Artificial Intelligence Review **14:** 149–152, 2000.
Stephen J. Hegner et al. (eds.), Intelligent Help Systems for UNIX.

Editorial
Intelligent Help Systems for UNIX: Planning and Knowledge Representation

STEPHEN J. HEGNER[1], PAUL MC KEVITT[2], PETER NORVIG[3] and ROBERT WILENSKY[4]
[1]*Umeå University, Sweden;* [2]*University of Ulster, Northern Ireland;* [3]*NASA Ames Research Center, California, USA;* [4]*University of California at Berkeley, USA*

This is the second of a series of three special issues on *intelligent help systems for UNIX*.[1] This issue addresses *planning and knowledge representation* whereas the first issue focussed on *computational models and systems* and the next will be on *natural language dialogue*. The papers in this issue are concerned with discovering what the user wants to do, and figuring out a way to do it as well as representing the knowledge needed to do so.

A *passive* consultant accepts questions from the user, but is otherwise ignorant of the user's goals. An *active* consultant continually monitors the user's actions, and tries to discover the user's goals from these actions. The contribution of **Hecking** is a theoretical analysis of the problem that an active consultant faces, the *plan recognition* problem. He describes two modules, one to recognize plans, and the other to advise the user of possible alternative plans that would achieve the same goal more efficiently. Along the way, a representation of plans, subplans, goals, actions, properties and time intervals is developed. In sum, all the results from the planning subfield of artificial intelligence (AI) are applicable to the UNIX consultant task.

Hegner discusses plan realisation in the UNIX help domain, as embodied in the Yucca-* help system. The primary goal of this system is to provide detailed expert advice on the use of the UNIX command language for complex queries, the solution of which may involve interconnection of several commands, each with multiple options. This requires not only the representation of dynamic knowledge about command behaviour at a level of detail sufficient to support solution of the query, but also a planning mechanism capable of interconnecting such knowledge into a cohesive solution. A key to realising such behavior is the observation that the form of knowledge representation necessary to support modelling of communication with the user is quite different than that which is appropriate for the representation of details about the behavior of an operating system. Yucca-* therefore embodies a two-

level architecture, comprised of an *understander* unit for interfacing with the human, and a *formal knowledge and solver* unit for detailed knowledge about UNIX. The paper focusses on the design of the formal knowledge and solver unit, with particular emphasis on aspects of the problem unique to the UNIX domain. It is argued that very general techniques (weak methods) are not sufficient in complex domains. Domain-specific shortcuts, such as clichés, are often needed to gain acceptable performance. Again, the UNIX consultant task provides an example of this AI truism.

Jones et al. describe a mechanism for identifying the explicit assumptions about the user which are necessary to account for the various hypotheses concerning what the user is thinking at every stage of interaction with an intelligent help system. The consistency of the assumptions is managed by an Assumption-based Truth Maintenance System (ATMS). Selection among different user models is based on the information which is extracted from the feedback that UNIX provides in response to user actions. The mechanism is used by the user modelling component of such a help system. First-order logical representations, such as those employed by Hecking and Hegner, are not well suited in cases where properties involve default values. Typically, the logic must be extended with some kind of non-monotonic component. One might think that such an extension is not necessary in the UNIX domain, since all commands have a well-defined semantics. However, while the result of an action on a given state of the world is well-defined, knowledge about the state of the world is not. The user may have misconceptions about how commands work, or about what is true in the current state. The work of Jones et al. searches for a model of the user which is consistent with the current state, while requiring the fewest number of non-default assumptions about the user's knowledge.

Kemke describes knowledge representation in the SINIX Consultant, an intelligent help system for the SINIX operating system, a UNIX derivative developed by Siemens AG. The SINIX Consultant answers natural language questions about SINIX concepts and commands and also gives unsolicited advice. The knowledge base is a taxonomical hierarchy of SINIX concepts, divided into objects and actions operating on these objects. A single concept in the knowledge base is described by a set of attributes reflecting structural or syntactical features, the use, application and purpose of the command or object, and additional information for explaining the concept to the user. In comparison to Yucca-*, the SINIX Consultant embodies much more detailed knowledge about the interactive behavior of complex individual commands, such as mailers, while the emphasis of Yucca-* is upon knowledge which allows one to combine relatively simple independent commands (e.g., *ls*, *cat*) together to form complex solutions.

The fundamental lesson which may be extracted from the papers of this issue is that the form of knowledge representation which is required depends heavily upon the goals of the system. In particular, those systems which have, as a goal, the delivery of complex technical information about UNIX (e.g., Yucca-* and the SINIX consultant) must embody a design fundamentally different from those which focus primarily upon modelling the user (e.g., the system of Jones et al., as well as a number of components of the UC system described in the previous issue.)

The articles in this issue are derived from papers originally presented at a workshop entitled *Knowledge representation in the UNIX help domain*, organised by Peter Norvig, Wolfgang Wahlster and Robert Wilensky at the University of California, Berkeley, USA, in December, 1987. The workshop was funded, in part, by the International Computer Science Institute and all participants were invited. The area of intelligent help systems for UNIX provides such a fruitful example domain of application for many AI techniques that we have decided to publish this work, which is still timely, more widely and particularly now where we have many spoken dialogue systems applied to such fixed domains (see Bernsen et al. 1998; BusinessWeek 1998).

Each article here has been reviewed by the editors and has been subsequently revised; furthermore, all authors have been asked to include a section on *recent developments* on their work. Related work which may be of interest to the reader can be found in Kobsa and Wahlster (1988) which focusses on user modelling and in Maybury and Wahlster (1998) which is a comprehensive publication on intelligent user interfaces. We regret John Jones passed away in 1994 in a climbing accident in Peru. Members of the Kluwer editorial and publishing staff are to be thanked for their help in producing this issue. It would not have been possible without Melanie Willow (Journals Editorial Office), André Diepenbroek (Desk Editor, Editorial Department), and Vanessa Nijweide (Kluwer Publishing Editor, Humanities and Social Sciences). Melanie and André have worked very hard with authors and reviewers to enable this to happen.

The Editors, June 2000

Stephen J. Jegner
Umeå University
Sweden

Paul Mc Kevitt
University of Ulster
Northern Ireland, UK

Peter Norvig
NASA Ames Research Center
California, USA

and

Robert Wilensky
University of California at Berkeley
USA

Note

[1] UNIX is a trademark of X/Open, Inc.

References

Bernsen, N-O., Dybkjær, H. and Dybkjær, L. (1998). *Designing Interactive Speech Systems: from First Ideas to User Testing*. Springer-Verlag: New York, NY.

BusinessWeek (1998). *Speech Technology is the Next Big Thing in Computing, Special Report*, Feb. 23. McGraw-Hill: New York.

Kobsa, A. and Wahlster, W. (1988). *User Models in Dialog Systems*. Springer-Verlag: Berlin.

Maybury, M. and Wahlster, W. (eds.) (1998). *Readings in Intelligent User Interfaces*. Morgan Kaufmann Publishers: Los Altos, CA.