A User Profiling Component Developed with the Aid of User Ontologies

István-Tibor Nébel, Dipl-Inf [1], Ralf Paschke, MD[1] , Barry Smith, PhD[2,3]

[1]Medical Learning and Information Systems, 3rd Department of Medicine
[2]Institute for Formal Ontology and Medical Information Science,
both: Faculty of Medicine, University of Leipzig, Leipzig, Germany
[3]Department of Philosophy, University at Buffalo, NY, USA

ABSTRACT

What follows is a contribution to the field of user modeling for adaptive teaching and learning programs especially in the medical field. The paper outlines existing approaches to the problem of extracting user information in a form that can be exploited by adaptive software. We focus especially on the so-called stereotyping method, which allocates users into classes adaptively, reflecting characteristics such as physical data, social background and computer experience. The user classifications of the stereotyping method are however ad hoc and unprincipled, and they can be exploited by the adaptive system only after a large number of trials by various kinds of users. We argue that the remedy is to create a database of user ontologies from which ready-made taxonomies can be derived in such a way as to enable associated software to support a variety of types of users.

INTRODUCTION

Our topic is the construction of interactive software systems which are able to recognize and to adjust themselves to the needs of particular users at every stage of use, whether there users be beginners or experts.[1]

Systems of this sort are called adaptive systems, and we have developed a number of computer-based interactive programs to facilitate individual learning by patients of diabetes mellitus[2] and celiac diseases.[3]

The program 'Hypoglycaemia', designed to facilitate patients' learning from hypoglycaemias of diabetes mellitus, is able to adjust itself to each user's current standard of knowledge. Its practicability has been evaluated on the basis of the learning success of 120 diabetes patients,[4] and the patients in our trial not only learned faster with an adaptive rather than a conventional instructional program but also obtained significantly better results. From this it can be concluded that individually adaptive instructional programs are better suited for learning than conventional interactive instructional programs. Such programs not only avoid conveying redundant information to the user but they are also able to maximize the quality of user support, in terms of both content and mode of presentation.

GENERATING USER-PROFILES

Each individual observes the world in his own fashion. The determination of individual requirements are therefore important prerequisites for an efficient application of a tutorial system.[5] Further, each individual user brings diverse needs and expectations to his interaction with a software system.

A user classification serves as a basis for an adaptive system; it saves and analyzes the data pertaining to each particular user and makes available information relevant to the program's adaptation to the user in each successive stage.[1,6,7] The data are stored for the most part in the form of attribute-value pairs, which represent the current state of knowledge of the user as well as his personal characteristics, features, preferences, and so forth.

Some users are already familiar with the system and with the domain of knowledge and so need little support. Others are familiar with neither. We thus have two-initial dimensions variability in our user classification reflecting level of technical competence and need for content-related assistance.

In addition to the general user classification we need a?? the means to construct user profiles, which is to say? Representations of individual users of their attributes and competences, and of the stage they have reached in their interaction with the program.

TYPES OF ADAPTIVE ALGORITHMS

The needed adaptive capability can be achieved through a variety of methods and algorithms which analyze user data and use this data as a basis for making inferences about appropriate system-behavior in the future. The range of methods, which is illustrated in Table 1, can be classified broadly into stereotype-based, rule-based,[8] and mathematical and statistical approaches. One way of classifying these alternatives is according to their different forms of representation (logical, statistical, and so on), as illustrated in Table 2.

Primary Acquisition Heuristics (EH)	Overlay Model (OV)
Stereotypes (ST)	Filter Techniques (FT)
Double Stereotyping (DT)	Candidate-Critic-Model (CM)
Preferences as Goals (PG)	Ratings Criteria as Goals (RC)
Genetic Algorithms (GA)	Evolving Agents (EA)
Rule-based Expert System Techniques (EX)	Fuzzy Logic (FL)
Bayesian Nets (BN)	Dempster-Shafer Theory (DS)
Neural Networks (NN)	Spreading-Activation Approach (SA)

Table 1. The most frequently used methods for generating user classifications

Table 1 makes clear that there exists a variety of procedures for establishing user classifications and user profiles. Some of these include the application of statistical or probabilistic methods in order to generate assumptions about users under conditions of uncertainty.

A classification of the different methods is illustrated in Table 3. The table tells us for example that the method of stereotyping is a statistical, heuristic and symbolic method, and that it lends itself to use with ontologies.

These methods are further classified according to whether they employ symbolic paradigm or the subsymbolic paradigm associated with connectionistic methods.

ad hoc	Rule-based		Mathematical			
	logical	symbolic	Heuristic	logic-based	statistical	Probabilistic
EH	OV	EX	RC	ST	FL	BN
PG	FT		EA	DT	NN	DS
			GA	CM		

Table 2. Classification by form of representation

	EH	OV	ST	CM	PG	RC	GA	EA	EX	FL	BN	DS	NN	SA
Connectionistic													●	●
Statistically			●				●						●	
Rule based								●						
Heuristic	●		●					●						
Probabilistic											●	●		
Symbolic	●		●					●			●	●		
Subsymbolic										●			●	
Individual							●			●	●	●	●	
Stereotypical			●					●						
Usable with ontologies			●				●	●		●	●	●	●	●

Table 3. Semantic matrix of adaptive methods

While, in the symbolic paradigm, the objects of the real world are displayed through formal rules with the help of symbols and via reference to certain sorts of dependencies between objects, the subsymbolic paradigm uses Connectionistic models in order to allow problems to be solved through 'blind' weighted associations between nodes.[11] The final row in Table 3 indicates which alternatives can be used in conjunction with ontologies along the lines described below.

THE METHOD OF STEREOTYPES

Here we are interested primarily in those approaches which rest on the symbolic paradigm. We are interested especially in the most common and hitherto most successful such approach, which is that of conceptual clustering, illustrated by Lebowitz's UNIMEM algorithm.[10] This rests on the method of *stereotyping*, by which classes of users – constructed for example according to their physical characteristics, social background and computer experience – are represented within what is called a stereotype hierarchy. Adaptive methods are then employed in the initial stages of use of the system to allocate users to specific classes in such a hierarchy, in such a way that previously unknown characteristics of users can be inferred on the basis of the assumption that they will share characteristics with other users in the same class.

Such clustering methods allow new stereotypes to be created on the fly, and thus to be included within a continuously evolving stereotype hierarchy. In the UNIMEM algorithm, information about the real world is learned via generalization from examples which are ranked for similarity in such a way that ever more detailed stereotypes can be formed.

One disadvantage of this method, however, is that not every characteristic can or ought to be taken into account. Some characteristics have no significance in relation to others, so that if they are incorporated into a hierarchy then unnecessary specializations will result. Hence a classification of characteristics is needed that is independent of the immediate products of the stereotyping process. Another disadvantage is that, in building the stereotype hierarchy, an adaptive system can come up with the needed derivations only after a number of uses by different kinds of users. The stereotype hierarchy thus fails to exploit in a systematic way the fact that there are user characteristics which re-occur in every class of users.

TOWARDS USER ONTOLOGIES

In light of the problems mentioned above we propose a new method for the creation of user-profiles through the construction of a database of user ontologies from which ready-made taxonomies of different types of users can be extracted *en bloc*. Such ontologies will constitute a shared resource that is available to all those engaged in the construction of adaptive software. At the same time the database should be constructed in such a way that the taxonomies of user-characteristics and user-types can be updated in light of the actual experience of users and system developers. The approach in terms of ontologies has the additional advantage that the principles used in the building of an ontology can be stated explicitly and evaluated and corrected on the basis of the successes and failures of ontology-building in different areas. User ontologies can then be used either as a method for creating user-profiles in its own right or as a supplement to the stereotyping method. Moreover they can be used either as a method of jump-starting the process of hierarchy construction or as a control on the quality of the results of such a process.

The term 'ontology' refers in software circles to a family of methods for structuring information via the establishment of standardized taxonomies and associated definitions and theories. On many common readings it refers to a logical theory which gives an explicit account of a conceptualization,[11] often by utilizing the machinery of one or other description logic (DL).[12] Much contemporary work in ontology is being carried out under the auspices of the Semantic Web project, where DL-based ontological applications are called upon to support the integration of highly diverse information resources by providing a system for annotating web documents in terms of standardized terminology hierarchies.

The term 'ontology' is of course also used in philosophical circles, where it refers to the study of 'the nature and organization of reality'.[13] Ontology in the philosophical sense attempts to discover theories that match the domain of reality under examination.[14] It, too, focuses primarily on the preparation of taxonomies of the types of entities existing in given domains (including the types of relations which unify these entities together into complex wholes of different sorts).

Ontology in the information systems sense normally begins with conceptualizations developed by human beings for particular practical purposes, and seeks to formalize such conceptualizations in ways that make them implementable in computer applications. Philosophical ontology, in contrast, seeks theories of reality prepared not on the basis of simplified models but rather with the goal of maximal descriptive adequacy to the world beyond. Here we shall seek to marry the two approaches, developing a realistic, detailed user ontology and exploring ways in which this ontology can be exploited by software systems.

A Multi-Categorial User Ontology: A user ontology in our sense will consist in a classification of users and of features of users, whereby each categorized class will be linked with associated information.

An adaptive software system needs machinery both to classify different types of users and to keep track of the ways in which user's characteristics change as a result of their experience in using the system. Hence our ontology needs to keep track not only of user-parameters in the narrow sense but also of parameters relating to the *processes* in which users are involved, especially processes of system use. In the spirit of philosophical uses of the term 'ontology' we will develop a highly general user ontology distinguishing the following dimensions of classification (which correspond to the top-level categories of the ontology BFO – for 'Basic Formal Ontology' – currently under development in Leipzig):[15]

1. types of users
2. characteristics of users
   a. permanent (independent of experience with the software system)
   b. variable
      i. change independently of use of system (for example: age, disease state)
      ii. change with experience of use of system
3. types of user behavior
   a. behavior independent of the system (including future behavior influenced by the system)
   b. behavior involving the system
      i. types of system use (keyboard actions, etc.; legal/illegal, etc.)
      ii. other behavior involving the system (rejection, etc.)
4. contexts/environments of users
   a. contexts independent of the system
   b. contexts of system use

We envision a general database of ontologies, each one addressing, all of these dimensions, to which the authors of adaptive software from the medical domain – and also from other domains – could contribute additional components as well as evaluation and criticism. The database of ontologies can thereby serve as a forum within which those working on adaptive teaching and learning software can interact and profit from results already gained.

The existence of such a unified database of user ontologies will also make it possible to avoid the costly and elaborate construction of user ontologies through the adaptive systems themselves. It will mean that we can categorize users in more specialized ways and at a very early stage in the use of the program. The initiation process for adaptive systems is thereby greatly simplified.

Such a database of user ontologies will help also in the development of adaptive methods which can be easily transferred from one domain to another. Thus it should be possible to coordinate work on adaptive methods by exploiting the fact that different groups employ the same implemented ontology system. Routines for handling different combinations of parameters, such as weighting of user properties, treatment of unknown properties and the like, can be shared across domains.

Some Examples: The method of user ontologies is designed to create a framework for maximal adaptivity. The users of a medical expert system such as Eliot's CARDIAC tutor[16] can be subdivided into nurses, assistants, doctors, etc. The content conveyed by the system can then in each case be coordinated to the skill-level and needs of the corresponding user group.

Some form of coarse classification of users can of course be effected by users themselves via direct input at the beginning of their interaction with the program. But even then an array of possible alternatives needs to be created in advance via something like an ontology of the type here envisaged. More detailed profiling of each specific user, for example according to level of knowledge, can only be established via comparisons, effected through the use of question and answer methods, with the corresponding characteristics stored in the user ontology.

In the domain of nutrition the ontology can establish classifications of eating habits and preferences for specific sorts of foods in terms of which each user can be assigned to a specific user type. The adaptation process can then give special indications in order to ensure the avoidance of specific sorts of erroneous diet on the part of patients of specific types. Clearly even in the single application domain of medicine, the scope of relevant user ontologies will be very broad.

SOFTWARE APPLICATION

The Adaptation Process: The principal procedure of an adaptation process is that of user-profiling. The working process follows the universally principle of *observing the user--reasoning--storing--intervene to the user*. On the proposal here advanced, this will include an ontology data pool as one sub-component.

In the applications developed in Leipzig, the user profiling component includes three modules, which have different tasks in the process of adaption. First, is the *Setter module*. This monitors the users' interaction with the system, the time needed for specific tasks, entries made, and so on. The users' interaction yields the Input-Value of the *strategy definition module*. Second, is the *decision module* which includes also the user ontology. Both modules yield the Output-Value. Third, is the *strategy definition module*, which compares the information from the Input- and Output-Values with the defined goals for the given application.

The system developer processes the comparison values and compiles adaptive interventions of the system and presents them to the user according to the values. This is done on the basis of the results of the comparison effected by the strategy definition taken together with user information derived from the *decision module* and the ontology database.

As Figure 1 indicates, an adaptive system sends the users' actions to the UPC. The actions will be stored in the user model within the *Setter module* and represents the Input-values for the *decision module* and the user ontology. The user ontology compares the Input-values with the stored taxonomies and allocates a cluster value as one Output-Value. From the *decision module* a conclusion about the user can be drawn, and subsequently the decision value can be allocated as Output-Value, also.

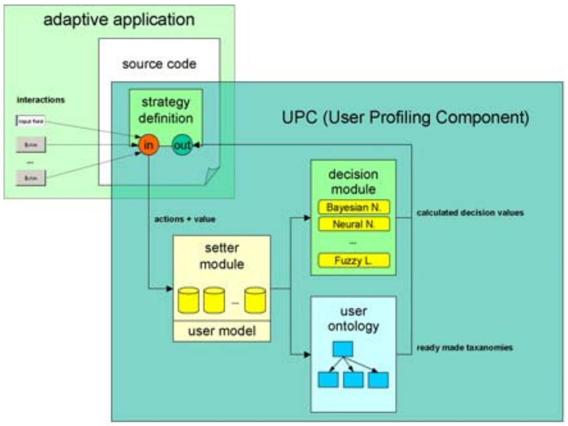

Figure 1. Data flow within the UPC

The feedback values both, from the decision module as well as from the user ontology are compared with the defined adaptive strategies and the current situation of the user. One example for such a defined strategy might be: show a student all low budget travel options. The setter module tells the UPC the type of user and which actions the user has selected. The comparison-process proposes into the Output-Value, which travel plans and which representation form the user prefers.

The User Profiling Component *Control Types:* Algorithms need to be developed for all adaptive methods described in Table 2 in such a way that, through the application of user ontologies, they lead to faster inferences. This means that the algorithms must be adjusted to allow the integration of ontologies, and this must be done in such a way as to preserve flexibility: the system should not need expensive implementation changes in order to incorporate changes in the user ontology.

To meet these requirements we have developed a new methodology of what we call Control Types,[17] which form a general framework for combining a plurality of adaptive methods in such a way that the advantages of each can be preserved while user ontologies are incorporated. For example it can be used to combine the method of stereotyping with that of Bayesian Nets in such a way as to improve the speed of a stereotyping algorithm's operation.

To achieve this end, a new language has been developed for describing the conditions realized in the course of interactive dialogues. This language includes the facility to use Control Type functions in such a way as to establish communication with user profile entries.

The end-result realizes the goal of adaptation via rules of the form *if-then-else*. The following example illustrates the use of Ontology by the rule description language in representing a simple dialogue:

```
<def>on=new(ontology)</def>
<if>(#um.game[1].count==2)
<dialog form=on.getPresentationForm(
on.getCluster(#um))>id(0)</dialog>
</if>
```

Source 1. Ontology-Interface

Or in English: if training situation 1 (game[1]) has been completed twice then the dialogue with the identifier 0 is to be displayed; otherwise, the dialogue with the identifier 1 is to be displayed. Here the dialogue is formatted in relation to the current ontology cluster, e.g. nurses, thereby calling forth information that is in part different from that associated, e.g. with the class doctors.

CONCLUSION

The advantages of the method outlined above include its easy adaptability, high speed of operation, automatic completion of inferences to yield new information, and automatic extendibility. We believe that these advantages outweigh the disadvantage in terms of high implementation costs, so that the construction of adaptive algorithms with integrated user ontologies can be expected to serve as a valuable tool for system developers in the future.

REFERENCES

1. Kobsa A. Adaptivität und Benutzermodellierung in interkativen Softwaresystemen. Proceedings of KI 1993. Humboldt University, Berlin. 1993.
2. Nebel IT, Blüher M, Paschke R. Evaluation of a computer based interactive diabetes education program designed to train the estimation of the energy or carbohydrate contents of foods, Pat Educ Couns 1519, Elsevier, 46: 55-59: 2002.
3. Meyer K. et al. Training of celiac patients: comparative analysis of conventional training and computer based interactive training for celiac disease patients, The American journal of clinical nutrition, 2003.
4. Nebel IT. Implementation und Evaluation eines adaptiven Diabetes-Schulungsprogramms auf der Basis der Benutzermodellierung mittels Controltypes, Projektbericht, KI-Zeitschrift, 2/02: 40-43: 2002.
5. Issing LJ et al. Information und Lernen mit Multimedia und Internet, Beltz, Weinheim, 2002.
6. Wahlster W. Cooperative access systems, Future generations in computer systems 1, Amsterdam, 1984.

7. Wahlster W, Kobsa A. User models in dialog systems, Springer, Berlin, 1989.
8. Blurock ES, Course: Machine learning, Research Institute for Symbolic Computation, 2000.
9. Strecker S, and Schwickert AC. Künstliche Neuronale Netze – Einordnung, Klassifikation und Abgrenzung aus betriebswirtschaftlicher Sicht, 4/1997, Justus-Liebig-University Gießen, 1997.
10. Lebowitz M. Concept learning in a Rich Input Domain: Generalization-based memory, in: Boulay B. Advances in artificial intelligence II, Elsevier Science Publishers B. V., 1986.
11. Gruber T. What is an Ontology? http://www-ksl.stanford.edu/kst/what-is-an-ontology.html.
12. Baader F. et al. The description logic handbook, Cambridge: Cambridge University Press, 2003.
13. Guarino N, and Giaretta P, 1995. Ontologies and knowledge bases: Towards a terminological clarifycation. In N Mars (ed.), Towards very large knowledge bases, Amsterdam: IOS Press.
14. Smith B. Ontology, in L. Floridi (ed.), Blackwell guide to philosophy, information and computers, Oxford: Blackwell, 2003.
15. Smith B. 2002 Basic Formal Ontology, http://ontology.buffalo.edu/bfo.
16. Eliot CR, Williams KA, Woolf BP. An intelligent learning environment for advanced cardiac life support, Proceedings AMIA Annual Fall Symposium, 7-11, 1996.
17. Nebel IT et al. Comparative analysis of conventional and adaptive computer-based interactive hypoglycaemia education programs, Pat Educ Couns, 2003.

Acknowledgement

This work is supported by the Alexander von Humboldt Foundation under the auspices of its Wolfgang Paul Program and by the Formel.1-Program of the German Ministry of Education and Research.