*2nd Hellenic Conf. on AI, SETN-2002, 11-12 April 2002, Thessaloniki, Greece,*
*Proceedings, Companion Volume, pp. 155-164*

# IFM: An Intelligent Graphical User Interface Offering Advice

Maria Virvou, Katerina Kabassi

Department of Informatics,
University of Piraeus,
Piraeus 18534, Greece
{mvirvou, kkabassi}@unipi.gr

**Abstract.** This paper describes a graphical user interface that provides intelligent help to users based on its user modelling component. The graphical user interface is called Intelligent File Manipulator (IFM). IFM monitors users while they work; in case a user has made a mistake, IFM intervenes automatically and offers advice. The reasoning of the user modelling component of IFM is largely based on an adaptation of a cognitive theory, called Human Plausible Reasoning (HPR). HPR is combined with techniques that refer to the individual user and stereotypes. In addition, IFM incorporates a goal recognition mechanism. The goal recognition mechanism is based on a notion called 'instability'. IFM has been evaluated in comparison with human experts acting as consultants. The results of the evaluation have shown that IFM can successfully improve human-computer interaction by producing advice helpful to users.

## 1 Introduction

A fundamental objective of human-computer interaction research is to make systems more usable, more useful, and to provide users with experiences fitting their specific background knowledge and objectives [5]. This is extremely difficult in systems that serve the needs of large and diverse user populations. An important feature to achieve better support for users in such complex systems is to provide intelligent help. Indeed, intelligent help systems may help users when they are accidentally involved in unpleasant situations. In addition, they may inform the users about useful functionality of the system that these users may not be aware of. However, in order to have effective and individualised help messages, one should infer the user's goals and characteristics.

In this paper we describe a Graphical User Interface (GUI) that provides intelligent help. The GUI is called Intelligent File Manipulator (IFM) and is meant to operate for the management of a user's file store, like the Microsoft Windows Explorer [9]. IFM monitors users' actions and reasons about them based on its user modelling component. As Fischer [5] points out, many user modelling approaches failed because they relied too much on one specific technique. In the case of IFM, user modelling is performed by combining 3 different reasoning mechanisms: a limited goal recognition mechanism, a stereotype-based mechanism and a simulator of users' reasoning based on a cognitive theory, called Human Plausible Reasoning theory. The cognitive theory is the principal reasoning mechanism and is used to generate hypotheses about possible users' beliefs concerning the use of the GUI. The limited goal recognition has been used to contribute more insight about the context of the users' actions and thus improve the system's control. Finally, stereotypes have been used to represent more domain-specific expertise concerning the use of commands. The inferential capability of stereotypes is combined with that of the domain-independent cognitive theory for the initialisation of user models.

The cognitive theory used is called Human Plausible Reasoning [4] (henceforth referred to as HPR). HPR is adapted in IFM in order to make inferences about possible users' errors based on evidence from users' interaction with the system. HPR has been used in IFM to simulate the users' reasoning, which may be correct or incorrect (but still plausible) and thus may lead to "plausible" user errors. The theory includes a variety of inference patterns that do not occur in formal-logic based theories or in the various non-classical logics such as fuzzy logic [21], intuitionist logic [7], or variable-precision logic [8].

The approach to user modelling taken in IFM is similar to that used for another intelligent help system for users of the UNIX operating system, called RESCUER [12], [13], [14]. However, the domain of a GUI, such as in IFM, is very different from the domain of a command language user interface, such as in UNIX. In addition, IFM also uses HPR to simulate the reasoning of a human expert advisor while s/he is forming hypotheses about users' actions in order to provide advice [16], [19]. In this sense, IFM has been constructed to provide a second case-study that examines the generality and the usefulness of the novel combination and adaptation of the ideas and techniques presented.

In case IFM diagnoses a problem, the system provides spontaneous assistance. IFM may also be used as a learning environment for novice users of a GUI [17] since its advice can improve novice users' skills in operating file manipulation programs.

However, tailoring explanation based on a user model is of little use unless a system has an effective means of building and updating that user model. Furthermore, tailoring based on a bad user model is probably worse than no attempts at user tailoring at all [2]. Therefore, IFM has been evaluated extensively throughout its life-cycle to ensure its effectiveness and usefulness.

## 2 Related Work

### 2.1 Intelligent Help Systems

The long history of help systems implies the need for their existence. Several approaches to intelligent help exist and all of them aim at improving the quality of help to the user. In most existing help systems, help is given after a user's request, like in UC [3], [20]. However, empirical studies [15] have revealed that there are cases where a user becomes involved in problematic situations without his/her realising it. This problem can be addressed by systems that intervene when they judge that there is a problem without the user having initiated this interaction. Examples of such systems are CHORIS [11] and Office Assistant [6]. An interesting variant on a help system is PHelpS [1], which models workers so that it can assist one worker in identifying a peer who can assist them.

IFM is quite similar to CHORIS in terms of errors. This is due to the fact that they both deal with mouse sensitive 'menu' commands that when selected, activate a prescribed action; CHORIS is an intelligent interface for manipulating emergency crisis management systems such as management systems for earthquakes. In the domain representation, CHORIS keeps all the objects relations and commands in a similar way as IFM keeps hierarchies of commands and objects. However, the two systems differ in the way their user modelling components acquire information about users. CHORIS maintains explicit user models, whereas IFM constructs implicit user models. Explicit user models are based on information that users have provided explicitly about themselves, whereas implicit models infer information, by observing and interpreting the users' behaviour [10].

Tip Wizard, an active help system that Microsoft has introduced, is very similar to the Office Assistant that is described in Lumiere Project [6]. Tip Wizard keeps implicit Bayesian user models that capture the uncertain relationships among the goals and needs of a user in order to provide to users more individualised help. Similarly to Tip Wizard, IFM operates as an active help system. However, IFM differentiates from Tip Wizard in the way that the user models are constructed and used. IFM pays more attention at users' errors and goals whereas Tip Wizard mainly infers users' current needs in order to inform them about the best way to reach their goals. IFM's objective, on the other hand, is to intervene only when this is considered really essential for helping users accomplish their plans without errors and not comment on the actual way a user may select to accomplish his/her goals. Therefore, in case IFM suspects that an action would not have the desired results for the user, it generates alternative actions that would achieve these hypothesised goals.

### 2.2 Human Plausible Reasoning

HPR is based on an analysis of people's answers to everyday questions about the world and tries to formalise plausible inferences that occur in people's responses to different questions [4]. The theory consists of a formal representation of plausible inference patterns that are frequently employed in answering everyday questions, a set of parameters that affect the certainty of people's answers to such questions and a system relating the different plausible inference patterns and the different certainty parameters. An example of a plausible inference is presented below. Let's suppose that the question asked was whether coffee is grown in Llanos region in Colombia. The answer would depend on the knowledge retrieved from memory. If the subject knew that Llanos was in a savanna region similar to that where coffee grows, this would trigger an inductive, analogical inference, and generate the answer yes.

According to the theory a large part of human knowledge is represented in "dynamic hierarchies". These hierarchies are used to model the reasoning of people with patchy knowledge. There are four kinds of relation between objects in hierarchies: generalisation (GEN), specialisation (SPEC), similarity (SIM) and dissimilarity (DIS). Statement transforms are the simplest class of inference patterns. However, the theory also introduces certainty parameters. These certainty parameters can affect the statement transforms.

The first two certainty parameters are applied to dependencies and implications and represent the conditional likelihood ($\alpha$ and $\beta$) that the right-hand (left-hand, respectively) side of the dependency or implication has a particular value given that the left-hand (right-hand, respectively) side has a particular value. SIM and DIS statements depend on the degree of similarity ($\sigma$), which represent the similarity of one set to another one. GEN and SPEC statements are affected by the degree of typicality and dominance. The degree of typicality ($\tau$) represents how typical is a subset within a set (for example, the cow is a typical mammal). Dominance ($\delta$) indicates how dominant is a subset in a set (for example, elephants are not a large percentage of mammals). Finally the only certainty parameter applicable to any expression is the degree of certainty ($\gamma$) or belief that an expression is true.

## 3 Operation of Intelligent File Manipulator

Intelligent File Manipulator (IFM) is an intelligent Graphical User Interface that works in a similar way as a standard file manipulation program, such as Windows 98/NT Explorer, as shown in Figure 1. IFM is meant to help users during their navigation and manipulation of the file store and provides advice in case this is considered necessary. In general, IFM tries to act as a human expert who watches the user over the shoulder and constantly helps him/her.

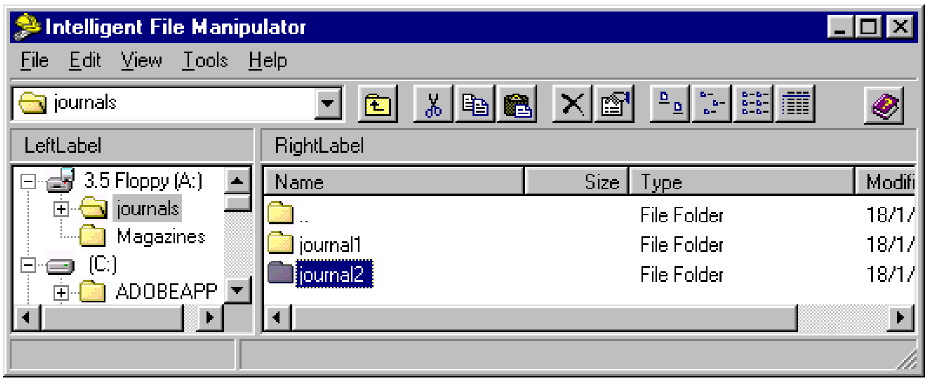


**Figure 1**: An example screen of the GUI of IFM

Every time a user issues an action, IFM 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 suggests alternative actions to the user. Otherwise, the action is executed normally. In cases when IFM decides to alert a user, the button of help is activated in order to inform the user that s/he has probably made a mistake. The user must click on that button to view the alternative actions proposed by the system.

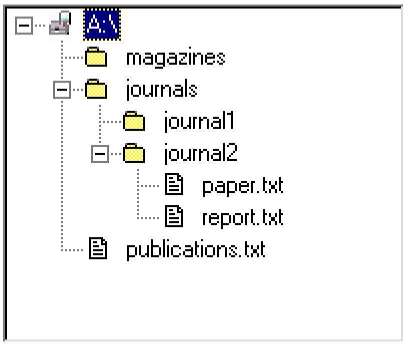


**Figure 2**: Initial file store state

An example of an interaction of a user with IFM is described below. The user's initial file store state is shown in figure 2. The user intends to delete A:\journals\journal1\. However, s/he accidentally attempts to delete A:\journals\journal2\. In this case, s/he runs the risk of losing the information stored in the two files that the folder A:\journals\journal2\ contains.

IFM would suggest the user to delete the folder A:\journals\journal1\ for two reasons:

1. the folder A:\journals\journal1\ is empty whereas the folder A:\journals\journal2\ is not and
2. A:\journals\journal1\ is very similar to A:\journals\journal2\, therefore, one could have been mistaken for the other.

## 4 Goal Recognition

In order to provide help to its users, IFM must be able to identify the user's goals and plans. Therefore, IFM uses a limited plan recognition mechanism based on a notion called "instability", which is associated with a user's file store state. The existence of instabilities implies that there are users' plans that have been started and have not been completed. Instabilities are added and/or removed from the file store as a result of users' actions. An instability is added when a plan is initiated whereas an instability is deleted when a plan is completed. The actions that both add and delete instabilities result in the continuation of a user's plan that started earlier. An empirical study [16] that was conducted in the early stages of IFM's life cycle revealed that a file store may be considered to be stable (e.g. not containing instabilities) if it does not contain:

1. empty directories
2. directories with only one child
3. multiple copies of a certain file
4. folders with the name 'New Folder'
5. files with the name 'New File'
6. one or more objects placed in clipboard by a cut or copy command and have no further use.

IFM constantly reasons about every user's action. In case an action contradicts the system's expectations it provides spontaneous advice. In order to evaluate the user's actions, IFM categorises them in one of four categories, namely 'expected', 'neutral', 'suspect', 'erroneous', depending on its degree of compatibility with the users' hypothesised intentions.

An action is categorised as expected if it is compatible with the user's hypothesised goals. Such actions are those that result in the continuation or completion of an already declared plan and, therefore, delete at least one instability. In case an action is believed to contradict the system's hypotheses about the user's goals then it is considered suspect. Actions that result in the initiation of a new plan, when there are other plans pending, are considered to be suspect. If the action is wrong with respect to the user interface formalities it is considered erroneous. In every other case, the action is considered neutral and has no influence in the list of instabilities.

The actions that are categorised as expected or neutral, are executed normally. However, if the action is categorised as suspect or erroneous then it is transformed based on the HPR theory. In order to find out the alternative action that the user intended, IFM also reasons about every alternative action generated and only those compatible with the user's hypothesised intentions are selected. However, this procedure usually results in the generation of many alternative commands and the question is how the system can identify which one of the alternative actions the user really intended.

In order to rank the alternative actions generated by the system, IFM tries to simulate human experts' reasoning by using the certainty parameters introduced in HPR. A degree of certainty is associated with each alternative action generated so that these may be sorted in a priority order. This degree represents the system's certainty that the user in fact intended the alternative action generated instead of the one issued. A detailed description of the procedure for the calculation of the degree of certainty is described in Section 5.

## 5 Advice based on User Modelling

In order to provide spontaneous advice, IFM constantly reasons about users' actions and makes hypotheses about their intentions. The generation of hypotheses is based on HPR, which is used as a tool to provide a simulation of possible users' errors. HPR detects the similarity/dissimilarity (SIM/DIS) relationship between a question and the knowledge retrieved from memory and drives the line (type) of inference. Every time the system suspects that a user might have been mistaken, it generates advice with respect to his/her hypothesized intentions. However, in order to select the most appropriate advice for the particular user, the system should know about the user's usual errors and characteristics. This information is included in a detailed user model, which is constantly consulted by the system.

The certainty parameters of HPR are used by the user modelling component of IFM to capture long-term information about the user. The adaptation of the certainty parameters of HPR used in IFM is presented below:

- Degree of typicality ($\tau$) of an action in the set of all actions issued by the user represents the estimated frequency of execution of the command by the particular user.
- Degree of similarity ($\sigma$), which is used to calculate the resemblance of two commands or two objects.
- Frequency ($\varphi$) of an error in the set of all actions represents how often a specific error is made by a particular user or the users of a particular stereotype.
- A user's weaknesses can be recognised by the dominance ($\delta$) of an error in the set of all errors.
- Degree of certainty ($\gamma$) represents the system's certainty that the user intended the alternative command generated. The degree of certainty is calculated by taking into account all the above certainty parameters and its calculation is going to be described in the Section 6.

Every time the user issues a command, IFM reasons about it with respect to its expectations about the user's goals. If an action contradicts the system's expectations, the system searches for alternative commands, which should be compatible with those expectations. However, there may be a variety of explanations for observed incorrect users' actions. Therefore, there is a need to attach priorities to different explanations. IFM uses the information stored in its user modelling component in order to select the most appropriate advice.

The system combines both stereotypes and individual user modelling in order to achieve more individualised help. The stereotypes are used in IFM only for capturing the initial impression about a user. A stereotype is activated after implicitly acquiring information by observing the user while s/he interacts with the system. The information provided by the stereotype is given in the form of values of the certainty parameters of HPR.

During the initial interactions of a user with the system, information about the user is provided by the stereotypes. However, the system is constantly collecting information about the particular user's behaviour and informs the user model. As the system collects more and more evidence about a user, the percentage of information provided by the stereotype diminishes while the percentage of acquisition by the individual user model increases.

When the system has received the necessary information from the user model, it uses the certainty parameters of HPR to determine the priority among actions belonging to the same category. More specifically, after the generation of alternatives, the system calculates a degree of certainty for each alternative command. The degree of certainty represents how confident the system is that the user really intended the alternative command generated and is calculated as a combination of all the certainty parameters.

Each parameter is multiplied by a weight, which is determined with respect to how important the particular certainty parameter is in the reasoning process of human experts. An evaluation of the advice generator of IFM [18] revealed how important each criterion was in the reasoning process of the human experts.

The most important criterion of a human expert when evaluating an alternative action, which was going to be proposed to the user, was the similarity of that action to the one issued by the user, because users usually tend to tangle up actions or objects that are very similar. The second most important criterion that human experts used was whether a particular user's error was the most frequent error of all errors that this user made.

A very important criterion when evaluating an alternative action was the frequency the user makes such an error while interacting with the system, even if this error is not his/her weakness. When proposing an alternative action to a user, the system must know if a user uses that particular command quite often or not. Although it is unlikely that the user had made a mistake in the execution of a command that s/he uses quite often, still there is a possibility that the user may have made a carelessness mistake.

In view of the above the current form of the formula for the calculation of the degree of certainty should be the following, so that the reasoning of IFM would be close to human experts' reasoning:

$$\gamma = 0.4 * \sigma + 0.3 * \delta + 0.2 * \varphi + 0.1 * \tau \qquad (1)$$

# 6 Usability Evaluation of IFM

The usability evaluation of IFM aimed at revealing how successful IFM was at generating advice and whether its function did indeed help the users. Therefore, a competitive usability test of IFM took place and IFM was evaluated in comparison with a standard explorer and with human experts. In the experiment of the evaluation, some protocols of users' interactions with a standard explorer were collected and were given to 10 human experts to comment on them. These protocols were also given as input to IFM so that IFM's reactions were compared to the human experts' comments.

The results were encouraging concerning the comparison of IFM's responses to human experts' responses. In cases when there was a total agreement of human experts' opinions, IFM produced either a very similar or exactly the same advice to that of the human experts. This usually corresponded to cases where the error was "obvious" to human advisors such as the error presented in the example, where there was 90% of agreement among the experts. However, there were cases where there was a diversity of human experts' opinions. In those cases IFM's advice was either identical to the advice of the majority of human experts or in fewer cases it was compatible to the advice provided by a minority of experts. In general, the degree of compatibility between the majority of human experts and IFM's advice was 63%, which was quite satisfactory.

## 7 Conclusions

In this paper we described an intelligent graphical user interface for a program that manipulates files, called IFM. The main focus of this paper has been on how user models can be constructed using a combination of methods. In particular, stereotypes of users are combined with the principles of Human Plausible Reasoning theory and a limited goal recognition mechanism in order to have a better comprehension of users' actions and enrich the system with human-like abilities. The inferences made by the stereotypes concern some certainty parameters that are defined by the theory and are adapted to be used in IFM. Depending on the values of these certainty parameters, IFM selects the most appropriate alternative command to be suggested to a user in case s/he has issued a problematic one.

The system has been evaluated in terms of some usability issues. The results of the evaluation experiment revealed that IFM provided a user-friendly environment for file manipulation and proved to be quite successful in providing individualised help to its users. This individualised help alleviated the user during his/her interaction with the system, as s/he did not suffer so many consequences of errors as s/he would in case of a standard explorer.

## References


1. Bull S., Greer J., McCall G., Kettel L. and Bowes J.: 'User Modelling in I-Help: What, Why, When and How'. In Bauer M., Gmytrasiewicz P.J. and Vassileva J. (eds.): User Modeling: Proceedings of the Sixth International Conference, (2001) 117-126.
2. Cawsey A.: 'User Modelling in Interactive Explanations'. In User Modeling and User-Adapted Interaction, Vol. 3, (1993) 221-247.
3. Chin D. N.: 'KNOME: Modeling What the User Knows in UC'. In Kobsa A. and Wahlster W. (eds.), User Models in Dialog Systems, (1989) 74-107.
4. Collins, A., Michalski R.: 'The Logic of Plausible Reasoning: A core Theory'. In Cognitive Science 13, (1989) 1-49.
5. Fischer G.: 'User Modeling in Human-Computer Interaction'. In User Modeling and User-Adapted Interaction, Vol. 11, (2001) 65-86.

6. Horvitz E., Breese J., Heckerman D., Hovel D., and Rommelse K.: 'The Lumiere Project: Bayesian User Modeling for Inferring the Goals and Needs of Software Users'. In Proceedings of the fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, Morgan Kaufmann: San Francisco, (1998) 256-265.
7. Martin-Lof: 'Constructive mathematics and computer programming'. In Methodology and Philosophy of Science VI, Amsterdam: North Holland Publishing Company, (1982).
8. Michalski R.S., Winston P.H.: 'Variable precision logic'. In Artificial Intelligence, Vol. 29, (1986) 121-146.
9. Microsoft Corporation: 'Microsoft® Windows® 98 Resource Kit', Microsoft Press (1998).
10. Rich E.: 'Stereotypes and User Modeling'. In Kobsa A. and Wahlster W. (eds.): User Models in Dialog Systems, (1989) 199-214.
11. Tyler S. W., Schlossberg J. L., Gargan Jr. R. A., Cook L. K. and Sullivan J. W.: 'An Intelligent Interface Architecture For Adaptive Interaction'. In Sullivan Joseph W., Tyler Sherman W. (eds.), Intelligent User Interface, ACM Press, New York. Addison-Wesley Publishing Company, (1991) 85-109.
12. Virvou, M.: 'RESCUER: Intelligent Help for Plausible User Errors'. In Proceedings of ED-MEDIA/ED-TELECOM 98, World Conferences on Educational Multimedia and Educational Telecommunications, Vol. 2, (1998) 1413-1420.
13. Virvou, M.: 'Automatic reasoning and help about human errors in using an operating system'. In Interacting with Computers, Vol. 11, No. 5, (1999) 545-573.
14. Virvou M., Du Boulay, B.: 'Human Plausible Reasoning for Intelligent Help'. In User Modeling and User-Adapted Interaction, Vol. 9, (1999) 321-375.
15. Virvou M, Jones J. and Millington M.: 'Virtues and Problems of an Active Help System for UNIX'. In Hegner St. J., Mc Kevitt P., Norvig P. and Wilensky R. (eds.) Artificial Intelligence Review, Intelligent Help Systems For Unix, Vol. 14, No. 1/2, (2000) 23-42.
16. Virvou M., Kabassi K.: 'An Empirical Study Concerning Graphical User Interfaces that Manipulate Files'. In Bourdeau J. & Heller R. (eds.): Proceedings of ED-MEDIA 2000, World Conference on Educational Multimedia, Hypermedia & Telecommunications, AACE, Charlottesville VA, (2000a) 1117-1122.
17. Virvou M., Kabassi K.: 'An Intelligent Learning Environment for Novice Users of a GUI'. In Gauthier, G., Frasson, C. & VanLehn, K. (eds.): Lecture Notes in Computer Science (Intelligent Tutoring Systems, Proceedings of the 5th International Conference on Intelligent Tutoring Systems, ITS 2000), Vol. 1839, Springer, Berlin, (2000b) 484-493.
18. Virvou M., Kabassi K.: 'Evaluation of the advice generator of an intelligent learning environment'. In Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT 2001), IEEE Computer Society, (2001a) 339-342.
19. Virvou M., Kabassi K.: 'Reasoning about users' actions in a Graphical User Interface'. To appear in Human-Computer Interaction, (2002).
20. Wilensky R., Chin D.N., Luria M., Martin J., Mayfield J., and Wu D.: 'The Berkeley UNIX Consultant Project'. In Hegner St. J., Mc Kevitt P., Norvig P. and Wilensky R. (eds.) Artificial Intelligence Review, Intelligent Help Systems For Unix, Vol. 14, No. 1/2, (2000) 43-88.
21. Zadeh L.A.: 'Fuzzy sets'. In Information and Control, Vol. 8, (1965) 338-353.