Towards Active Assistance

Suresh K. Bhavnani, James H. Garrett, Jr., Ulrich Flemming, Doris S. Shaw

Introduction

The exploding functionality of current computer-aided engineering (CAE) systems has provided today's users with a vast, but under-utilized collection of tools and options. For example, MicroStation, a popular CAE system sold by Intergraph, offers more than 1000 commands including 16 ways to construct a line (in different contexts) and 28 ways to manipulate elements using a "fence". This complex array of functionalities is bewildering and hardly exploited to its full extent even by frequent, experienced users. In a recent site visit to a federal design office, we observed ten architects and three draftsmen using MicroStation. Although all of these users had been formally trained, they all used the system in sub-optimal ways. For example, they consistently neglected commands that could accomplish a task much faster than the one they used (a full account of this visit is forthcoming).

The sub-optimal use of software has been observed in other contexts: text editors [Rosson 84]; command-based CAE systems [Drisis 92]; and Unix [Hanson 84, Greenberg 88] Such broadly available systems are sub-optimally used even by frequent users because of a "production bias" on the users' part [Carroll 87b], and traditional aids are unable to alter this production bias. Training is usually short and marred by low retention. Although on-line help can be useful for experienced users [Houghton 84], it is not effective in helping users learn new commands. Documentation does not focus on the production of real work output [Carroll 86]; as a result, users at every level of experience resist reading instructions and prefer to consult colleagues or a local expert [Hiltz 84]. However, this assistance from colleagues or local experts depends on the availability of, access to, and smooth relationships with such people, as well as a user who is motivated to ask for advice in the first place. The problem is exacerbated when the system continues to grow during rapid version changes, which leave users often with outdated and inefficient ways to perform tasks.

Although there is room for improvement in the design of traditional aiding devices (e.g., the *Minimal Manual* by [Carroll 86]), they are inherently limited due to their passive nature. We argue that most software users, and especially users of complex CAE software, require *active assistance* during their interaction with the system. A CAE system must actively recognize the need for and be able to present better or alternate ways of using the system as a user performs real work tasks. We begin by discussing current research in active assistance systems and argue that the development of such systems is impeded by the absence of knowledge about usability issues. We propose that these issues can be understood mainly by observing users in real-world situations. The expectation is that an understanding of these issues will allow us to develop active assistance systems that not only improve users' productivity, but also the quality of the designs they produce.

Active Assistance Systems

An essential component of an active assistant systems is a monitoring component that tracks the user's interactions with the system. In addition, the system has access to an application domain model and a user model that contains specific information about the current user. The user model can be static (provided a priori to the system), dynamic (derived from interaction with the user) or a combination of both. Finally, the system contains strategic knowledge about the kind of help to be provided in different domain and user contexts.

Possibilities for active assistance are currently being explored in several directions. These include active assistance for system usage, routine tasks, and design space navigation as discussed below.

Active Support for System Usage

Current help systems rely solely on the user to seek help and identify the topic on which help is to be provided. These aids have knowledge of all the commands available in a system, but since they are passive, they are ineffectual for users who are unaware of the sub-optimal use they make of the system.

Active help systems attempt to overcome the shortcomings of existing help systems. They utilize knowledge of both the user and the application to intervene spontaneously with advice or assistance. They achieve this by monitoring and reasoning about the user's actions while these actions are taking place (see [Jones 91], for a detailed discussion of intelligent help systems).

Active Support for Routine Tasks

Design organizations using CAE systems use hundreds or thousands of pre-drawn components (referred to as "cells" in MicroStation) located in cell libraries that are often too large to browse. A user who does not know that a cell already exits often creates a duplicate. To help such users, [Yang 92] have prototyped a cell retrieval system. It recognizes which cells in the cell libraries are similar to the one being drawn and points their existence out to the user. Such a system will obviously allow a user to (1) get a task done more effectively, with higher quality, and in accordance with standards; and (2) help extend the user's knowledge of the CAE tool and the available cell libraries.

A similar idea underlies case-based design, which is the process of retrieving a past solution and adapting it to a similar problem situation [Kolodner 91]. This approach is useful in a design environment where new designs are frequently similar to designs done in the past because they respond to similar constraints or contexts. An active support system could recognize what a user is intending to do and show instances of successful solutions to similar problems or "cases" that may be of interest to the user. This is being explored as part of the SEED system currently under development [Flemming 93].

Active Support for Design Space Navigation

The SEED system mentioned above is an example of a "generative" design system that is able to take a more active part in the development of a design description, especially to take over many of the routine tasks that have to be completed. The system is able to do this because it has access to an explicit problem statement, which enables it to manage goals and constraints on its own (to the degree that they are explicitly stated). We expect that this will speed up the generation of design descriptions considerably and allow users to explore more fully "spaces" of interesting designs and design alternatives. When these design spaces grow rapidly, users can easily lose their way, that is, go down paths they have already explored or overlook those that lead to more promising alternatives. We are experimenting with effective navigation aids that assist users in exploring portions of this space.

The explicit problem statement provides an important basis for this and other types of assistance. It may, in fact, lead to active decision aids able to turn a software system into a true partner in the design process with the ultimate goal of improving not only the design process, but also the *product* of that process.

Usability Issues

Compelling as the case for active assistance systems may be, there are impediments in their development. These impediments reflect a general lack of knowledge about usability issues in the system design community.

The current literature on active assistance is almost completely restricted to issues of system architecture and implementation. Although Carroll and McKendree [Carroll 87a] stress the importance of user studies, user feedback is typically relegated to post-implementation research. This can be a costly mistake. Active assistance systems monitor the user's actions all the time. Usability issues, like those discussed below, must therefore be a focus of system *design* from the start and remain an area of concentration through the entire development and delivery process.

Motivation. While observing a human expert giving advice to a user, [Coventry 89] discovered that being interrupted and told of a better way to do things might not be enough inducement to change the user's behavior. Given their production bias, it is not clear when users will consider it worth their while to suspend work and to learn a more efficient way. Carroll and Rosson [Carroll 87b] introduce several motivational devices that remain unexplored: motivational techniques inherent in games or performance feedback could give intrinsic rewards; "training wheels" and no-risk experimentation could make learning easier; and advice that is task-oriented could play into the user's production bias.

Credibility. Active assistance systems (like all systems that rely on heuristic knowledge) can be wrong or inaccurate at times. These systems should therefore be designed to maintain user confidence for continued use. If users come to believe that the system "knows all", they are more likely to be disappointed with the slightest deficiencies [Carroll 88]). The design of active assistance systems may have to consider several possibilities as to how its advice is to be worded and pay careful attention to that wording. In addition, transparency of reasoning and an explicitly stated rationale may play a role in making clear the limits of the system.

Timing. The timing of intervention is critical and could make the difference between a user being helped or annoyed [Carroll 88]. For example, a user fighting a deadline might not be interested in learning a new command. Giving advice for something the user has already figured out but, for some reason, chosen to ignore, is also very likely to be annoying.

Privacy. Active assistance systems depend on a user model to adapt their advice to individual users. They therefore contain various types of performance measurements and inferences about individual users, who might not want to use such systems if they do not know who has access to the internal user model. For this reason, the model should be kept confidential and secure [Pratt 87].

It is unclear whether the above usability issues can ever be completely solved by better designed software. Many issues of motivation and timing require far too much real-world knowledge at a level not yet demonstrated by any system. However, it is important to be aware that these issues play a major role in the success of future active assistance systems and can be factored into their design at a broader level. For example, such a system could include the management of the organization using the system in the loop of providing extra resources or motivation, or even allow management to modify high-level system controls. This possibility was strongly suggested by the user study mentioned above.

Whatever form the final system designs take, they must necessarily emanate from a better understanding of usability issues like those discussed. These issues are concerned not only with performance measures that can be developed in laboratory settings, but also with issues of management, work style, and group interaction, which are best observed in the real-world. We suggest in the next section that anthropology offers a way to understand usability issues in real-world settings.

Addressing Real-World Usability Issues

Although there have been many advances in techniques to predict and study human-computer interaction, these techniques have been based on modeling cognitive processes derived from laboratory experiments. There is a growing awareness that such studies do not capture many real-world problems [Suchman 87] such as motivation and privacy as discussed earlier. Techniques from ethnography (a branch of cultural anthropology) have been used in field studies to give clues to computer usability problems. This approach uses informal, qualitative techniques to provide feedback to designers [Forsythe 93].

The technique used by ethnographers is commonly known as *participant observation* and involves a field worker who immerses him- or herself in the real-world context of the users. The field worker, by participating in the user's context, begins to bring meaning to the observations he or she makes. This allows the field worker to understand users from their own perspective, including their interactions with others in the group, their values, beliefs, and assumptions.

The field worker often builds a rapport with the users while attempting to make unobtrusive observations. This is complemented by open-ended interviews that help to explain or validate the observations. More formal methods of recording can also be used. The technique is deliberately designed to be informal and qualitative so that the research question that a field worker starts with can be easily modified to account for unexpected results. This flexibility helps to reject preconceived notions of user needs and to develop a sense of what the real issues are.

Although participant-observation can be difficult for observers who have not been trained in studying people in natural work environments, a trained ethnographer can obtain crucial insights into the design of a system. For example, an ethnographer can study how computer and design knowledge flows in a design office and factor it into the design of active assistance systems. By incorporating this increased understanding of usability issues before and during design and implementation, active assistance systems have a better chance of increasing productivity and quality of design.

The vision of developing active assistance systems based on a deeper understanding of usability issues, derived from a combination of traditional cognitive techniques and ethnographic studies, remains, of course, to be seen. Although at the present time it is not more than a hypothesis, it presents exciting possibilities that make it worth pursuing through a concentrated effort.

Conclusion

Traditional forms of user assistance like training, on-line help, and documentation are insufficient to improve the productivity of the users of CAE systems. Active assistance systems show promise in overcoming some of these deficiencies. Usability issues such as motivation, timing, credibility and privacy must be given careful attention in designing and developing active help systems. Empirical studies in the real-world settings are needed to confirm the significance of these issues and to establish promising ways of dealing with them. The underlying paradigm of continuously monitoring, modeling, and actively responding to the actions of users during their work with a CAE system can turn the computer into a true design partner that affects productivity as well as the quality of the designed product.

Bibliography

[Coventry 98] Coventry, L., “Some Effects of Cognitive Style on Learning Unix,” *International Journal of Man-Machine Studies*, 31, pp. 349–365, 1989.

[Carroll 86] Carroll, J.M., Smith-Kerker, P.L., Ford, J.R., and Mazur, S.A., *The Minimal Manual*, IBM Research Report, RC 11637, 1986.

[Carroll 87a] Carroll, J.M., and McKendree, J., “Interface Design Issues for Advice-giving Expert Systems,” *Communications of the ACM*, 30, pp. 14–31, 1987.

[Carroll 87b] Carroll, J.M., and Rosson, M.B., “The Paradox of the Active User,” in *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, J.M. Carroll, Ed., Bradford Books/MIT Press, Cambridge, MA, 1987.

[Carroll 88] Carroll, J.M., Aaronson, A.P., “Learning by Doing with Simulated Intelligent Help,” *Communications of the ACM*, 31, pp. 1064–1079, 1988.

[Drisis 92] Drisis, L., “A CAD-User Survey: Experience and Results,” Proceedings, *Third International Conference on Human Aspects of Advanced Manufacturing and Hybrid Automation*, pp. 26–28, 1992.

[Flemming 93] Flemming, U., Coyne, R., Woodbury, R., “SEED: A Software Environment to Support the Early Phases in Building Design,” in *ARECDAO93*, Proceedings, IV International Conference on Computer-Aided Design in Architecture and Civil Engineering, Barcelona, Spain, pp. 111–122, 1993.

[Forsythe 93] Forsythe, D.E., “Using Ethnography in the Design of an Explanation System,” submitted to *Expert Systems with Applications*, 1993.

[Greenberg 85] Greenberg, S., and Witten, I.H., “Adaptive Personalized Interfaces—A Question of Viability,” *Behavior and Information Technology*, 4, 1, pp. 31–45, 1985.

[Hanson 84] Hanson, S.J., Kraut, R.E., and Farber, J.M., “Interface Design and Multivariant Analysis of UNIX Command Use,” *ACM Transactions of Office Information Systems*, 2, 1, 1984.

[Hiltz 84] Hiltz, S.R., *Online Communities: A Case Study of the Office of the Future*, Norwood, NJ., 1984.

[Houghton 84] Houghton R.C., Jr., “Online Help Systems: A Conspectus,” *Communications of the ACM*, 27, 2, pp. 126–133, 1984.

[Jones 91] Jones, J., and Virvou, M., “User Modelling and Advice Giving in Intelligent Help Systems for Unix,” *Information and Software Technology*, 33, 2, pp. 121–133, 1991.

[Kolodner 91] Kolodner, J., “Improving Human Decision Making through Case-Based Decision Aiding,” *AI Magazine*, 12, 2, pp. 52–68, 1991.

[Pratt 87] Pratt, J.M., “The Social Impact of User Model,” Interact ’87, Proceedings, *Second IFIP conference on Human-Computer Interaction*, Stuttgart, Part: 1–4, September 1987.

[Rosson 84] Rosson, M.B., “Effects of Experience on Learning, Using, and Evaluating a Text-Editor,” *Human Factors*, 26, pp. 463–475, 1984.

[Suchman 87] Suchman, L.A., *Plans and Situated Actions: The Problem of Human Machine Communication*, Cambridge University Press, 1987.

[Yang 92] Yang, D., Garrett, J. H., and Shaw, D.S., “Cell Management Using Neural Network Approaches,” Proceedings, *Government Neural Network Applications Workshops 1*, pp. 19–23, 1992.