Personalised adult e-training on computer use based on multiple attribute decision making

Katerina Kabassi, Maria Virvou*

Department of Informatics, University of Piraeus, 80 Karaoli and Dimitriou St, Piraeus 185 34, Greece

Received 20 June 2003; accepted 5 November 2003

Abstract

This paper examines the utility of a multiple attribute decision making method, the Simple Additive Weighting (SAW), for the purposes of an Intelligent Learning Environment (ILE) that provides adults with personalised e-learning. The ILE is called Web Intelligent Trainer and is meant to help novice users learn how to manipulate the file store of their personal computer. The generation of advice makes use of adaptive hypermedia techniques and is adapted to each individual learner’s needs, depending on their knowledge level, age, habits and difficulties. SAW has been applied in the ILE and has been evaluated with respect to the performance of the ILE. As a result, SAW seems particularly appropriate for the ILE.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Adult e-learning; Human–computer interface; Intelligent tutoring systems; Information technology skills

1. Introduction

The advances of Information Technology have affected the way that most kinds of professional work is performed. Digital literacy and media sophistication is becoming necessary to succeed in every aspect of life; even jobs that were once considered fairly routine (e.g. secretaries, farmers) have come to require complex information processing (Collins et al., 2000). Thus, people today need to have education on computer science on top of other more traditional domains. This is also the case for many adults who have not had any opportunity to receive any computer education when they were at school or university because computer science courses were not included in the curriculum then.
These adults usually have two main problems associated with the computer training that they have to receive. First, they are probably professionals who may not have time to spare to take courses at institutions far from their work or home. Second, they may be completely unfamiliar with computers and at times they may even be computer phobic. The older they are the less familiar they probably are with computers, because the culture of computers was not so widespread in the past as it is now. In such cases, learning takes place in a slower pace than it would for younger people.

An excellent solution for these problems is the facility of personalised e-training, which may take place anywhere, at any time and without necessarily the presence of a human teacher. Thus, e-training can take place at people’s work or at home, at the time available. Moreover, the personalisation can ensure that the system will take into account the particular strengths and weaknesses of each individual who is using the program. It is simple logic that response personalised to a particular student must be based on some information about that student; in the area of Artificial Intelligence in Education, this realisation led to student modelling, which became a core or even defining issue for the field (Cumming and McDougall, 2000). As a result, student modelling is used in Intelligent Tutoring Systems (ITSs), Intelligent Learning Environments (ILEs) and Adaptive Hypermedia Systems (AHS) to achieve personalisation and dynamic adaptation to the needs of each individual student.

In the case of Information Technology domains, ILEs provide environments that can monitor users while they work with the computer, help them perform their tasks and provide them with feedback in a manner that contributes to their learning process. Moreover, in Information Technology domains, student modelling can be performed in a more accurate way than in other domains. This is so because Information Technology skills of students have the advantage of being observable by computers. Therefore, when users interact with a computer, they provide a great deal of information about themselves. This information can be collected and stored in a user model in order to be further processed and used to adapt the system’s interaction to each individual user.

In view of the above, we have developed Web Intelligent Trainer (Web-IT), a Web-based ILE for novice adult users of a Graphical User Interface (GUI) that manipulates files, such as the Windows 98/NT Explorer (Microsoft Corporation, 1998). Web-IT provides a protected environment to novice users who can work, as they would normally do, while the system silently reasons about their actions and offers adaptive tutoring. For this reason, Web-IT stores information about each learner centrally on a Learner Modelling Server (Kabassi and Virvou, 2003).

Web-IT’s tutoring is based on the ‘relevance principle’ of Sperber and Wilson (1986), which states that humans pay attention only to information that seems relevant to them. Therefore, the system uses the information about each user that is stored in the Learner Modelling Server, in order to adapt the tutoring process to the specific user. However, the effectiveness of the system depends on the system’s ability to make good decisions about when and what the learner should learn. Such a decision is based on certain attributes of the learner that are stored in a detailed learner model. In ILEs that were developed prior to Web-IT (Virvou and Du Boulay, 1999; Virvou and Kabassi, 2001) such decisions were based on a cognitive theory, called Human Plausible Reasoning (Collins and Michalski, 1989). However, the theory did not have a formal way for calculating the weights of
the attributes. In the research described in this paper, a solution to this problem was given by the use of the Multiple Attribute Decision Making (MADM). MADM involves making preference decisions (such as evaluation, prioritisation, selection) over the available alternatives that are characterised by multiple, usually conflicting attributes (Hwang and Yoon, 1981).

Although most of the problems in education are multi-attribute ones, MADM has not been exploited for that domain. Therefore, we have examined the utility of the MADM method called SAW in the context of the ILE. This method was selected in the first place due to its simplicity and effectiveness in solving diverse problems in virtually any topic, for example public policy making (Massam, 1999), medical science (Azar, 2000), etc. This method has also been used for solving MADM problems in computer science. For example, Naumann (1998) uses the SAW method in order to select the best information source when a user submits a query. In the case of Web-IT, which is an ILE, this method has been used for the automatic selection of the most appropriate advice to be given to a user who is having problems with the interaction. After its completion Web-IT was evaluated and the results were analysed in order to test the validity and robustness of the SAW method in educational problems.

2. Background

2.1. Multi-attribute decision making

A multi-attribute decision problem is a situation in which, having defined a set $A$ of actions and a consistent family $F$ of $n$ attributes $g_1, g_2, \ldots, g_n$ ($n \geq 3$) on $A$, one wishes to rank the actions of $A$ from best to worst and determine a subset of actions considered to be the best with respect to $F$ (Vince, 1992). According to Triantaphyllou and Mann (1989) there are three steps in utilising any decision making technique involving numerical analysis of alternatives:

1. Determining the relevant attributes and alternatives.
2. Attaching numerical measures to the relative importance of the attributes and to the impacts of the alternatives on these attributes.
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’s 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.

Yoon and Hwang (1995) provide an excellent review of MADM methods, presenting them in two distinct categories: compensatory methods that allow the stronger attributes of an option to make up for its weaknesses, and non-compensatory methods that do not permit such tradeoffs. In non-compensatory methods, comparisons are made on an attribute-by-attribute basis, which does not really correspond to the human decision-making process. Therefore, in the case of an ILE the compensatory methods seem more
promising. The most basic compensatory methods calculate weighted scores for each option, using Simple Additive Weights (SAW) (Fishburn, 1967; Hwang and Yoon, 1981).

The SAW (Fishburn, 1967; Hwang and Yoon, 1981) method is probably the best known and most widely used decision making method. SAW consists of two basic steps:

1. **Scale the values of the n attributes to make them comparable.** There are cases where the values of some attributes take their values in [0,1] whereas there are others that take their values in [0,1000]. Such values are not easily comparable. A solution to this problem is given by transforming the values of attributes in such a way that they are in the same interval.

2. **Sum up the values of the n attributes for each alternative.** As soon as the weights and the values of the n attributes have been defined, the value of a multi-attribute function is calculated for each alternative as a linear combination of the values of the n attributes.

Scaling the values of the n attributes to make them comparable is the first step in applying the SAW method. However, the way that the values of the attributes should be scaled up is not defined by SAW and, therefore, different approaches have been proposed in order to overcome this problem. The one proposed by Naumann (1998) is a very good one as it transforms all values so that they take their values in the interval [0,1] and has been applied in solving decision making problems in computer science. More specifically, Naumann (1998) uses the scaling factor of formula (1) in order to normalise the values of attributes such as understandability, extent, availability, time and price

\[
x_{ij} = \frac{d_{ij} - d_{j}^{\text{min}}}{d_{j}^{\text{max}} - d_{j}^{\text{min}}}
\]

where \(d_{ij}\) is the old value and \(x_{ij}\) is the transformed value of the \(j\) attribute for the \(i\) alternative and \(d_{j}^{\text{max}}\), \(d_{j}^{\text{min}}\) are the maximum and minimum values of the \(j\) attribute for all the alternatives. Using this scaling the values of all attributes are in the interval [0,1].

The SAW approach consists of translating a decision problem into the optimisation of some multi-attribute utility function \(U\) defined on \(A\). The decision maker estimates the value of function \(U(X_j)\) for every alternative \(X_j\) and selects the one with the highest value. The multi-attribute utility function \(U\) can be calculated in the SAW method as a linear combination of the values of the \(n\) attributes

\[
U(X_j) = \sum_{i=1}^{n} w_i x_{ij}
\]

where \(X_j\) is one alternative and \(x_{ij}\) is the value of the \(i\) attribute for the \(X_j\) alternative.

### 2.2. ILEs and adaptive support

The key to effective adaptation in ITSs or ILEs is student modelling. Student modelling is an important task but has been recognised as rather complex and difficult. Therefore, several AI methods and approaches have been proposed in order to serve the learner modelling procedure. For example, machine learning techniques have been extensively
used in ITSs and ILEs to automatically extend the background knowledge of learners as well as to automatically induce the learner model (Sison and Shimura, 1998). A quite different approach is adopted by systems using Bayesian Networks (Petrushin and Sinitsa, 1993; Martin and VanLehn, 1995). More specifically, the use of Belief Bayesian Networks entails the development of a model of how students actually reason, identify their misconception and provide adaptive tutoring and instruction (Duncan et al., 1994).

Most of the methods used for adaptation in ITSs and ILEs focus on the generation of hypotheses about the learner’s reasoning. However, in an ITS or ILE what is even more important and has been overlooked in the related literature is identifying how human tutors reason while they teach a certain domain. In this respect, MADM methods may be extremely useful in ITSs and ILEs in order to model the decision making process of human tutors.

Decision processes with multiple attributes deal with human judgment, which is not easy to model. Furthermore, as it was shown in Triantaphyllou and Mann (1989) MADM methods can give different answers to the same problem. Since the truly best alternative is the same, regardless of the method chosen, an estimation of the accuracy of each method is highly desirable. It is our goal to examine how SAW might be applied in Web-IT and find out how appropriate it is in modelling human tutors’ reasoning while evaluating different pieces of advice that could be given to students. Such pieces of advice suggest to a user further reading on parts of the theory that are relevant to a possible problem that has occurred to a learner during his/her interaction with the system. SAW is used in order to rank the set of all the candidate parts of the theory and thus select the best one to suggest to a user. After deciding what the user should learn, the system uses adaptation techniques in order to adapt the lesson to each user’s characteristics, needs and level of knowledge.

Additionally, for the adaptation of teaching and learning adaptive hypermedia methods and techniques provide a very good solution. These techniques use information about a particular learner stored in the student model and the adaptivity of learning depends on factors such as the learner’s habits, prior knowledge and skills. Indeed, educational systems such as InterBook (Brusilovsky et al., 1998) and AHA! (De Bra and Calvi, 1998) use adaptive hypermedia to guide users through the teaching material.

The two main adaptive hypermedia techniques that exist are: (i) adaptive presentation, where adaptation is performed at the content level and (ii) adaptive navigation support, which is performed at the link level (Brusilovsky, 1996). Both these technologies have been evaluated and the results offer strong evidence that their use in an AHS can improve human–computer interaction (e.g. De Bra et al., 1999; Murray et al., 2000). Web-IT uses both adaptive presentation and adaptive navigation support to tailor the information presented to the user.

3. Web intelligent trainer

Web-IT is an ILE for novice adult users of a GUI that manipulates files, such as the Windows 98/NT Explorer (Microsoft Corporation, 1998). Web-IT is meant to help users during their navigation and manipulation of the file store and provide adaptive tutoring in case this is considered necessary. In general, Web-IT tries to act as a trainer that helps
adults while interacting with the computer at work or at home and constantly helps them by presenting them pieces of knowledge they are not very familiar with.

Every time a user issues an action, Web-IT reasons about it in terms of the system’s expectations about the user’s recognised goals. In case this action contradicts the system’s expectations, it tries to identify the user’s misconception and provide adaptive tutoring. Web-IT uses adaptive hypermedia techniques to protect learners from information overflow and to help them understand new pieces of knowledge that are being taught.

In Web-IT, adaptive presentation techniques are used to present examples of use of an unknown command in the context of the learner’s own file-store. Therefore, the system generates examples dynamically, which use the names of the particular learner’s existing files and folders. More specifically, Web-IT selects the names of files and/or folders that are frequently used by the learner to make sure that the learner is very familiar with them.

Moreover, Web-IT uses adaptive link annotation techniques to present other parts of knowledge that are believed to be of interest to the learner for the particular case. The idea of adaptive link annotation is to augment links with some form of comments, which can tell the user more about the current state of the nodes behind the annotated links. Web-IT uses different font types and icons to provide adaptive navigation support. Whenever a link appears on a page, the font type and the icon that appears in front of the link are annotated so as to reflect the state of the topic or exercise behind the link, with respect to the student’s current knowledge state. In particular, the system recommends the links that are in bold letters and have an arrow pointing to them. Furthermore, links in normal fonts that have a cycle arrow in front of them represent links that a user has already visited but needs to revise. The links that have an exclamation mark in front of them and are italicised are considered to be known by the user. Finally, the links that have smaller fonts and a ‘halt’ sign in front correspond to topics that are not considered ready for the user to visit. Examples of links are shown in Fig. 1.

A simple example of the system’s operation taken from a real interaction of a user with Web-IT is presented below. The user of our example is a 37-year-old female adult that is novice in the usage of computers and file manipulation programs. A typical task of the user is presented in this example. The user’s initial file store state is shown in Fig. 2. This user has been given by her employer a CD-ROM that contains useful information about two projects that have been assigned to her. Now she wants to copy the files ‘Project1.doc’ and ‘Project2.doc’ from the CD-ROM to the folder ‘C:\Projects\’. In order to achieve her goal, the user selects the CD-ROM (D:\) and executes the command `copy`. Indeed, `copy` is the first command in the sequence of commands `copy–paste` that a user has to execute in order to copy objects. However, the user has not selected the objects she wants to copy. Then she selects C:\Projects\ and executes the command `paste`. In such cases, in a standard Explorer, the command `copy` fails and the user is prompted with an error message such as the one presented in Fig. 3 without any further notice or instruction. In contrast, Web-IT finds the particular action as ‘Unintended’ and tries to diagnose the user’s misconception. Therefore, it applies MADM and concludes that the user needs more tutoring on copying objects.

As a result, the user is presented with a lesson that involves the parts of the theory ‘Copying Files’ and she is advised to revise the parts of theory ‘Copying Objects in the file store’ and ‘Copying Folders’ (Fig. 1). Furthermore, the user is provided with an example
Fig. 1. A sample screen for the tutoring of 'Copying Files'.
about copying a file, which is adapted to the user’s own file store. More specifically, the user is shown how to copy the file ‘Project1.doc’ from CD_ROM D:\ to the folder C:\Projects\. More information about the system’s reasoning and how it draws inferences about the user are given in the following sections.

4. Personalisation in Web-IT

4.1. Specifying the attributes and their weights based on stereotypes of age

The main goal of Web-IT is to model the decision processes of a human tutor, who watches the user over the shoulder and provides adaptive tutoring and instruction in case this is considered necessary. More specifically, the decision process involves
the evaluation of each part of the theory to be taught with respect to its relevance with the current user’s problem and the necessity for it to be presented to the user.

To model the behaviour of human tutors, we needed first to specify the attributes that human tutors take into account when making decisions about what the students should learn. Then, we needed to specify the relative importance of these attributes according to human tutors. Therefore, we conducted an empirical study involving human tutors of file manipulation programs (Virvou and Kabassi, 2001). In particular, 20 human tutors were asked about the attributes that they take into account when making decisions about what the student should learn. After analysing their answers, we concluded that the attributes that most human tutors take into account in order to provide personalised help and tutoring are:

- The degree of relevance of the user’s problematic action to each topic of the theory ($e$). The value of this attribute shows how relevant the action that the user intended to issue is with the action involved in a particular piece of the theory to be taught.
- The number of times that the user has visited each part of the theory ($v$). If a user has visited a part of the theory many times, it is likely that s/he knows the commands connected with that part, therefore s/he may not need further tutoring on this part.
- The number of the wrong executions of a command ($c$). The higher this number the greater the need for the particular user to visit the part of the theory that is connected with this command.
- The degree of difficulty of the theory ($d$). It has been observed that some parts of the theory are not easily comprehensible by the user. Therefore, the higher the degree of difficulty of a theory the higher the need for tutoring on that particular subject.

After specifying the attributes that the system should take into account in order to make a decision about what the student should learn, the human tutors were asked to define the relative importance of these attributes. However, as human tutors revealed, the same attribute does not have the same importance in making decisions about the learning process of people who are differently aged. An attribute that is very important for children may not be that important for adults. Therefore, in Web-IT, which aims at training adults of different ages as well as children, users are categorised to different stereotypes according to their age. Stereotypes (Rich, 1989, 1999) are used in user modelling in order to provide default assumptions about users belonging to the same category according to a classification of users that has previously taken place. In the case of Web-IT, default assumptions include the weight of importance of each attribute for a user belonging to a certain age group.

Stereotypes constitute a common user modelling technique for drawing assumptions about users belonging to different groups. However, it is not very common for stereotypes to be used in conjunction with a MADM theory. In one case though, in a system called Travel Planner (Chin and Porage, 2001), stereotypes were used in conjunction with the Multi-Attribute Utility Theory, which is a MADM theory. In that system, similarly to Web-IT, stereotypes were used to draw assumptions about the weights of several attributes. However, Travel Planner used a different MADM theory from Web-IT.
Moreover, the context of that system referred to travelling preferences of remote customers rather than tutoring needs of remote learners of varying ages.

In Web-IT, users have been categorised to four stereotypes according to their age. The first stereotype includes all users that are younger than 18-years-old. The second contains all adults that are younger than 30-years-old and older than 18. The users that belong in the third stereotype are older than 30 but younger than 50. Finally, the fourth stereotype consists of all users that are older than 50-years-old.

In view of the above, human tutors were asked to rank the four attributes for each group of users with respect to how important these attributes were in their reasoning process. For each stereotype, every one of the human tutors was asked to assign one score of the set of scores (1, 2, 3, 4) to each one of the four attributes and not the same one to two attributes. The sum of scores of the elements of the set of scores was 10 (1 + 2 + 3 + 4 = 10). For example, for the stereotype of users that are between 30 and 50, a human tutor could assign the score 4 on the number of visits in each part of the theory (attribute $v$), the score 3 on the degree of relevance (attribute $e$), the score 2 on the degree difficulty of the theory (attribute $d$) and the score 1 on the number of wrong executions of a command (attribute $c$). However, the same human tutor could assign different scores to the respective attributes for a different stereotype (e.g. the stereotype of users that are under 18-years-old). Indeed, as the empirical study revealed, the attribute about the number of times that a user has visited each part of the theory was considered more important for adults than for children or teenagers as a teenager user might learn a concept immediately after having read the theory just once. However, an adult may have read the theory many times and still not know the concept because adults tend to be more forgetful and have less assimilation ability than young learners. In addition, the human tutors thought that the attribute $d$, which corresponds to the degree of difficulty of a part of theory was also more important for adult users. Indeed, the more difficult a theory part, the less comprehensible this theory part for adult users, whereas children at the same time may find it rather understandable.

As soon as the scores of all human tutors for each stereotype were collected, they were used to calculate the weights of the attributes. The scores assigned to each attribute had been summed up and divided by the sum of scores of all attributes (10 × 20 human experts = 200). In this way, the sum of all weights could be equal to 1.

The above process resulted in the calculation of the weights of the above mentioned attributes for each stereotype. Table 1 presents the weights of the different attributes for every stereotype of users that interact with the system. As a result, it was revealed that the most important attribute that human tutors had used when they evaluated candidate topics of the theory for all users was the degree of relevance of the user’s action to each topic of the theory ($e$). This was due to the fact that the more relevant a topic of the theory is to the user’s action the more likely it is for it to be of the user’s interest. For example, the weight of the particular attribute that represents its relative importance is 0.30 for the users that were between 19 and 30-years-old. The particular weight had been resulted from the division of 60, which was the sum of the scores that the human experts assigned to the attribute, by 200, which was the sum of scores of all attributes. The same weight had also been assigned in the particular attribute for users that were between 31 and 50. However, the weight was higher for children and teenagers (0.33) and lower for adult users (0.30 for users aged 19–50 and 0.28 for users older than 50-years-old).
The second most important attribute that human tutors had used in their reasoning process about users that were younger than 30-years-old was the number of the wrong executions of a command ($c$). The higher this number the greater the need for the particular user to visit the theory that is connected with this part of the theory. However, the importance of the attribute is lowered, as the users get older. In the case of people that are older than 30-years-old, the human tutors thought that the attribute about the difficulty of a part of a theory was more important than the number of errors that a user may make. This is probably due to the fact that people tend to forget what they have learned as they get older. Therefore, the difficulty of a part of the theory is among the most important attributes for people that are older than 50-years-old whereas is less important for younger users.

Finally, the human tutors also took into account the number of times that the user had visited each part of the theory ($v$). That attribute was not considered that important for younger users as one may know a command without having visited the theory topic many times. However, the particular attribute was more important for older users. This was due to the fact that as users get older they have to read more and more a part of a theory before they can learn everything about it.

### 4.2. Values of attributes

The main feature of Web-IT is that it can adapt its interaction to each individual learner. In order to do so, Web-IT constantly observes the student while s/he is actively engaged in his/her usual activities and records the information collected in a learner model. The learner models are maintained centrally on a Web Server and the communication with Web-IT is performed through Web Services protocols. Web Services, in the general meaning of the term, are services offered via the Web. Lately, though this term refers to a collection of specific protocols regarding remote application interaction. More specifically, Web Services are self-contained, modular applications that provide a set of functionalities to anyone that requests them.

In case the system suspects that a user is involved in a problematic situation, it provides spontaneous and individualised advice and tutoring based on the information that is stored in the learner model. More specifically, the learner model contains information about the user’s activities, plans and goals as well as detailed information about the learner’s strengths and weaknesses. The information that is stored in the learner model is used to calculate the values of the attributes that the system takes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Younger than 18</th>
<th>19–30</th>
<th>31–50</th>
<th>Older than 51</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e$</td>
<td>0.33</td>
<td>0.30</td>
<td>0.30</td>
<td>0.28</td>
</tr>
<tr>
<td>$c$</td>
<td>0.3</td>
<td>0.29</td>
<td>0.25</td>
<td>0.24</td>
</tr>
<tr>
<td>$d$</td>
<td>0.23</td>
<td>0.26</td>
<td>0.29</td>
<td>0.28</td>
</tr>
<tr>
<td>$v$</td>
<td>0.14</td>
<td>0.15</td>
<td>0.16</td>
<td>0.20</td>
</tr>
</tbody>
</table>
into account while making decisions about which theory topic should be proposed to him/her.

The system constantly keeps detailed information about the visits of the user to each part of the theory. This information is further used by the system to calculate the value of the attribute that corresponds to the number of times that the user has visited each part of the theory \((v)\). Every time a user executes a command not correctly (e.g. the learner tried to execute a copy command without having selected an object) then Web-IT updates the information of the learner model that refers to the particular command. This information is further used by the system in order to calculate the value of the attribute \(c\), which corresponds to the number of wrong executions of a command.

The only attributes that do not take their values from the information that is stored in the learner model is the degree of relevance of the user’s action to the each topic of the theory and the degree of difficulty of that topic. The value of the degree of relevance is dynamically estimated based on the similarity of actions that is acquired by the domain representation component. The domain representation component contains a hierarchical representation of the domain being taught. The value of the degree of difficulty is predefined and has been estimated by using as input the opinion of human tutors that participated the empirical study.

For the user of the example in Section 3, the system found three possible theory topics that the user should learn. The three alternatives \((X_1, X_2, X_3)\) corresponded to the topics ‘Copying Files’, ‘Copying Folders’ and ‘Copying Objects’, respectively. The relevance of the command that the user had executed with the part of the theory ‘Copying Files’ was one \((e = 1)\). Furthermore, the relevance between the command issued by the user and the part of the theory ‘Copying Objects’ and the part ‘Copying Folders’ was 0.95 and 0.90, respectively.

The values of the other attributes were given by the information that was kept in the individual learner model. The user had visited the topics ‘Copying Files’ and ‘Copying Folders’ once \((v = 1)\) and the theory ‘Copying Object’ just twice \((v = 2)\). Furthermore, the user had been mistaken in copying a file 10 times \((c = 10)\). That meant that the user did not probably know how to copy files in the file store. The user had also been mistaken in copying folders once \((c = 1)\).

The degree of difficulty for the topic ‘Copying Files’ is 0.40. The topic ‘Copying Folders’ is slightly more difficult and, therefore, its degree of difficulty is 0.50. However, the topic ‘Copying Objects’ is considered less difficult than all the others as it contains only the basic principles of copying objects \((d = 0.30)\). In view of the above the values of the attributes are presented below:

\[
D = \begin{pmatrix}
X_1 & 1.00 & 1 & 10 & 0.40 \\
X_2 & 0.90 & 1 & 1 & 0.50 \\
X_3 & 0.95 & 2 & 11 & 0.30
\end{pmatrix}
\]

As soon as the values of the attributes have been defined, the system calculates the value of a multi-attribute utility function \(U\) for each theory topic. The theory topic that maximises
the function $U$ is selected by Web-IT in order to be presented to the user. For the calculation of the value of the function $U$ we have used the SAW theory. The way that this method has been applied in the context of Web-IT in Section 5.

5. The SAW method in Web-IT

The decision problem in Web-IT is to find the theory topic that is most appropriate to be presented to a user so that s/he may acquire the piece of knowledge needed for him/her to achieve his/her initial goals. This problem is addressed in the following way: first, the system generates an action other than the one issued by the user, which was problematic. The generated action should be what the user really intended instead of the problematic action issued. Then, the system relates the generated action with a part of the theory that the user should read so that s/he acquires the pieces of knowledge that s/he needed for the use of this intended action. Thus the decision problem in Web-IT requires the calculation of a multi-attribute utility function $U$ for each alternative theory topic. The decision maker selects the theory topic for which the multi-attribute utility function $U$ is maximised.

In order to calculate the multi-attribute utility function $U$, we have applied the SAW (Fishburn, 1967; Hwang and Yoon, 1981). More specifically, we have used this method to evaluate different parts of the theory and rank the set of all parts of the theory. Thus the most appropriate part of the theory may be selected to be suggested to a user. In order to evaluate the degree of a tutoring need, this method takes into account the values of the attributes and their weights that were presented in Section 4.

5.1. Application of SAW

The attributes that correspond to the degree of relevance of the user’s action to each topic of the theory ($e$) and to the degree of difficulty of the topics take its values in $[0,1]$. However, all the other attributes identified by the human tutors take their values in the interval $[0,1000]$, as they take their values from a counter that is maintained in the user model. For example, a user may have made no errors or may have made 250 errors if we are counting for a very long period of time. As the attributes take their values in different intervals, they are not comparable. Therefore, the first step of the SAW method addresses the problem of scaling the values of all the attributes so that they are in the same interval that would make them comparable.

Web-IT uses the same scaling factor that was used in (Naumann, 1998). This scaling factor was employed for a similar problem in computer science. As a result, the system uses the function (1) presented in Section 2.1 to normalise the values of all the attributes. More specifically, the new values of the attribute $e$ are calculated using the function (4)

$$e_i' = \frac{x_{ei} - x_{e\min}}{x_{e\max} - x_{e\min}}$$

where $x_{ei}$ is the old value of the $e$ attribute for the $i$ alternative, $e_i'$ is the transformed value of the attribute $e$ for the $i$ alternative and $x_{e\max}, x_{e\min}$ are the maximum and minimum values of the attribute $e$ for all the alternatives generated by the system.
Similarly to the attribute \( e \), the values of the attribute \( v \) are calculated using the function

\[
v_i = \frac{x_{vi} - v_i^{\text{min}}}{v_i^{\text{max}} - v_i^{\text{min}}}\tag{5}
\]

where \( x_{vi} \) is the old value of the \( v \) attribute for the \( i \) alternative, \( v_i \) is the transformed value of the attribute \( v \) for the \( i \) alternative and \( v_i^{\text{max}}, v_i^{\text{min}} \) are the maximum and minimum values of the attribute \( v \) for all the alternatives generated by the system.

Following the above rationale, the values of the attributes \( c \) and \( d \) are estimated using the functions (6) and (7), respectively

\[
c_i = \frac{x_{ci} - c_i^{\text{min}}}{c_i^{\text{max}} - c_i^{\text{min}}}\tag{6}
\]

\[
d_i = \frac{x_{di} - d_i^{\text{min}}}{d_i^{\text{max}} - d_i^{\text{min}}}\tag{7}
\]

where \( x_{ci} \) is the old value of the \( c \) attribute for the \( i \) alternative, \( c_i \) is the new value of the attribute \( c \) for the \( i \) alternative and \( c_i^{\text{max}}, c_i^{\text{min}} \) are the maximum and minimum values of the attribute \( c \) for all the alternatives generated by the system.

For the example described in Section 3, the values of the attributes are transformed using the functions (4)–(7) so that they would all be in the interval [0,1]. Indeed, the new values of the attributes are

\[
\begin{pmatrix}
e
v
c
d
\end{pmatrix} = \begin{pmatrix}
1 & 0.9 & 0.5 \\
0 & 0 & 0 & 1 \\
0.5 & 1 & 1 & 0
\end{pmatrix}
\]

As soon as the values of the attributes have been transformed using the above scaling and all their values are in [0,1], Web-IT calculates the values of the multi-attribute utility function \( U \) for each alternative action generated by the system. More specifically, function \( U \) takes its values as a linear combination of the values of the four attributes described in Section 4

\[
U_{\text{SAW}}(X_i) = w_e e_i + w_v v_i + w_c c_i + w_d d_i
\]

where \( X_i \) is the evaluated alternative, \( w_e, w_v, w_c, w_d \) are the weights of the attributes and \( e_i, v_i, c_i, d_i \) are the values of the attributes for the \( i \) alternative. The values \( w_e, w_v, w_c, w_d \) are acquired from the stereotype that the user interacting with the system belongs to. The weights for all the stereotypes are shown in Table 1. Thus the weights of the attributes that are used for the user of the example who is 37-years-old are shown in Table 2.

Applying the weights of the attributes, as these have been identified by the empirical study described in Section 4, the function (8) can be further simplified to function (9)
Applying the values of the attributes in function 9, the multi-attribute function for the theory topic of copying files takes the value 

\[ U_{SAW}(X_1) = 0.67 \]

whereas the values of the same function were 

\[ U_{SAW}(X_2) = 0.29 \] and 
\[ U_{SAW}(X_3) = 0.56 \]

for the domain of copying folders and objects, respectively. Therefore, the system decided that the theory topic that was most appropriate for the particular user was ‘Copying Files’. Additionally, the system decided to advise the user to revise the topic of ‘Copying Folders’ and ‘Copying Objects’ as their values of the multi-attribute utility theory were not equal to null.

5.2. Discussion of the application of SAW

In order to evaluate the use of SAW in Web-IT, we conducted a formative evaluation. The evaluation experiment involved 15 novice users of various ages of a standard file manipulation program, who were asked to work with Web-IT as they would with a standard explorer during their day-to-day activities. Web-IT silently reasoned about every user’s action and in case the user had made a mistake, the system used the SAW method to generate adaptive help and tutoring based on the user model information. The assistance provided in each case by the application of the SAW method was recorded. Then, the assistance that was produced by SAW was removed from the protocols collected and they were given to 10 human tutors to comment on them. The human tutors’ comments were compared with the results of the application of SAW. In this way, we could find out whether the particular decision making method produced assistance close to the human tutors’ reasoning.

As a result, the SAW method provided identical reasoning to the majority of human tutors in 68.81% of the total cases that the system provided some kind of advice. The degree of compatibility revealed that the SAW method was quite successful at making decisions about what the user needs to learn in order to achieve his/her goal.

However, it is important to note that the SAW method relies heavily on the selection of the weights of attributes, which have to be selected very carefully. Additionally, as the empirical study revealed, the weights may vary with respect to the users’ age, as the attributes are not equally important when making decisions about users of different age. If the system were designed only for children then one set of weights would be enough. However, in the case of a system that is target to both children and adult users, the usage of

Table 2
The weights of attributes for a 37-year-old user

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>0.30</td>
</tr>
<tr>
<td>v</td>
<td>0.16</td>
</tr>
<tr>
<td>c</td>
<td>0.29</td>
</tr>
<tr>
<td>d</td>
<td>0.25</td>
</tr>
</tbody>
</table>

\[ U_{SAW}(X) = 0.30e + 0.16v + 0.29c + 0.25d \]
stereotypes for keeping different weights of the attributes can be rather effective and provide more individualised learning.

Indeed, if the system had used only one set of weights, then the system’s decision could be incorrect for some cases. For example, the weight of the attribute of \( e \) is 0.33 for children and teenagers whereas, it is only 0.28 for people older than 50-years-old. The usage of the first weight for a person of 60-years-old would probably lead to a wrong conclusion. Therefore, each stereotype of users contains default inferences about what the importance of attributes is for the users belonging to that stereotype. However, the customisation of these weights and the corresponding stereotypes should be based on information that will state clearly their relative importance. In this way, a system could reproduce, to a satisfactory extent, the reasoning of human tutors while making decisions about what a student should learn. This information can only be acquired by an empirical study with the participation of experts that will analyse their reasoning process.

6. Conclusions

In this paper, we have described Web-IT, an ILE over the Web that helps adult users learn how to operate their file store. The system constantly watches the user and in case it suspects that the user is involved in a problematic situation, it provides adaptive tutoring. Web-IT aims at providing the ‘right’ pieces of information in the ‘right’ way and at the ‘right’ time. Therefore, we have applied one MADM method, the SAW, so that we could examine its efficiency in making decisions about what a learner should learn in order to achieve his/her goals while interacting with the computer.

In order to capture the human experts’ reasoning and provide personalised tutoring we have used a combination of stereotypes with the SAW method. Stereotypes have been extensively used in the past for modelling users in ITSs and ILEs but only occasionally in combination with some kind of theory. In this paper, we propose a novel combination of age stereotypes with the SAW method and we present the main conclusions drawn by its application in an ILE. When a tutoring system is targeted to adult learners, then the age of these learners may vary considerably and so does the way they learn. Therefore, it is important for the tutoring system to be dynamically adapted to the needs of users belonging to different age groups. The combination of age stereotypes with the SAW method in the way presented in this paper can achieve this dynamic adaptation of the tutoring system to the needs of different age groups and thus it can serve more effectively the aim of adult e-training.

According to the results of the evaluation where the ILE’s responses were compared to the responses of human tutors, the combination of stereotypes with the SAW method can enable a system reproduce, to a satisfactory extent, the reasoning of human tutors while making decisions about what a student should learn. However, the customisation of weights is the most important phase so as to ensure the effectiveness of the SAW method. Therefore, an empirical study involving human experts would be needed prior to this selection.
References


