is presented in Fig. 2.

The user first moves the contents of the folders "C:\Supplier2\" and "C:\Supplier3\" to the folder "C:\Supplier1\" and then he selects the folder "C:\Supplier1\" and issues the command delete. However, MBIFM finds the particular action as possibly not intended since the user in his previous actions had moved the contents of two other folders and, therefore, the folder possibly contains many files that may contain useful information. As a result, the system generates alternative actions to be suggested to the user.

The alternative actions that are generated by the system at first are the following:

AA1. Cut(C:\Supplier1\)
AA2. Copy(C:\Supplier1\)
AA3. Delete(C:\Messages)
AA4. Delete(C:\Supplier2\)
AA5. Delete(C:\Supplier3\)

TABLE 1
The Values of the Criteria for Every Alternative Action

	s	g	f	e	d
AA1	0.2	0.5	0.6	0.9	0.7
AA2	0.2	0.5	0.6	0.1	0.4
AA3	0.7	0.5	0.2	0.4	0.5
AA4	0.9	1	0.4	0.4	0.5
AA5	0.6	1	0.2	0.4	0.5

The first two alternative actions, AA1 and AA2, result from the assumption that the user has tangled up the command she wanted to issue (e.g., in AA1, the user may have clicked and selected the command delete instead of the command cut because he is not aware of the usage of the commands). The other three alternative actions (AA3, AA4, and AA5) were generated after substituting the selected target (folder) with other possible targets of the user (e.g., the user may have made an accidental slip and selected the wrong folder). For every alternative action that has been generated, the system uses the information from the user model as well as the information of the knowledge representation component in order to calculate the values of each criterion. Table 1 presents the values of the different criteria for every one of the five alternative actions.

The user of the example is novice and very prone to errors that are related to the selection of the wrong command. Therefore, the highest value of the criterion f is taken for the alternative actions AA1 and AA2, which have been generated based on the assumption that the user has selected the wrong command. However, the user is also prone to tangling up objects that are both similarly named and neighboring in the graphical representation of the file store. Therefore, the value of the criterion f is also high for the alternative action AA4, which has been made taking into account the assumption that the user has repeated such an error again.

The value of the criterion e is being estimated based on the percentage of wrong executions of a particular command. The command that seems to be very difficult for the particular user is the command cut as the 90 percent of this command's executions were erroneous. However, this percentage is a bit fictitious, as the user executes the particular action very rarely in contrast to the command delete that has been issued by the particular user many times. This is probably why human experts gave a lower degree of weighting to that criterion.

However, the command cut is generally considered very difficult. This is also verified by the knowledge representation component, which assigns 0.7 to the value of the criterion d for the command cut. Meanwhile, other commands, such as copy or delete, have lower values for the particular criterion as they are considered to be more easily comprehensible by the user.

The value of the criterion s shows how similar an alternative action is to the one originally issued by the user. The most similar alternative to the one issued by the user is AA4. This is so because the folders "C:\Supplier1\" and

TABLE 2
The Weighted Ratings for Every Alternative Action

	v_s	v_g	v_f	v_e	v_d
AA1	0.062	0.130	0.096	0.135	0.077
AA2	0.062	0.130	0.096	0.015	0.044
AA3	0.217	0.130	0.032	0.060	0.055
AA4	0.279	0.260	0.064	0.060	0.055
AA5	0.186	0.260	0.032	0.060	0.055

"C:\Supplier2\," which are involved in the action issued and AA4, respectively, are both similarly named and neighboring in the graphical representation of the user's file store. Therefore, the degree of similarity for that alternative is 0.9. The degree of similarity for the fourth alternative action is lower since the folders "C:\Supplier1\" and "C:\Supplier3\" have very similar names, but are not neighboring in the graphical representation of the file store. Finally, the two first alternative actions have even lower degrees of similarity and this is due to the fact that their similarity to the user's initial action lies only on the similarity of the commands.

The degree of relevance to the user's goals is estimated by taking into account the information about the user's goals that is stored in the individual short-term user model. If the alternative action that is evaluated results in the achievement of a goal or a subgoal, then the value of the particular criterion is 1; otherwise, its value is 0.5. The user had previously deleted the contents of the folder C:\Supplier2\ and C:\Supplier3\. Therefore, the deletion of the particular folders results in the completion of the deletion goals of the user. So, the value of the particular criterion takes the value 1 for the alternative action AA4 and AA5 and the value 0.5 for all the other alternative actions.

Then, MBIFM calculates the weighted ratings of the above mentioned criteria. The weighted values for the criteria are presented in Table 2.

The best weighted ratings are used for the formulation of the Positive-Ideal alternative action:

$$A^* = \{0.279, 0.260, 0.096, 0.135, 0.077\}.$$

Similarly, the Negative-Ideal solution is the composite of all worst attribute ratings attainable:

$$A^- = \{0.062, 0.130, 0.032, 0.015, 0.044\}.$$

For every alternative action, MBIFM calculates the Euclidean distance from the Positive-Ideal alternative action using (1): $S_1^* = 0.253$, $S_2^* = 0.282$, $S_3^* = 0.176$, $S_4^* = 0.084$, and $S_5^* = 0.137$. Using (2), MBIFM calculates the Euclidean distance of all alternative actions from the Negative-Ideal alternative action: $S_1^- = 0.140$, $S_2^- = 0.064$, $S_3^- = 0.162$, $S_4^- = 0.259$, and $S_5^- = 0.186$.

Finally, MBIFM estimates the similarity to the positive-ideal solution for every alternative action using (3):

$$\gamma_1^* = \frac{S_1^-}{S_1^* + S_1^-} = \frac{0.140}{0.253 + 0.140} = 0.356,$$
$$\gamma_2^* = \frac{S_2^-}{S_2^* + S_2^-} = \frac{0.064}{0.282 + 0.064} = 0.185,$$
$$\gamma_3^* = \frac{S_3^-}{S_3^* + S_3^-} = \frac{0.162}{0.176 + 0.162} = 0.479,$$
$$\gamma_4^* = \frac{S_4^-}{S_4^* + S_4^-} = \frac{0.084}{0.259 + 0.084} = 0.754,$$

and $\gamma_5^* = \frac{S_5^-}{S_5^* + S_5^-} = \frac{0.186}{0.186 + 0.137} = 0.575$.

These degrees show how similar each alternative is to the ideal alternative action A^*. Therefore, MBIFM selects the fourth alternative action (AA4) because it has the largest similarity index and that means that it is more similar to the Positive-Ideal alternative action.

Indeed, the fourth alternative is very similar to the user's initial action (the value of the criterion s is assumed for the alternative 0.90 and is the highest value for all alternative actions) and completes a previous goal of the user. Furthermore, the user seems to be very prone to errors that are related to the selection of the wrong object (the value of the criterion f is assumed for the fourth alternative 0.40) and the 40 percent of the execution of the command delete were erroneous. In view of the above, the system finds the fourth alternative as more likely to have been intended by the user and makes the following suggestion: "Your action was considered suspect by the system. Are you sure that was your real intention? If not, is the following your real intention? Delete(C:\Supplier2\). The system finds this action compatible to your goals. Furthermore, the degree of similarity of this action to your initial action is 0.90 and the degree of difficulty of the command you tried to execute is 0.50. Finally, the values of the criteria relating to the frequency of your errors shows that you have frequently made similar errors." Indeed, the user examined the system's advice and took it.

8 TESTING

In the construction phase, during the procedural step of testing, the final version of the system is evaluated. It is only by evaluating a user interface that one can be sure that an IUI really works and addresses the needs of real users. When a user interface incorporates a decision-making theory, the evaluation phase plays an important role for showing whether the particular theory is effective or not.

In the MBIUI life-cycle framework, it is considered important to conduct the evaluation of a decision-making model by comparing the IUI's reasoning with that of the human advisors' reasoning. This is so because the human experts' reasoning constitutes the reasoning that the system tries to model in the first place. Therefore, in this experiment, it is important to evaluate how successful the application of the decision-making model is in selecting the alternative action that the human experts would propose in the case of a user's error. For this reason, it has to be checked whether the alternative actions that are proposed by the human experts are also highly ranked by the application of the decision-making model. In case this comparison reveals that the decision-making model is not

TABLE 3
Summative Results about the System's Reasoning and the Correctness of Advice

Actions	1096
Suspect	146
Erroneous	47
Wrong alert	48
Right alert	145
Wrong advice	8
Right advice according to majority (among the alternatives)	107
Right advice according to minority (among the alternatives)	30
Right first advice according to majority (goal recognition mechanism)	49
Right first advice according to majority (TOPSIS)	82

adequate, another iteration of the life-cycle has to take place and another decision model should be selected. This iteration continues until the evaluation phase gives satisfactory results.

In view of the above, MBIFM was evaluated in order to ensure the completeness of its design and the usefulness of its operation. As MBIFM's main aim is to render the interaction more human-like in terms of intelligent and plausible responses of the system to users' errors, its evaluation aimed at revealing how successful MBIFM was at making decisions about what the user's real intentions were in comparison to human experts' opinions who observed the user's interaction with the system and provided spontaneous advice.

For the above purposes, 25 users (14 males and 11 females, age range 18-45) of different levels of expertise in the use of file manipulation programs and computers, in general, were asked to interact with a standard file manipulation program that did not incorporate intelligence. Each one of the users worked separately from all the others and their actions were recorded. The protocols collected during this process were given to 10 human experts who were asked to comment on them. In particular, each human expert had to study carefully each one of the user actions separately from all the others and reasoned about every user action. In case they found that an action was erroneous or unintended, they provided one alternative action, which had been what they thought the user's real intention was.

The protocols collected were also given as input to the IUI and MBIFM reasoned about every user's action. In case MBIFM found an action as not intended, it generated alternative actions and used TOPSIS to select the one that would propose to the user. The alternative action proposed by MBIFM was compared to the one proposed by the human experts in order to check how successful the system was in comparison with a human expert who constantly observed the user and provided spontaneous advice.

The users' protocols that were examined in Table 3 consisted of 1,096 users' actions. From these actions, MBIFM categorized 193 actions as not really intended by the user and was alerted to generate alternative actions. More specifically, MBIFM was alerted in 146 actions because it categorized them as suspect and in 47 that were erroneous. Human experts agreed with MBIFM in 145 of the 193 actions (percentage of success in identifying an unintended action = 75.13 percent). This meant that in 24.87 percent of the actions where MBIFM was alerted, the human experts did not agree with the system. In each case of right alert (where MBIFM was in accordance with the majority of human experts), the kind of advice that MBIFM generated was compared with the kind of advice given by the human experts.

In 107 out of the 145 actions that MBIFM was rightfully alerted, the advice proposed by the human experts had been among the pieces of advice generated by the system. In another 30, MBIFM generated advice that was proposed by a minority of human experts. However, one important aim of the particular experiment was to check the efficiency of the decision-making model that had been used. Therefore, much attention was paid to the evaluation of the efficiency of the ranking of alternative actions to be suggested to a user by the system. The application of TOPSIS in MBIFM managed to select the same advice as the human experts in 82 out of the 107 cases where MBIFM succeeded to identify the alternative action that the majority of human experts proposed. This means that the degree of success for the particular theory is 76.64 percent, which is considered to be rather satisfactory. What seems also rather important is that the application of TOPSIS increased the effectiveness of the system to great extent. Indeed, if the IUI did not incorporate a decision-making model, then the corresponding degree of success would be just 45.79 percent.

If the system using a particular theory can reproduce human advisor's advice to a satisfactory extent, then it is selected. As Agah and Tanie [29] point out, such a system could make mistakes in determining the implicit intentions of the human user; but, that is the case with any intelligent system trying to understand human intents. Even humans cannot be 100 percent successful in such a task. This is one important reason why MBIFM's advice is compared to that of the human advisor.

9 Discussion and Conclusions

In this paper, we have described a general framework of how a multicriteria decision-making theory can be incorporated in an Intelligent User Interface. This framework is called the MBIUI life-cycle framework. The IUI that has been developed as a test bed for our research is called MBIFM and is a file manipulation system that works in a similar way as Windows/NT Explorer but constantly reasons about every user's action and provides spontaneous advice, in case this is considered necessary. The particular domain of the user interface was selected because it is targeted to a very large number of computer users of varying backgrounds. The need for intelligent support in such user interfaces is greater than in other kinds of user interfaces that are targeted to users of more restricted backgrounds.

As a basis for our MBIUI framework we have used RUP. The particular object-oriented software life-cycle model was selected because it advocates multiple iterations of the development process. The particular software life-cycle model seems very promising, especially for the adaptation of multicriteria decision-making theories, which require several experiments for their adaptation in user interfaces.

In view of the above, the MBIUI life-cycle framework that was presented in this paper, describes in detail at which stage of the software life-cycle and in what way the experimental studies should be conducted for a multicriteria decision-making theory to be adapted in an IUI. The main characteristic of the proposed framework is the multiple iterations in the software life-cycle. These iterations are very important for the successful adaptation and refinement of a decision-making theory so as to give the best results for a particular domain.

According to the MBIUI life-cycle framework, in the inception phase, during requirements capture, a usability evaluation of a standard system, that does not incorporate intelligence, is conducted. The information collected by this process is further used in order to specify the requirements for intelligent help and develop a user interface prototype. In the same phase, during the analysis of the system, two different experiments should be conducted for a multicriteria decision-making theory to be adapted in a user interface. The first experiment aims at capturing the criteria that human advisors take into account while helping users in their interaction with computers and the second aims at determining the weights of importance of these criteria. The first experiment should involve a satisfactory number of human experts (16 experts) that have sufficient experience related to the domain being reviewed. These experts are asked about the criteria that they used to take into account when providing individualized advice in that domain. The first experiment is essential for the application of any multicriteria decision-making theory since multicriteria decision-making theories provide exact mathematical models but do not define the criteria.

Similarly, the setting of the second experiment described in the context of MBIFM, can be used as it is for the application of many multicriteria decision-making theories, such as TOPSIS and SAW. However, there are also theories that require a different kind of experiment for the calculation of the weights of importance of the criteria but these theories embody the setting of the experiment. Such theory is MAUT. Finally, there are theories (e.g., DEA) that support dynamical calculation of the weights of the criteria and in such theories, the second experiment could be completely omitted.

An important problem in the application of any multicriteria decision-making theory is how the system collects the information that is used for the calculation of the values of the criteria. This problem can be incorporated in the user modeling mechanism of the IUI and such decisions are made during the design of the system, in the elaboration phase. In this paper, MBIFM uses an individual user model that is constantly updated by information that is gathered implicitly during the user's interaction with the system. This approach was selected because the system may collect information about the users that interact with it without annoying them by asking too many questions concerning their habits, preferences, goals, etc. After solutions have been given to the above mentioned problems, the adaptation of the multicriteria decision-making theory for the particular domain takes place.

As soon as the final product is ready, emphasis should be given on its evaluation, which is conducted in transition, during testing. Though multiple evaluations are also advocated by user interface designers [30], [31], [32], in the case of knowledge-based user interfaces that incorporate user modeling techniques, evaluations are often neglected completely [26]. This may have disastrous consequences on the system's overall credibility and effectiveness.

When a user interface tries to simulate the reasoning of human users or experts by incorporating a decision-making theory, its evaluation is an essential part of the adaptation of the theory. The evaluation of a decision-making theory shows whether the criteria that have been specified and used in the IUI as well as their weights are successful with respect to the operation of the user interface. Despite the importance of the empirical evaluation, systems incorporating decision-making theories are often not empirically evaluated [33]. In MBIUI, the experiment that is specified as more appropriate for evaluating such theories is comparing the outcome of a user interface that incorporates a decision-making theory with the reasoning of human experts that watch users' interaction with a similar "nonintelligent" user interface and provide spontaneous advice.

In view of the above, MBIUI advocates the evaluation of the particular decision-making theory that has been used by comparing the output of the reasoning of the IUI to that of human advisors. More specifically, the evaluation of an IUI should involve real users of the system as well as human advisors. The users interact with a standard system that does not incorporate a decision-making theory. The protocols collected are given to human advisors to comment on them as well as to the IUI and its responses are compared to the human advisors' responses. If the results of this comparison are satisfactory, then the selection of the particular decision-making theory and the criteria that it involves are considered appropriate. Otherwise, another iteration of the software's life-cycle takes place. The results of the evaluation experiment of the IUI that we developed were quite satisfactory and, therefore, TOPSIS seems adequate for our system and is selected.

REFERENCES

[1] F. Naumann, "Data Fusion and Data Quality," *Proc. Conf. New Techniques and Technologies for Statistics*, 1998.

[2] W. Schütz and R. Schäfer, "Bayesian Networks for Estimating the User's Interests in the Context of a Configuration Task," *Proc. UM2001 Workshop Machine Learning for User Modeling*, pp. 23-36, 2001.

[3] T. Bohnenberger, O. Jacobs, A. Jameson, and I. Aslan, "Decision-Theoretic Planning Meets User Requirements: Enhancements and Studies of an Intelligent Shopping Guide," *Pervasive Computing: Proc. Third Int'l Conf.*, pp. 279-296, 2005.

[4] D. Chin and A. Porage, "Acquiring User Preferences for Product Customization," *Proc. Eighth Int'l Conf. User Modeling*, pp. 95-104, 2001.

[5] D. Kudenko, M. Bauer, and D. Dengler, "Group Decision Making Through Mediated Discussions," *Proc. Ninth Int'l Conf. User Modeling*, 2003.

[6] S. Delisle and B. Moulin, "User Interfaces and Help Systems: From Helplessness to Intelligent Assistance," *Artificial Intelligence Rev.*, vol. 18, no. 2, pp. 117-157, 2002.

[7] C.L. Hwang and K. Yoon, "Multiple Attribute Decision Making: Methods and Applications," *Lecture Notes in Economics and Math. Systems*, vol. 186, 1981.

[8] P.C. Fishburn, "Additive Utilities with Incomplete Product Set: Applications to Priorities and Assignments," *Operations Research,* 1967.
[9] P. Vincke, *Multicriteria Decision-Aid.* Wiley, 1992.
[10] W.W. Cooper, L.M. Seiford, and K. Tone, *Data Envelopment Analysis.* Kluwer Academic, 1999.
[11] M.E.C. Hull, P.S. Taylor, J.R.P. Hanna, and R.J. Millar, "Software Development Processes—An Assessment," *Information and Software Technology,* vol. 44, pp. 1-12, 2002.
[12] M. Del Socorro Bernardos, "Guideline for Developing a Software Life Cycle Process in Natural Language Generation Projects," *Computational Linguistics and Intelligent Text Processing, Lecture Notes in Computer Science,* vol. 2945, pp. 355-359, 2004.
[13] IEEE Std. 1074-1997, IEEE Standard for Developing Software Life Cycle Processes, 1997.
[14] C. Bernon, M.P. Gleizes, S. Peyruqueou, and G. Picard, "ADELFE: A Methodology for Adaptive MultiAgent Systems Engineering," *Eng. Soc. in the Agents World III, Lecture Notes in Artificial Intelligence,* vol. 2577, pp. 156-169, 2003.
[15] I. Jacobson, G. Booch, and J. Rumbaugh, *The Unified Software Development Process.* Addison-Wesley, 1999.
[16] D.F. D'Souza and A.C. Willis, *Objects, Components, and Frameworks with UML: The Catalysis Approach.* Addison-Wesley, 1999.
[17] I. Graham, B. Henderson-Sellers, and H. Younessi, *The OPEN Process Specification.* Addison-Wesley, 1999.
[18] T. Matsuo and T. Ito, "A Designated Bid Reverse Auction for Agent-based Electronic Commerce," *Proc. 15th Int'l Conf. Industrial and Eng. Application of Artificial Intelligence and Expert Systems,* 2002.
[19] L. Swartz, "Why People Hate the Paperclip: Labels, Appearance, Behavior and Social Responses to User Interface Agents," BS thesis, Stanford Univ., 2003.
[20] M. Virvou and K. Kabassi, "Reasoning about Users' Actions in a Graphical User Interface," *Human-Computer Interaction,* vol. 17, no. 4, pp. 369-399, 2002.
[21] M. Virvou and K. Kabassi, "Experimental Studies within the Software Engineering Process for Intelligent Assistance in a GUI," *J. Universal Computer Science,* vol. 8, no. 1, pp. 51-85, 2003.
[22] M. Virvou and K. Kabassi, "Adapting the Human Plausible Reasoning Theory to a Graphical User Interface," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 34, no. 4, pp. 546-563, 2004.
[23] E. Triantaphyllou and S.H. Mann, "An Examination of the Effectiveness of Four MultiDimensional Decision-Making Methods: A Decision-Making Paradox," *Int'l J. Decision Support Systems,* vol. 5, pp. 303-312, 1989.
[24] R. Kass and T. Finin, "The Role of User Models in Cooperative Interactive Systems," *Int'l J. Intelligent Systems,* vol. 4, pp. 81-112, 1989.
[25] M.F. McTear, "Intelligent Interface Technology: From Theory to Reality?" *Interacting with Computers,* vol. 12, pp. 323-336, 2000.
[26] D.N. Chin, "Empirical Evaluation of User Models and User-Adapted Systems," *User Modeling and User Adapted Interaction,* vol. 11, nos. 1-2, pp. 181-194, 2001.
[27] K. Kabassi and M. Virvou, "Combination of a Cognitive Theory with the Multi-Attribute Utility Theory," *Proc. Knowledge-Based Intelligent Information and Engineering Systems (KES '03),* 2003.
[28] Y. Zhu and A. Buchman, "Evaluating and Selecting Web Sources as External Information Resources of a Data Warehouse," *Proc. Third Int'l Conf. Web Information Systems Eng. (WISE '00),* pp. 149-160, 2000.
[29] A. Agah and K. Tanie, "Intelligent Graphical User Interface Design Utilizing Multiple Fuzzy Agents," *Interacting with Computers,* vol. 12, pp. 529-542, 2000.
[30] D.R. Olsen Jr., *Developing User Interfaces.* Morgan Kaufmann, 1998.
[31] B. Shneiderman, *Designing the User Interface: Strategies for Effective Human-Computer Interaction,* third ed. Addison-Wesley, 1998.
[32] I. Sommerville, *Software Engineering.* Addison-Wesley, 1992.
[33] R.S. Sojda, "Empirical Evaluation of Decision Support Systems: Basic Concepts and an Example for Trumpeter Swan Management," *Proc. Int'l Conf. Int'l Environmental Modeling and Software Society iEMSs,* 2004.

Katerina Kabassi received a degree in informatics (1999) and the DPhil degree (2003) from the Department of Informatics, University of Piraeus, Greece. She has been elected as a faculty member in the Department of Ecology and the Environment, Technological Educational Institute of the Ionian Islands, Greece, and she is currently a postdoctoral research fellow in the Department of Informatics, University of Piraeus, Greece. She has authored more than 37 articles, which have been published in international journals, books, and conferences. She has served as a member of program committees and/or reviewer of international conferences. Her current research interests are in the areas of knowledge-based software engineering, human computer interaction, personalization systems, multicriteria decision making, user modeling, Web-based information systems, and educational software.

Maria Virvou received a degree in mathematics from the University of Athens, Greece, in 1986, the MSc degree in computer science from the University of London (University College London), United Kingdom, in 1987, and the DPhil degree from the School of Cognitive and Computing Sciences at the University of Sussex, United Kingdom, in 1993. She is an associate professor in the Department of Informatics at the University of Piraeus, Greece. She is the sole author of three computer science books: *Object Oriented Software Engineering*, *Object Oriented Analysis and Design*, and *Introduction to Compilers*. She has authored or coauthored more than 150 articles, which have been published in international journals, books, and conferences. The international journals and conferences where she has published articles are in the areas of knowledge-based software engineering, Web-based information systems, computers and education, personalization systems, human computer interaction, and student/user modeling. She has served as a member of program committees and/or reviewer of international journals and conferences. She has supervised or is currently supervising 12 PhD candidates in the areas of Web-based information systems, knowledge-based human computer interaction, personalization systems, software engineering, e-learning, e-services, and m-services. She has served or is serving as the project leader and/or project member in 15 R&D projects in the areas of e-learning, computer science, and information systems. In the year 2003-2004, she received the first research award from the Research Center of her University as the member of faculty (among 200 colleagues of hers) having the highest number of research publications in high-quality research journals. Many articles of hers have been among the top five most downlowded articles of the respective journals where they were published. She is a member of the IEEE.