WHERE SHOULD THE INTELLIGENCE IN INTELLIGENT INTERFACES BE PLACED?

Robert Neches
Thomas S. Kaczmarek

USC / Information Sciences Institute
4676 Admiralty Way
Marina Del Rey, CA 90292
Phone: (213) 822-1511

PURPOSE OF THE PANEL

Obviously, many AI techniques are applicable to building better human-machine interfaces. Intelligent interfaces have explored the use of problem solving, planning, heuristic search, discourse models, user models, knowledge representation, expert systems, and natural language text understanding and generation. The particular techniques researchers have chosen have led to two very different paradigms in intelligent interfaces. On the one hand, knowledge based, "intelligent apprentice" systems like the UNIX Consultant seek to provide assistance based on an understanding of the user's intentions and task domain. On the other hand, "power tool" systems like COUSIN emphasize a powerful command set, but leave the responsibility for selecting and applying those commands in the hands of the user. Behind these distinctions are some very different fundamental assumptions about the problems that users need help with, and the AI techniques that can be applied to helping them.

If our understanding of intelligent interfaces is to grow, it is important that we continue to examine such distinctions in greater detail. The purpose of this panel is to extend the discussion of interaction paradigms in two directions: first, by trying to bring out more specific issues within these paradigms and by presenting alternative paradigms, and second, by focusing attention on intelligent interfaces to a variety of kinds of systems including knowledge-based systems.

Thus, the kinds of questions panel members will address include:

- What are the fundamental problems of users that intelligent interfaces ought to be addressing? Understanding the system? Correcting errors? Identifying referents of discourse? Keeping track of the state of plans and goals during interleaved activities?

  Planning how to use a system to accomplish a particular goal?
- What are the general approaches towards addressing those problems? For example, one might approach the understandability issue by trying to broaden the range of inputs handled so that users can get by with less knowledge, or by building in more self-knowledge so that systems could generate justifications of their own reasoning and behavior as done in expert systems.
- How can particular AI techniques help realize those approaches? This question is not to be answered at the cliche level of "natural language would save users from having to learn a command language", but rather at the level of particular solutions in particular systems.
- How can the intelligence that is built into knowledge-based systems be exploited to provide a better user interface? It would seem that sharing intelligence between the interface and the system would reduce cost and enhance consistency. Is this true? Are there special concerns that must be addressed when interfacing to intelligent systems? Many expert system designers have added explanation capabilities based on the belief that users need to build confidence in the system and to learn from it. Are there other needs or other consequences? Are there other kinds of knowledge-based systems that have distinctive interface needs?

The panelists have been chosen to represent a wide range of experiences with interface design. Each has developed a uniquely different style of interaction. There is also great diversity in the kinds of systems for which the panelists have created interfaces. Following is a description of the experience and interests of the panelist:

- John Seely Brown has a long history of involvement with intelligent interaction systems dating back to his work on SOPHIE in the early '70s. Combining his interests in the semantics of procedures and his collaboration with the Operability Project at Xerox PARC, he is currently concerned with the intelligibility of interfaces and support for communicative repair in user-system interaction.
- Phil Hayes has been involved with the construction of cooperative user interfaces. His work on COUSIN is a pioneering effort in the construction of user interface management systems. It is a prime example of the power tools approach of interface design. His current interests include interfaces to knowledge-based systems and combining natural language with other command modes.
- Tom Malone's interests have been aimed at organizational computing and the kinds of interfaces needed in such environments. His work on the Information Lens exemplifies the power tools approach. He also influenced RABBIT, which is a blend of the power tools and intelligent assistant approaches. His current interests involve understanding the impact on the interface that results from the distinctions between expert support systems and expert systems and also in contrasting natural intelligence with artificial intelligence.

- Dick Waters has contributed significantly to the programming environment through his work on the Programmer's Apprentice and KBEmacs. The interface defined in this work combines aspects of power tools and intelligent assistance. Since the interface combines both aspects, there is an interest in discovering and separating those aspects of the environment to provide assistance for from those to provide tools for.
- Bob Wilensky has worked extensively on natural language interfaces. His work on the UNIX Consultant is a prime example of an intelligent assistant. The emphasis in this work has been on the naive user.
- Mike Williams has contributed in the development of intelligent interfaces to data bases that do not involve natural language. The descriptive retrieval paradigm found in the RABBIT system employs an interface that presents an intelligent assistant that the user communicates with through a set of power tools. It provides the feeling of direct manipulation, as found in power tools, but uses an understanding of the data base to provide the power.