User Modeling and User-Adapted Interaction **4:** iii–v, 1995.
© 1995 *Kluwer Academic Publishers. Printed in the Netherlands.*

Editorial

ALFRED KOBSA

The development of comprehensive user modeling components in application systems can be quite tedious and expensive if they have to be built from scratch. User modeling shell systems promise help in this respect. When filled by the developer with application-specific user modeling knowledge, user modeling shell systems provide a number of services to the application, including the following ones that can be frequently found:

- the representation of assumptions about one or more types of user characteristics in models of individual users (e.g., assumptions about their knowledge, misconceptions, goals, plans, preferences, tasks, and abilities);
- the representation of relevant common characteristics of users pertaining to specific user subgroups of the application system (so-called stereotypes);
- the classification of users as belonging to one or more of these subgroups, and the integration of the typical characteristics of these subgroups into the current individual user model;
- the recording of users' behavior, particularly their past interaction with the system;
- the formation of assumptions about the user based on the interaction history;
- the generalization of the interaction histories of many users into stereotypes;
- the drawing of additional assumptions about the current user based on initial ones;
- consistency maintenance in the user model;
- the provision of the current assumptions about the user, as well as justifications for these assumptions;
- the evaluation of the entries in the current user model, and the comparison with given standards.

The contributions to the first part of this special issue describe user modeling shell systems that offer several of the above services to a higher or lesser extent. Orwant's DOPPELGÄNGER system focuses on recording users' behavior and generalizing this behavior into user patterns that can be used for prediction (pages 107–130). The BGP-MS system (described in the article by Kobsa and Pohl, pages 59–106) emphasizes complex user model representation and inference methods, as well as stereotype mechanisms. In the second part of the special issue (issue 4:3) Kay's um system allows the application developer to link powerful evaluation processes into the user modeling component. And Paiva's and

Self's TAGUS system, finally, is particularly strong in the area of reasoning and consistency maintenance.

This special issue of UMUAI does not cover all systems that fall under the category of user modeling shells. A comprehensive list would at least include GUMS (Finin, 1989), UMT (Brajnik and Tasso, 1994) and PROTUM (Vergara, 1994). If one also considers user modeling *tool* systems that provide a more limited user modeling functionality, then at least (Huang *et al.*, 1991) and (Y. Kono and Mizoguchi, 1994) must be mentioned as well.

As with any other software tools, developers must carefully study the user modeling needs of their application and compare them with the services of each user modeling shell system in order to decide whether and which shell should be used. Future shell development will be strongly driven by experiences with first practical applications of the current shell systems and the resulting "market demands". Developments that one can anticipate include the following:

- The communication between the application and the user modeling shell will have to be standardized. This will allow application developers (a) to clearly define the boundary between the application and the user modeling components; (b) to delay the selection of a user modeling shell until the user modeling needs of the application are clearly defined; and (c) to switch to a different user modeling shell when the user modeling needs of the application increase or decrease, or when a new shell system with the same functionality becomes available that is preferable for other reasons.
- Shell systems will have to cater better to uncertainty and possibly even vagueness in user modeling, since at least the former seems to be inherent in assumptions about users.
- Shell systems will have to provide more and better services for the generation of assumptions about users and user groups based on (experimentally controlled) interaction with an application system.
- Shell systems will have to deal with the fact that users employ applications not only on networked computers which allow for centralized user models, but also on unconnected or only occasionally connected computers, and on platforms whose standard operating system does not request a formal user login and therefore cannot trivially detect that the user has changed.
- Shell systems will have to explicitly deal with the fact that users employ several applications (and sometimes even several instances of the same application) interchangeably at the same time.
- Shell systems will have to consider security, access and privacy issues to a much larger extent than this has been the case so far.

The shell systems described in this issue already provide very exciting solutions for many user modeling requirements. Quite a few challenges have, however, still to be met before user modeling shells can become useful tools for real-world applications.

References

G. Brajnik and C. Tasso. A shell for developing non-monotonic user modeling systems. *Int. J. Human-Computer Studies*, 40:31–62, 1994.

T. W. Finin. GUMS: A general user modeling shell. In A. Kobsa and W. Wahlster, editors, *User Models in Dialog Systems*, pages 411–430. Springer, Berlin, Heidelberg, 1989.

Xueming Huang, Gordon I. McCalla, Jim E. Greer, and Eric Neufeld. Revising deductive knowledge and stereotypical knowledge in a student model. *User Modeling and User-Adapted Interaction*, 1(1):87–115, 1991.

H. Vergara. PROTUM: A prolog based tool for user modeling. WIS Memo 10, WG Knowledge-Based Information Systems, Department of Information Science, University of Konstanz, Germany, 1994.

M. Ikeda Y. Kono and R. Mizoguchi. Themis: A nonmonotonic inductive student modeling system. *Journal of Artificial Intelligence in Education*, 5(3), 1994.