Artificial Intelligence Review **14:** 1–4, 2000.
Stephen J. Hegner et al. (eds.), Intelligent help systems for UNIX.
© 2000 *Kluwer Academic Publishers. Printed in the Netherlands.*

Editorial
Intelligent Help Systems for UNIX: Computational Models and Systems

This is the first in a series of three special issues focussed on *intelligent help systems for UNIX*,[1] each with its own emphasis: (1) computational models and systems, (2) planning and knowledge representation and (3) natural language dialogue. In this first issue focussing on *computational models and systems* there are five papers, one addressing empirical foundations, another virtues and problems, with the final three describing comprehensive implemented systems.

Jerrams-Smith presents some empirical studies conducted on UNIX users which resulted in a taxonomy of error types so that users' errors could be classified. In turn, this information is encapsulated as production rules within a knowledge base which forms the basis for an intelligent interface to UNIX. The prototype makes inferences about users' mental models and uses these to select appropriate tutorial advice. A comparison was conducted between users of the prototype intelligent interface and the usual UNIX interface. Users of the prototype made fewer errors, had fewer misconceptions and took less time to complete given tasks.

Virvou et al. look at the virtues and problems of an active help system for UNIX. A *passive* consultant accepts questions from the user, but is otherwise ignorant of the user's goals, while an *active* consultant continually monitors users' actions, attempting to discover their goals from these actions. An empirical study encompassing a cross-section of UNIX users at an academic site showed the need for an active, rather than passive, help system. Examples of system interaction supporting this view are given, and a discussion of the help needed in these examples is provided. It is proposed that to provide such help requires the construction and maintenance of a model of each user.

The next three papers focus on comprehensive systems which have been built to perform as intelligent help systems for UNIX. In the first, **Wilensky et al.** describe UC (UNIX Consultant), a natural-language interface that allows naive users to learn about the UNIX operating system. The development of UC was undertaken because the task was thought to be a useful domain for AI research, as well as a useful application of AI work in planning, reasoning, natural language processing, and knowledge representation. UC contains a number of components for language understanding (ALANA), inference (concretion mechanism), goal and plan analysis and generation (PAGAN, UCEgo), domain planning (KIP), content expression (UCExpress), and language generation (UCGen), user's knowledge state with respect to UNIX (KNOME), and enabling addition of knowledge of both English and UNIX facts to UC's knowledge base (UCTeacher). All components of UC make use of a knowledge representation system called KODIAK.

Mc Kevitt describes OSCON (Operating System CONsultant), which is also a natural-language operating system consultant capable of answering queries on over 40 commands from both the UNIX and MS-DOS[2] operating systems. OSCON differs from UC in that it is specialised more on the domain of operating systems, whereas UC has the broader goal of studying knowledge representation, planning and natural language processing through operating systems. UC employs a much more general knowledge representation and planning mechanism with a more cognitive flavour. Mc Kevitt also points to more recent developments where empirical studies with subjects using UNIX through Wizard-of-Oz techniques give frequencies of types of questions asked and enable the incorporation of user modelling into OSCON.

Matthews et al. present USCSH (University of South Carolina SHell), an active intelligent help system for UNIX. USCSH can operate in two modes, the active mode and the intelligent mode. In the *active mode*, USCSH monitors users' interactions and at appropriate times makes suggestions on how the user might better use the system. This sense of active is similar to that used by Virvou et al. above. In the *intelligent mode*, the system accepts natural language questions and responds to them, taking into consideration the context and user's ability. A trend that has been showing up in commercial applications of AI: translating a working prototype into a language that can be more easily embedded into a running application, was employed in the development of this system. A prototype in Lisp was rewritten in C so that it could be called directly from the C shell.

To sum up, there are a number of ways of looking at a domain of application such as UNIX, depending on whether one is interested in modelling the domain itself or the cognitive models applied to the domain. Also, empirical studies are useful for obtaining knowledge about how subjects interact with domains such as UNIX, and then this knowledge can be incorporated within systems. Finally, "active" rather than "passive" help systems are more appropriate for situations in which the system attempts to model the plans of the user over time. The latter problems of planning and knowledge representation will be the subject of the next special issue on intelligent help systems for UNIX.

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. Members of the Kluwer editorial and publishing staff are to be thanked for their help in producing this issue. We regret that John Jones passed away in 1994 in a climbing accident in Peru. A special thanks to David Chin for preparing the paper on UC herein.

The Editors, April 2000,

Stephen J. Hegner
Umeå University
Sweden

Paul Mc Kevitt
The Queen’s University of Belfast
Northern Ireland

Peter Norvig
NASA Ames Research Center
California, USA

and

Robert Wilensky
University of California at Berkeley
USA

Notes

[1] UNIX is a trademark of X/Open, Inc.
[2] MS-DOS is a trademark of Microsoft Corporation.

References

Bernsen, N-O., Dybkjær, H. & 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*. BusinessWeek, Feb. 23. New York: McGraw-Hill.

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

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