Intelligent User Interfaces

Joseph W. Sullivan and Sherman W. Tyler (editors)
(Lockheed Missiles and Space Company)

New York: ACM Press and Reading,
MA: Addison-Wesley (ACM Press
Frontier Series), 1991, xv + 472 pp.
Hardbound, ISBN 0-201-50305-0, no
price listed

Reviewed by
Paul Luff
Rank Xerox Cambridge EuroPARC and
University of Surrey

Given the range of systems, tools, and techniques discussed in this collection, it appears that the editors have adopted quite a broad definition of the notion of 'intelligence.' Although the book contains descriptions of computer systems that would be conventionally classified as intelligent—those that aim to develop models of their users—it also includes descriptions of sophisticated tools for assisting with building interfaces, for defining human–computer interaction styles, and for presenting information on computer displays. Consequently, the collection appears to be intended not only for practitioners in AI but also for those in Human–Computer Interaction (HCI).

There are two principal reasons why this book may be of interest to computational linguists. Firstly, some of the systems described use computational methods for analyzing and generating language; for example, parsers, semantic networks, and text generators. Secondly, as implementations of techniques from artificial intelligence, the theoretical foundations of many of the papers are similar to those used by many computational linguists; for example, both can rely on notions of users' plans and goals and the assumption that it is possible to infer the users' intentions, plans, and goals from their utterances (*cf.* Perrault and Allen 1980). Therefore, intelligent interfaces could be viewed as a testbed for both developed and more novel techniques of natural language understanding and generation.

The introduction to the collection, by Miller, Sullivan, and Tyler, suggests four requirements of intelligent interfaces: human–computer interactions need to be made clearer; interfaces need to support users' plans and goals; information needs to be presented clearly; and interfaces need to be designed and implemented more easily. These requirements are then addressed in the four sections of the book: multi-modal communication; models, plans, and goals; dynamic presentation design; and knowledge-based tools for interface design. Although Miller et al. state that the requirements of intelligent interfaces are essentially those of human–computer interaction research in general, it is unclear how these requirements have been derived. Rather than defining a framework in which to consider each contribution, the editors use the questions to divide up the contents of the book. This lack of a framework has important consequences for the coherence of the collection as a whole; there are obvious overlaps between what is described in individual sections, and it is also unclear how papers in one section relate to those in another.

In this review, only the first two sections of the book will be examined in any detail. These sections contain research and developments that are likely to be of direct interest in computational linguistics. The others, which relate to systems that automatically design system output and tools that aid interface designers, appear to be of more interest to practitioners in HCI. The review concludes by drawing together some of the general concerns raised by the discussion of individual contributions.

1. Summary of Contributions

Apart from Oviatt and Cohen (Chapter 4), all the contributors describe some computer system or tool, most of which would appear to be at the 'prototype' stage of development. Oviatt and Cohen's paper is also distinctive because it develops a rather novel extension of the channel metaphor for communication, and it describes an experiment. By examining different modalities (audiotape, telephone, and keyboard connections), the authors attempt to separate out the effects of interaction in speech-based communication. Although they find significant differences in features such as the frequency of repetitions and pronouns used through these modalities, they do not provide any indication of how such findings could be used for designers of computer systems. Wahlster (Chapter 3) uses the results of an experiment that examined the use of an early version of a system called TACTILUS, which provided for a range of pointing gestures from the user. This experiment suggested that there were ambiguities in these pointing gestures and led to a redesign of the system. In the new system, text and gestures can be used to clarify each other, and different images of pointing icons can be presented to the users to avoid ambiguity. Unfortunately, Wahlster does not give any indication of whether the revised system is more successful.

The pointing gestures that Neal and Shapiro (Chapter 2) analyze in their example military planning system (CUBRICON) are made with a mouse and are thus both more precise and less varied. The authors are not only concerned with developing a multi-modal interface, but also describe mechanisms that automatically select the way information is presented to a user. Given various recent developments in this domain, the examples of CUBRICON's capabilities are of particular interest. For instance, the system can simultaneously display a visual pointing gesture accompanying the word "here" when generating the speech "the Dresden airbase is located here," this occurring while a map of the border region between 'East' and 'West' Germany is presented on a color monitor (p. 31). Furthermore, CUBRICON can also display only what it infers to be 'relevant' information for the user. Thus, luckily for their owners, country cottages are not displayed to the military planners (p. 20). In the paper, Neal and Shapiro give no details of any evaluation of the system. However, the designers appear to recognize that there may be difficulties for the users in presenting information on two screens, in speech, and in text, and therefore various statements are generated to direct the user where to look. It would be interesting to investigate whether these utterances obtrude in the interaction between the user and the system. Similarly, it is not clear that automatic adaptation of the interface necessarily improves usability; it could make using a system rather unpredictable.

Although the papers in this first section are all concerned with relating aspects of language with aspects of interaction such as gestures, the details of the integration in the systems mentioned will probably be insufficient for a computational linguist. However, the authors refer to their other publications written for such an audience. Section 2, therefore, on models, plans, and goals may be of more interest. Both the papers by Tyler et al. (Chapter 5) and Kass and Finin (Chapter 6) propose general architectures for intelligent interfaces. Their approaches are slightly different, Tyler et al. focusing on the components necessary for such a system, and Kass and Finin aiming to show that there are some general rules that can be used to infer users' goals from their actions. Tyler et al. provide more details about their architecture, which is being used to develop a system for emergency crisis management (CHORIS). This system allows for input in natural language as well as through menus, and this is handled by a modification to the Chat-80 system (Warren and Pereira 1981) that can cope with deixis, anaphora, and imperative sentences. Acquiring vocabularies for the system is assisted by a tool used in conjunction with the Chat-80 parser. As with the CUBRICON system, the CHORIS system can adapt the way it presents material to the user on the screen, and also in common with Neal and Shapiro it is unclear how the users, in this case managers of emergencies, cope with a system that can change in such a way. Tyler et al. describe the problems of adaptability, from the point of view of the system, as being when to adapt, what to adapt, and how to adapt (pp. 98–99), yet they ignore the associated issues for the user; for example, how to make what the system is doing apparent to the user. Both Tyler et al. and Kass and Finin are optimistic about the relevance of their models, yet from the evidence presented here, it is still unclear whether such a general model can inform the design of interfaces that are accurate, robust, or useful.

The system described by Chin (Chapter 9) is probably the most detailed example in the collection of an interface attempting to understand the goals and plans of its users. This system is one of the major contributions to what could now be considered a program of research that aims to make the Unix operating system more usable. The Unix Consultant (UC) is a large computer system including a language recognizer, a semantic network, a goal analyzer, a domain planner, a user model, and a language generator. Chin focuses on one module of UC, the intelligent agent component called UCEgo, which can create and execute various plans according to its own goals. For the purposes of the system, goals are defined in terms of states that either the system or the user wishes to achieve. Although Chin states that "almost any state can be a goal, provided that some individual wishes to achieve that state" (p. 186), he still thereby provides the clearest definition of the notion of a goal available in the whole volume. In fact, UCEgo is based on a range of conceptual distinctions such as plans and goals, but also including 'themes' and 'meta-goals.' Themes, which give rise to goals, appear to be the most interesting of these concepts. Chin gives three examples: an ethical theme, a self-preservation theme, and a theme associated with being a consultant. It is unclear how these themes have been derived. Certainly, they do not appear to result from studies of consultancy in natural settings, particularly as the consultant theme gives rise to the 'recurrent' goal of helping the usei. The ethical theme appears mainly to be concerned with preventing users deleting others' files, and the preservation theme with stopping users from altering or even deleting the program itself. However, such themes and their resulting goals necessarily have to be expressed generally and there will always be circumstances for which they cannot account. Chin points to limitations of the system because it cannot assess its own state, *in situ*, and therefore can appear to be less than cooperative. It is also easy to envisage contexts in which users may be given advice on how to delete another's files and cases in which the system would still naively tell users information that could contribute to its own downfall.

Chapters 7 and 8 take different approaches to fitting the system to the plans and tasks of the user. Rather than designing systems that detect the user's goals and then respond according to the assumed knowledge of the user, Bonar and Liffick (Chapter 7) describe two systems that provide a range of capabilities for 'novices' and 'experts' alike. In their program tutoring system called 'Bridge,' various levels are given to the students to traverse. Students move through the levels of the system, displayed graphically, confining the range of goals they could be attempting to achieve. The interface editor of Young (Chapter 8) also provides the user with graphical displays;

these show activities that need to be carried out and a history of the activities that have been performed. Although both these papers present systems that appear to be less 'intelligent' than the others in the first two sections, it is not necessarily the case that they will be harder to use. Rather, by focusing on the presentation and provision of various facilities of the system, the workings of the system may be more transparent and are therefore more predictable.

The remaining two sections of the book concentrate on systems that could further improve the display of information and facilitate the design of better interfaces. Although the papers in these sections propose some novel ideas for human–computer interaction and interface design, they may not be of direct interest in computational linguistics. However, one of the concerns that arises out of the work described in Section 3 is the problem of integrating text and graphical displays, in particular how the interrelationships between the two media can be established and how the appropriate system processing can be ordered. Roth et al. (Chapter 10) specify an architecture based on a rhetorical model of discourse processing (*cf.* McKeown 1985). To achieve cohesion between pictures and text, they have to divide discourse processing into two phases, one for outlining and one for elaboration. Before the second phase, information from a graphics module has to be processed for the elaborated text to refer to the generated pictures. Feiner (Chapter 12) and Mackinlay (Chapter 13) offer alternative architectures to tackle similar problems. Mackinlay suggests that a Blackboard may be the appropriate way of sequencing system activities. Interestingly, in the next section, Hollan et al. (Chapter 14) incorporate a Blackboard into their Human Interface Tools Suite. The fourth section of the book describes a range of tools and techniques for interface designers, including: icon, graphics, gesture, and knowledge editors (Hollan et al., Chapter 14), an interface definition language (Foley et al., Chapter 15), programming interfaces by example (Myers, Chapter 16) and specifying interfaces by reformulation (Yen et al., Chapter 17). Unfortunately, there is no concluding chapter that could draw together any general concerns raised by the four sections of the volume.

2. Discussion

As may have been gathered, a wide range of interfaces is considered to be intelligent in this volume. Tyler et al. suggest that the authors commonly agree that an intelligent interface should be knowledge-based and modular, infer and evaluate the user's goals, and adapt its behavior to users and their tasks (p. 87). Very few of the systems described in the book fit these requirements and, in fact, the only feature they all have in common is modularity. In the last two sections of the book, it is probably better to consider the systems as 'sophisticated' rather than 'intelligent.' Similarly, the notion of 'goal' is used throughout the volume; however, in the later papers it becomes more associated with the system or even the designers than with the users. With critical concepts being used in conflicting ways, it is hard to determine what can and what cannot be considered an intelligent user interface. The book could be considered as a general collection of papers concerning human–computer interaction; however, although several authors are concerned with displaying information in multimedia, using text, speech, and pictures, there is no mention of other, more novel, modes of interaction using videos and projected images (e.g., Ishii 1990; Tang and Minneman 1991).

More importantly there is a lack of empirical evidence that any of the wide diversity of approaches to designing intelligent interfaces actually results in systems that are easier to use. One premise of the book rests on the notion that some systems are too complex to adopt a direct, tool-based style of interaction, and many of the contributors suggest that only intelligent approaches will provide users with the flexibility they require. However, when prototype systems have been built for specific domains, the authors fail to give even anecdotal reports of what the users felt about the systems. It appears that managing to build a system according to a particular architecture is sufficient to warrant optimistic claims for the validity and utility of that architecture.

Perhaps the most important drawback of the collection is the failure to reflect on the theoretical and conceptual debates surrounding intelligent systems. In recent years there have been sustained criticisms concerning the foundations of much work in AI and HCI. For example, Winograd and Flores (1986) describe the limitations to the scope of 'the rationalistic tradition' of intelligent system design, and Suchman (1987) questions the assumptions that human activity is plan-based and goal-oriented; plans are necessarily vague and are used as resources for *post hoc* accounts of social actions. All the papers in this collection ignore these critiques. A debate of such issues would have contributed in clarifying concepts that are left unanalyzed and undefined. Instead, the contributors rely on commonsense notions of actors' plans, goals, tasks, knowledge, and intentions to inform the design of their systems. This might be the only way to proceed, but it may be useful for researchers in this field to explicate their own practices and procedures.

In addition to there being no discussion of alternative theoretical and conceptual frameworks, there is also little debate between the contributors. Although the authors cross-reference one another, they do not discuss in any depth distinctions between their approaches. This is unfortunate, as there do appear to be some interesting underlying differences between the papers; such debates would have provided general themes for the collection and made it more coherent. Instead, the papers tend to focus on architectures for designing intelligent interfaces, but even when discussing these, where again there are significant differences, there is little discussion of the relative merit of the various designs and models.

Given these reservations, it is best to view the collection as a set of proceedings from a workshop. However, the original workshop was held in March 1988, three years before the book was published. Since then, there have been numerous developments in HCI, details of which are available in the proceedings of many recent conferences. There are also some more focused volumes on related themes; for example, on adaptable interfaces (Browne, Totterdel, and Norman 1990) and on interfaces to complex systems (Wier and Alty 1991). It is probable that only the first two sections of the book are of direct interest to computational linguists, and even these might be of insufficient detail. It may be more fruitful to use the volume as a source for finding related work by the same contributors, either written directly for an audience of computational linguists or giving more details of the relevant computational mechanisms. In most cases both seem to be available.

References

Browne, D.; Totterdel, P.; and Norman, M. (1990). *Adaptable User Interfaces*. London: Academic Press.

Ishii, Hiroshi (1990). "TeamWorkStation: Towards a seamless shared workspace." In *Proceedings, CSCW '90*, Los Angeles, 13–26.

McKeown, Kathleen R. (1985). *Text Generation*. Cambridge, UK: Cambridge University Press.

Perrault, C. Raymond, and Allen, James (1980). "A plan-based analysis of indirect speech acts." *American Journal of Computational Linguistics*, 6(3–4), 167–182.

Suchman, Lucy A. (1987). *Plans and Situated Actions*. Cambridge, UK: Cambridge University Press.

Tang, John C., and Minneman, Scott L. (1991). "VideoDraw: A video interface for collaborative drawing." *ACM Transactions* *on Information Systems*, **9**(2), 170–184.

Warren, David H. D., and Pereira, Fernando C. N. (1981). "An efficient easily adaptable system for interpreting natural language queries." Research Paper 155, Department of Artificial Intelligence, University of Edinburgh.

Wier, G. R. S., and Alty, J. (1991). *Human–Computer Interaction and Complex Systems*. London: Academic Press.

Winograd, Terry, and Flores, Fernando (1986). *Understanding Computers and Cognition: A New Foundation for Design*. Norwood, NJ: Ablex; reprinted, Reading, MA: Addison-Wesley.

Paul Luff is a Research Fellow at both the University of Surrey and at Rank Xerox Cambridge EuroPARC. His main research interests are how recent developments in the social sciences could inform system design, in particular how detailed studies of work practices and social interaction could influence novel technologies. He is a joint editor of the collection *Computers and Conversation*, and his principal address is Rank Xerox Cambridge EuroPARC, 61 Regent Street, Cambridge CB2 1AB, UK. e-mail: luff.europarc@rx.xerox.com