A User Profiling Component with the aid of User Ontologies

István-Tibor Nébel[1], Barry Smith[2,3], Ralf Paschke[1]

[1]Medical Learning and Informationsystems, 3rd Department of Medicine, University of Leipzig, Leipzig, Germany

[2]Institute for Formal Ontology and Medical Information Science, 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 initially 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 readymade taxonomies can be derived in such a way as to enable associated software to support a variety of different types of users.

1 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 these users be beginners or experts [Kobsa, 1993].

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 [Nebel *et al.*, 2003] and celiac diseases. The program 'Hypoglycaemia', designed to facilitate patients' learning from hypoglycaemias of diabetes mellitus, is able to adjust itself to each user's current state of knowledge. Its practicability has been evaluated on the basis of the learning success of 120 diabetes patients [Nebel, 2002], 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. This provides evidence for the thesis instructional programs are better able to support 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.

2 Generating User-Profiles

Each individual observes the world in his own fashion and each individual user brings diverse needs and expectations to his interaction with a software system. The determination of the requirements of each individual user is therefore an important prerequisite for an efficient application of a tutorial system [Issing, 2002].

To this end a user classification is needed, which can serve as a basis for an adaptive system; this must save and analyze the data pertaining to each user and make available information relevant to the program's adaptation to the user in each successive stage ([Kobsa, 1993], [Wahlster, 1984], [Wahlster *et al.*, 1989]). The data are stored for the most part in the form of attribute-value pairs, which represent the user's current state of knowledge as well as his personal characteristics, features, preferences, and so forth. A collection of such attribute-value pairs constitutes a user profile, a representation of an individual user, of user attributes and competencies, and of the stage the user has reached in his interaction with the program. In addition to such user-profiles the systems needs a general user classification in terms of which the user profiles can be organized and interpreted.

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

3 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 can be classified broadly into stereotype-based, rule-based [Blurock, 2000], and mathematical and statistical approaches.

From this it is clear that there exists a variety of procedures for establishing user models or user profiles. The mathematical procedures can include also the application of statistical or probabilistic methods in order to generate assumptions about users under conditions of uncertainty.

The methods which use user characteristics for drawing conclusions about a user, such as stereotypes allow us to implement ontologies in support, because the user at-

tributes to detecting the relevant stereotype are the same for detecting the associated cluster of user ontology.

4 The Method of Stereotypes

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 [Lebowitz, 1986]. 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.

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. Such clustering methods allow new stereotypes to be created on the fly, and thus to be included within a continuously evolving stereotype hierarchy.

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. If we are to control for this some 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-appear in every class of users.

5 Towards User Ontologies

In light of the problems, we propose a new method for the creation of user-profiles through the construction of a central database resource 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 new user ontologies can be added and existing taxonomies of user-characteristics and user-types can be updated in light of the actual experience of users and system developers. This approach has the 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 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 [Gruber, 1986], often by utilizing the machinery of one or other description logic (DL) [Baader, 2003].

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.

In the domains of e-recruiting and human resource development competency ontologies have been developed. They have been used for example within Semantic Web services environment as a basis for just-in-time learning. A competency ontology is a rich, semantic description of the competencies an employee must possess in order to participate in specific activities of the business processes of a company [Woelk, 2002].

The term 'ontology' is of course also used in philosophical circles, where it refers to the study of 'the nature and organization of reality' [Guarino *et al.*, 1995]. Ontology in the philosophical sense attempts to discover theories that match the domain of reality under examination [Smith, 2003]. 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, drawing also on existing work in the competency ontology domain.

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, such as interests, knowledge, preferences and so on.

An adaptive software system must be in a position 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. Similar to our ontology are the PAPI (public and private information)- and the IMS-project. The PAPI determined user profiles by using meta datas [Dolog, 2003]. The IMS-Project (IMS Learner Information Packaging Information Model Specification) which was described in [IMS, 2001] is a collection of information about a Learner or a Producer of learning content. The IMS Learner Information Package (IMS LIP) specification addresses the interoperability of internet-based Learner Information systems with other systems. The intention of the specification is to define a set of packages that can be used to import data into and extract data from an IMS compliant Learner Information server.

We have used a different approach to adaptive software systems. In the spirit of philosophical uses of the term 'ontology' we will first 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: [Smith, 2002]):

1. types of users

2. characteristics of users

   (a) permanent (independent of experience with the software system)
      i. interests
      ii. attitudes
      iii. personality
      iv. skills
      v. knowledge
      vi. abilities
      vii. preferences
   (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 [Eliot *et al.*, 1996] 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.

6   Software Application

The Adaptation Process: The principal procedure of an adaptation process is that of user-profiling. The working process follows the universal principle of *observing the user–reasoning–storing–intervening with 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 *user profiling component (UPC)*. The actions will be stored in the user profile 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. This implemented ontology is a software component within the UPC. From the *decision module* a conclusion about the user can be drawn, and subsequently the decision value can be allocated as Output-Value, also.

The feedback values from both the decision module and the user ontology are compared with the developer-defined adaptive strategies and the current situation of the user. The implemented ontology One example for
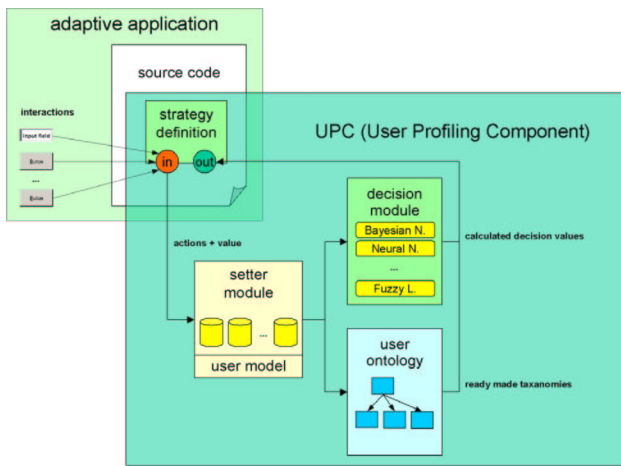
Figure 1: Data flow within the UPC

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 adaptive methods 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 [Nebel, 2003], which form a general framework for combining a plurality of adaptive methods in such a way that the advantages of each can be preserved when ontologies are incorporated. For example it can be used to combine the method of Stereotyping with that of Neural Networks 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 language of control types in representing a simple dialogue:

Source 1 (Ontology-Interface)

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

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 some given ontology cluster, e.g. that associated with

nurses, thereby calling forth information that differs from that associated, say, with the class doctors.

7 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.

Acknowledgments

References

[Kobsa, 1993] A. Kobsa. *Adaptivität und Benutzermodellierung in interkativen Softwaresystemen*. Proceedings of KI 1993, Humboldt University, Berlin. 1993.

[Nebel *et al.*, 2003] IT. Nebel, M. Blüher, and R. Paschke *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.

[Nebel, 2002] IT. Nebel. *Implementation und Evaluation eines adaptiven Diabetes-Schulungsprogramms auf der Basis der Benutzermodellierung mittels Controltypes*. Projektbericht, KI-Zeitschrift, 2/02: 40-43: 2002.

[Issing, 2002] LJ. Issing. *Information und Lernen mit Multimedia und Internet*. Beltz, Weinheim, 2002.

[Wahlster, 1984] W. Wahlster. *Cooperative access systems, Future generations in computer systems*. Amsterdam, 1984.

[Wahlster *et al.*, 1989] W. Wahlster and A. Kobsa. *User models in dialog systems*. Springer, Berlin, 1989.

[Blurock, 2000] ES. Blurock. *Course: Machine learning*. Research Institute for Symbolic Computation, 2000.

[Strecker *et al.*, 1997] S. Strecker and AC. Schwickert. *Künstliche Neuronale Netze - Einordnung, Klassifikation und Abgrenzung aus betriebswirtschaftlicher Sicht*. 4/1997, Justus-Liebig-University Giessen, 1997.

[Lebowitz, 1986] M. Lebowitz. *Concept learning in a Rich Input Domain: Generalization-based memory, in: B. Boulay. Advances in artificial intelligence II*. Elsevier Science Publishers B. V., 1986.

[Gruber, 1986] T. Gruber. *What is an Ontology?* http://www-ksl.stanford.edu/kst/what-is-an-ontology.html.

[Baader, 2003] F. Baader. *The description logic handbook*. Cambridge: Cambridge University Press, 2003.

[Guarino *et al.*, 1995] N. Guarino and P. Giaretta. *Ontologies and knowledge bases: Towards a terminological clarifycation. In N. Mars (ed.), Towards very large knowledge bases*. Amsterdam: IOS Press, 1995.

[Smith, 2003] B. Smith. *Ontology*, in L. Floridi (ed.), Blackwell guide to philosophy, information and computers. Oxford: Blackwell, 2003.

[Smith, 2002] B. Smith. *Basic Formal Ontology*. http://ontology.buffalo.edu/bfo, 2002.

[Eliot *et al.*, 1996] CR. Eliot, KA. Williams, and BP. Woolf. *An intelligent learning environment for advanced cardiac life support*. Proceedings AMIA Annual Fall Symposium, 7-11, 1996.

[Nebel, 2003] IT. Nebel. *Comparative analysis of conventional and adaptive computer-based interactive hypoglycaemia education programs*. Patient Education and Counselling, in press, 2004.

[Woelk, 2002] D. Woelk. *e-Learning, Semantic Web Services and Competency Ontologies*. Proceedings of the ED-MEDIA, Denver, CO, 2002.

[Dolog, 2003] P. Dolog, R. Gavriloiaie, W. Nejdl, J. Brase. *Integrating Adaptive Hypermedia Techniques and Open RDF-baed Enviroment*. Proceedings of The 12th International WWW-Conference 2003, Budapest, Hungary, ACM, 2003.

[IMS, 2001] IMS Global Learning Consortium, Inc. *IMS Learner Information Packaging Information Model Specification – Final Specification Version 1.0*. http://www.imsproject.org/profiles/lipinfo01.html, 2001.

Supplementary Tables

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

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