A FLEXIBLE TOOL FOR ASSUMPTION-BASED USER MODELING

Giorgio Brajnik, Carlo Tasso, Antonio Vaccher
Laboratorio di Intelligenza Artificiale
Dipartimento di Matematica e Informatica
Università di Udine

ABSTRACT. The aim of this paper is to present a new flexible general-purpose shell, called UMT2 (User Modeling Tool 2), which supports the design and development process of user modeling applications and which features an original strategy for performing the modeling activity in a nonmonotonic way. More specifically, UMT2 utilizes a modeling approach called assumption-based user modeling which exploits an ATMS-like mechanism for maintaining the consistency of the user model. The modeling task is thus divided into two separate activities, one devoted to user classification and user model management, and the other devoted to consistency maintenance of the models. Modeling knowledge is represented by means of stereotypes and production rules. The ATMS mechanism is capable of identifying, at any given moment during an interaction, all the possible alternative models which are internally consistent. The choice of the most plausible one among them is then performed according to a procedure exploiting an explicit preference criterion. UMT2 is also characterized by a very well defined and easy-to-use interface with the rest of the application, and by a specialized development interface which supports the knowledge engineer during the construction of specific applications. UMT2 has been developed in CLOS Common LISP.

1. INTRODUCTION

User modeling has been recognized as a very important and useful feature in several man-machine systems where the interaction with the user plays a crucial role (Kobsa and Wahlster, 1989). A user modeling component allows flexible and profitable adaptation of the system behavior to the specific characteristics and nuances of an individual user, improving in such a way the usability of the system, the economy of the interaction, and the capabilities of the system of understanding and processing user requests beyond the pure content of his/her utterances or commands. The application of user modeling techniques have been reported in many fields, such as dialog systems (Wahlster and Kobsa, 1989), and intelligent tutoring systems (Wenger, 1987). An area where user modeling has been considered a fundamental requirement is that of information retrieval systems (Belkin et al., 1987; Daniels, 1986; Brajnik, Guida and Tasso, 1987b).

Nevertheless the design and development of user modeling systems constitutes a very ad-hoc and demanding process and for this reason the idea of exploiting generic shells specifically devoted to user modeling has emerged, and a few systems have been developed along this line, such as GUMS (Finin, 1989), UM-tool (Brajnik, Guida and Tasso, 1990), and BGP-MS (Kobsa, 1990), which can be integrated within a specific application in order to provide user modeling services. Even if no established and general theories underlying the user modeling task exist, all the above mentioned tools have been designed following some common idea. More specifically, all of them are based on an explicit information structure, called user model, for storing the individual representation of the current user of the system, and all of them exploit the idea of stereotypes (Rich, 1983) for representing different classes of users.

However, none of them includes satisfactory mechanisms for approaching one of the most typical features of the user modeling task, i.e. its non-monotonicity. In fact user models usually contain two main kinds of information: known, observed facts about the user and assumptions about the user. Information belonging to this last category may be possibly subjected to revisions, if evidence against the specific assumption is collected. The need of explicitly including such mechanisms within user modeling systems has been recognized by several researches (Finin, 1989; Rich, 1989; Kobsa, 1990; Brajnik, Guida and Tasso, 1990).

The aim of this paper is to present a new flexible general-purpose shell, called *UMT2* (*User Modeling Tool* 2), which supports the design and development process of user modeling applications and which features an original strategy for performing the modeling activity in a nonmonotonic way. More specifically, UMT2 utilizes an ATMS-like mechanism (DeKleer, 1986) for maintaining the consistency of the user model, and for this reason the modeling strategy has been called *assumption-based user modeling*. The modeling task is thus divided into two separate activities, one devoted to user classification and user model management, and the other devoted to consistency maintenance of the modeling process. Modeling knowledge is represented by means of stereotypes and production rules. The ATMS mechanism is capable of identifying, at any given moment during an interaction, all the possible alternative models which are internally consistent. The choice of the most plausible one among them is then performed according to a procedure using an explicit preference criterion. UMT2 is also characterized by a very well defined and easy-to-use interface with (the rest of) the application, and by a specialized development interface which supports the knowledge engineer during the construction of specific applications.

UMT2 has been developed and experimented within IRNLI, an expert interface between casual users and information retrieval systems (Brajnik, Guida and Tasso, 1987a). A previous version of IRNLI already included a generic user modeling shell (the UM-tool system already mentioned above), which however utilized only ad-hoc procedures for dealing with possible inconsistencies arising during the modeling process.

The paper is organized in the following way. Section 2 discusses the main concepts underlying the assumption-based user modeling approach. Section 3 illustrates the UMT2 shell, its architecture and mode of operation. Section 4 concludes the paper.

2. ASSUMPTION-BASED USER MODELING

The main original contribution of UMT2 concerns a new approach to the modeling activity which is based on the use of an assumption-based mechanism for building and maintaining user models. Some of the general features of UMT2 have been derived from UM-tool (Brajnik, Guida and Tasso, 1990), and are briefly summarized at the beginning of this section. In the rest of the section we will focus on assumption-based user modeling.

In our approach, the modeling activity is devoted to modeling single individual users, by means of an explicit inferential process, capable of repeatedly classifying the user during a single session.

The main knowledge sources utilized in our approach are taxonomies of the potential user population and empirical associations of features typical of certain classes of users. Classes (possibly overlapping) of users are described by a finite set of (possibly multivalued) attributes, they are characterized by necessary and sufficient conditions, and a set of rules constrain the possible user descriptions. Domain specific knowledge is used in order to derive the values of some user properties on the basis of values of other properties.

A number of basic hypotheses is made concerning the assumption-based user modeling process. We assume that *(i)* the information about the user at any moment is inherently incomplete; *(ii)* that user modeling is an incremental process, based on iterative extension, refinement and revision activities, which can take into account any new piece of user information as soon as it becomes available; and *(iii)* that the information contained in the model can never be considered as definitive, since new information may be acquired which is not compatible with the old one.

It appears very clearly from the above hypotheses that the modeling process features a *nonmonotonic* nature, which motivates the utilization of some kind of belief revision system. As already mentioned, in UMT2 we have adopted an assumption-based TMS mechanism.

3. USER MODELING TOOL 2

The architecture of UMT2 is constituted by two modules:
- the MODEL MANAGER, devoted to carry on the model management activity. More specifically the Model Manager carries out the following tasks: (i) handling the interaction of UMT2 with the knowledge engineer and the main application program; (ii) performing user classification; (iii) modifying the user models after a user reclassification or when new data about the user are available; (iv) choosing the Current User Model among the Alternative User Models provided by the other module.
- the CONSISTENCY MANAGER, devoted to assuring that the current user model and the alternative user models are all in a consistent state. The Consistency Manager is based on an ATMS.

The operation of UMT2 is supported by the following knowledge bases:
- the STEREOTYPES knowledge base, containing all the descriptions, called stereotypes, of user classes. Stereotypes are organized in a multiple inheritance hierarchy through an IS-A relation defined by the inclusion relation holding between corresponding classes. The root of the hierarchy is the *Generic User Stereotype*, which is a very general description of a potential user and is used to define the set of user attributes. All other stereotypes are called *Class Stereotypes*.

  Each stereotype is divided in two parts: a *trigger*, which is a predicate expressing necessary and sufficient membership conditions of a specific user with respect to the class denoted by the stereotype, and the *defaults*, a list of multivalued attributes along with their values representing typical traits of the users belonging to the class. A stereotype is said to be *active* if and only if its trigger is satisfied or the stereotype is the parent (through the IS-A relation) of an active stereotype.
- the MODELING RULES base, containing *(i) constraints* on the admissible values of a single attribute or combinations of attributes, and *(ii) inference rules* allowing the derivation of some attribute values given certain values of other attributes. When a constraint is not satisfied, an inconsistency is raised.
- the DERIVATION SOURCES base, containing a list of the *derivation sources*, which intuitively represent the possible sources from which an assumption has been derived. Each derivation source is associated to a priority weight which represents a preference relation on derivation sources and it is used for choosing the most plausible model among those made available by the CONSISTENCY MANAGER module.

UMT2, during its operation, exploits two working memories:
- the CURRENT USER MODEL: a representation of the current user as it is perceived, up to that moment, by UMT2. It is constituted by a collection of attributes/values pairs being either:
  * individual facts (e.g. the user's age) supplied by the main application program;
  * default information inherited from the Generic User Stereotype;
  * default information inherited from the active class stereotypes;
  * the result of internal derivation performed by applying some modeling rule.
- the set of ALTERNATIVE USER MODELS: a set of alternative models of the current user generated by the assumption-based modeling process. They represent alternative, plausible, consistent and less preferred representations of the current user.

Finally, UMT2 includes also the USER MODELS database, storing the individual models of all the users that used in the past the application. These models are appropriately retrieved at the beginning of each interaction, and stored back at the end.

3.3 THE MODELING PROCESS

The main activity performed during the assumption-based modeling process is the modification of the user model, performed throughout the session in a demand-driven fashion and aimed at integrating in the user models new data about the user provided by the main application program and at managing inconsistencies. More specifically:

1. when new data about the user are asserted, all the applicable Modeling Rules are fired; maximality and consistency of each user model are automatically handled by the Consistency Manager module;
2. when inconsistencies are signalled by the main application program to UMT2, these are notified to the Consistency Manager module that manages it through standard ATMS procedures;
3. when active stereotypes are identified, their instantiation causes the assumption of their default values and the firing of applicable Modeling Rules. Analogously for non-active stereotypes and their deistanciation.

Whenever an inconsistency among two or more assertions is discovered (either internally to UMT2 during model modification or signalled by the main application program) the ATMS algorithms resolve it by identifying the underlying assumptions and generating new user models which are maximal and internally consistent.

Since the Consistency Manager provides several possible model of the user at the time and, on the other hand, the Model Manager has to consider only one user model, a selection has to take place, which identifies the Current User Model out of the available user models so far computed, called Alternative User Models. The selection is performed by the Model Manager whenever a reclassification takes place or an inconsistency has been dealt with by the Consistency Manager. The decision is based upon an order relation induced on the Alternative User Models by the priority weights associated to each derivation sources. Informally defined, the *most plausible* model is the one which features the highest number of occurrences of the most preferable derivation source, or, when such number is the same for two or more models, the highest number of occurrences of the second preferable derivation source, and so on.

4. CONCLUSIONS

In this paper we have illustrated the organization of UMT2, a flexible tool for building user modeling subsystems, which is centered on an assumption-based user modeling process. The main advantage of such a modeling process derive from the very concise, simple and abstract interface of the module devoted to maintaining the consistency and from the possibility to choose the most plausible model according to a general, explicit and customizable criterion. These features reflect on the easiness with which a knowledge engineer can design and construct the user modeling knowledge (stereotypes and rules); in particular, the knowledge engineer is not asked to deal with explicit mechanisms for handling inconsistencies, ubiquitous in user modeling processes.

Two main limitations of UMT2 are: (i) the combinatorial explosion of the number of alternative models, and (ii) the monotonic growth of the graph internally handled by the Consistency Manager and representing all believed assertions and all justifications among them. We plan to approach both problems by adopting a heuristic criterion for pruning those models that are likely to never become very plausible. Other planned extensions of UMT2 concern the definition of a powerful and declarative language to specify the inheritance network of stereotypes, and the definition of a declarative language to specify choice criteria for the Current User Model among the available ones.

UMT2 is implemented in Common Lisp with the CLOS extension. It was initially developed on a Symbolics Mac-Ivory machine, and later ported to other Common Lisps. The testbed application utilized for experimentation concerns a touristic advice giving system.

Future research openings concern the development and experimentation of a user modeling subsystem based on UMT2 included in the FIRE prototype, a flexible environment for developing interfaces to information retrieval systems. After a laboratory validation of such a user modeling subsystem, an experimentation of FIRE with real users will be carried out in order to evaluate the benefits offered by the inclusion of a user modeling component within an expert interface for on-line bibliographic databases.

REFERENCES

Belkin, N et al. 1987. Distributed expert based information systems: an interdisciplinary approach. *Information Processing & Management* vol. 23(5).

Brajnik G., Guida G., and Tasso, C. 1987a. Design and experimentation of IR-NLI: an intelligent user interface to bibliographic data bases. in L. Kerschberg (Ed.), *Expert Data Base Systems*. Menlo Park, CA: The Benjamin/Cummings Publishing Company, pp. 151-162.

Brajnik G., Guida G., and Tasso, C. 1987b. User modeling in intelligent information retrieval, *Information Processing & Management*, vol. 23(4), pp. 305-320.

Brajnik, G. , Guida, G. and Tasso, C. 1990. User Modeling in Expert Man-Machine Interfaces: A Case Study in Intelligent Information Retrieval. *IEEE Transactions on Systems, Man, and Cybernetics* Vol. 20(1), 166-185.

Daniels, P.J. 1986. The user modelling function of an intelligent interface for document retrieval systems, in *Proc. IRFIS 6: Intelligent Information Systems for the Information Society*, Amsterdam, NL: North-Holland, pp. 162-176.

De Kleer, J. 1986. An Assumption-based Truth Maintenance System. *Artificial Intelligence* 28, 127-162.

Finin, T. W. 1989. GUMS - A General User Modeling Shell. In: *User Models in Dialog Systems*. A. Kobsa, W. Wahlster (Eds.), Springer Verlag, pp. 411-430.

Kobsa, A and Wahlster, W. (Editors) 1989. *User Modeling in Dialog Systems*. Springer Verlag, Berlin, Germany.

Kobsa, A. 1990. Modeling the user's conceptual knowledge in BGP-MS, a user modeling shell system. *Computational Intelligence* 6, 193-208.

Rich, E. 1983. Users are individuals: individualizing user models. *Int. Journal of man-Machine Studies* vol. 18, 199-214.

Rich, E. 1989. Stereotypes and User Modeling. In: *User Models in Dialog Systems*. A. Kobsa, W. Wahlster (Eds.), Springer Verlag, pp. 35-51.

Wahlster, W. and Kobsa, A. 1989. User Models in Dialog Systems. In: *User Models in Dialog Systems*. A. Kobsa, W. Wahlster (Eds.), Springer Verlag, pp. 4-34.

Wenger, E. 1987. *Artificial Intelligence and Tutoring Systems*. Morgan Kaufman, Los Altos, CA.