Design for Collaborative Learnability

R. T. Jim Eales[1] and Jim Welsh[2]

[1]*Department of Computer Science, University of Queensland and CRC for Distributed Systems Technology*
[2] *Department of Computer Science, University of Queensland*

Abstract

This paper considers computer-based support for the development of computer skills in the workplace. We suggest that computer systems should be designed to support collaborative learnability; to this end, we offer a number of collaborative learnability design principles. In particular, we emphasize that the prime objective should be user participation. We suggest some principles of collaborative visibility and highlight the importance of the demonstration in the sharing of skills. The various design principles are incorporated into a generic model for collaborative user support called MutualAid. A specific system based on this model is also described. This system uses multimedia demonstrations recorded by end users to support an interactive problem-solving forum and the development of a local database of computer-related practice.

Keywords — Human-Computer Interaction (HCI), user support, learnability, collaborative support systems, collaborative learnability, collaborative visibility, multimedia demonstrations.

1. Introduction

This paper considers computer-based support for the development of *computer skills*[1] in the workplace. The focus on computer skills reflects both our background in Human-Computer Interaction (HCI) and our practical experience in helping people to develop these skills. Although some of our design ideas are particular to the development of computer skills, we see the fundamental basis of our approach as relevant to many other areas of learning in the workplace.

Our approach is based on a simple premise - the type of support of greatest value to computer users is knowledgeable assistance from their colleagues. We contend that user support should be designed to take advantage and to improve the quality of this mutual assistance. We have no hesitation in defining our work as *computer support for collaborative learning*. However, the fact that this learning occurs in the workplace, and not in an educational institution, has a profound influence on the nature of the collaborative learning and on the form of the computer support. As Koschmann et al. [11] remind us "... the presence of ill-structured problems in ill-structured domains is probably typical of most substantial attempts to use knowledge effectively in the real world. Well-structured knowledge domains and well-structured problems are almost exclusively the property of schooling" (p. 232).

Our research focuses on the *learnability* of computer systems or computer-based tools. We define learnability as *the properties of a (computer) system that aid in the learning of that system's use*. Any definition of learnability, however, depends ultimately on a definition of learning. We use the term *collaborative learnability* to emphasize a view of learnability based on a collaborative notion of learning. We are particularly interested in developing a number of generic design principles encompassing the essence of collaborative learnability. The various principles are incorporated into a generic model called *MutualAid*. We further illustrate our approach by describing a specific instance - MutualAid1 - of a *collaborative support system* developed from this basic model.

2. Information Technology in the Workplace

Information technology (IT) is playing an increasingly important role in most workplaces. In many cases an organization's productivity and capacity to innovate is closely linked to the IT proficiency of its workers, now transformed into end users. Although the typical computer system, such as a PC running a word-processing or spreadsheet package has become a rather mundane organizational artefact, it is still a powerful, complex

[1] We use the term computer skills to describe the expertise associated with the use of a wide range of software applications such as word processors and spreadsheets.

and multi-functional tool. Not surprisingly there are a growing number of reports of end users utilizing computer-based systems in restricted and simplistic ways (for example [4],[19]). We believe that these reports represent only the tip of the iceberg, and highlight a significant problem. We suggest that if we could improve the longitudinal learnability (support for learning over time) of computer-based tool systems this would contribute to better understanding amongst end users and hopefully lead to significant organizational productivity gains. We are aware, however, that there is more to expert practice than tool skills and that the design of the tool is only one factor in the many that can affect actual tool use.

3. The Theoretical Approach

In order to make anything more learnable we believe that one needs at least a working understanding of what it means to learn. Generally, support for the development of computer skills has been monopolized by formal instructional approaches, typically short training courses and self-instruction by way of printed documentation or on-line tutorials and help. The end user support task is normally interpreted as *the transfer of essential objective knowledge into the heads of individual users.*

While all forms of user support can play a part in the development of computer skills, we believe there are whole areas of understanding beyond the scope of formal instruction. These areas can be described as highly localized and situated ways of knowing organized around structures of relevance [18]. Designers of instructional courses and materials do not and cannot know these areas. Understanding completely the functionality of a tool (in the way that developers do) is nothing like understanding the situated application of that tool in workplace activities. Fortunately, in their efforts to come to terms with these areas of understanding the worker is rarely alone, he or she shares many of the situational factors with fellow workers. We see this as the basis for collaborative learning in the workplace. There is considerable evidence to support the proposition that *asking a colleague* is the most common and most valued form of support amongst computer users [6,7,19].

Although, some have speculated on the use of technology to support such mutual assistance (for example,[2]), generally, this approach has not received a great amount of serious research attention. Perhaps the prevailing belief amongst many is that this approach would be a case of "the blind leading the blind" or "computer-supported misinformation". A related area of research investigates organizational memory or know-how systems such as Answer Garden [1] and FISH [20]. However, these systems are generally text-based, organization-wide, involve universal structuring of information and do not make the collaboration process particularly visible.

We have been particularly influenced by ideas on *situated learning* [3] or *situated practice* [12]. We see the "situated argument" as an important contribution to the general process of recognition of mutual assistance as a valuable method of user support. Lave and Wenger [13] have suggested that the consideration of learning as "legitimate peripheral participation in communities of practice" can be a valuable analytical perspective. Most of the examples used by Lave and Wenger to illustrate their perspective are traditional systems operating in more or less stable environments. In many workplaces, however, mastery is in short supply and what is required is a kind of collaborative bootstrapping of expertise.

We use a number of basic working principles derived from theories of collaborative learning and situated activity:

- Learning in the workplace is primarily motivated by the everyday dilemmas and needs involved in work activities. Given the right conditions and the right support, these dilemmas can be turned into learning opportunities.
- What is of greatest value to the computer user is not "universal" instruction but access to and participation in knowledgeable situated practice.
- The fundamental operational unit of support is the small group. The mutual understanding and mutual commitment that can develop within these groups is a valuable resource in learning.
- Even with full collaboration local expertise may be limited. A group should always have ways of extending or improving its collective understanding by learning from the practice of others.

Our fundamental argument is that technological systems can and should be designed to support and to utlilize the potential for collaborative support.

4. The Social and Technical Basis of Support

Our approach is based on the notion of a sociotechnical system, in that it consists of a social sub-system and a technical sub-system. Both sub-systems are vital to the overall effectiveness of the system. As the sociotechnical movement found [15], if the technical system is optimized at the expense of the social sub-system the results obtained will be sub-optimal.

The fundamental social basis of support in our approach is the *support group* - a small, "closed" and hopefully cohesive group of ordinary computer users. This group should be engaged in similar or related work tasks involving the use of the same computer-based tools. Normally, this support group would be based on an existing work group. Within the support group an important role is that of the *support person* or local expert. The support person requires a combination of technical and social skills. They have to know in detail the tasks of the group, they need a knowledge of, or at least an interest in, computer systems, and they have to have a genuine commitment to assisting other members of the group in their various needs. Generally, a support person "emerges" from the group and is then sometimes recognized semi-formally by the organization. The importance of social factors in this kind of support system means that the success and effectiveness of the technical sub-system is crucially dependent on factors that cannot be directly addressed by technology. For example, the cohesiveness of the support group, the quality of the support person or the general organizational climate for learning can all have a significant influence on the success or failure of a collaborative support system.

Initially, we are interested in developing auxiliary systems that seek to facilitate collaborative support around *existing computer tools*. We believe, however, that ultimately collaborative support may be best served by integrating characteristics and functions that support collaboration within the design of the tools themselves. The growing number of computers connected to computer and telecommunication networks is providing opportunities for new and hopefully more successful approaches to user support. At present, most organizational end users can only be described as partially distributed. They still have the option of asking a question of a person sitting near them or of visiting someone nearby. In the future many workers may be part of a fully distributed organization where opportunities for face-to-face contact are extremely limited (for an illustration see Robertson [17]). In the partially distributed organization we believe that a collaborative support system can make a useful contribution to existing methods of formal and informal support. In a fully distributed organization such a system may be the only way to provide certain kinds of essential support.

5. Design for Collaborative Learnability

> *Jim: One section has produced a local procedures manual.*
>
> *Kylie: I don't see why we should have to write the manual for this section. We could do it, but we don't have the time to sit down and write all these things.*
>
> *Jim: What if someone comes to you with a problem?*
>
> *Kylie: But that's different, then they're asking you something and you're showing them. But to write it down - you don't know what they are going to ask. You would have to write down everything.*

The above section of a transcribed interview comes from a study we undertook into organizational end users and their methods of skill development [5]. We believe this very short section illustrates a number of crucial factors in the design of computer support for collaborative learning in the workplace. The comment "we don't have the time" highlights the principal learning constraint of the workplace - the amount of time and effort a worker is prepared do devote to learning. In collaborative user support we also have to consider how much time and effort a worker is prepared to actively devote to the learning of others. If we want people to use any kind of user support we have to minimize the amount of time and effort demanded of the user while maximizing the quality of the support provided. When Kylie says "I don't see why we should have to write the manual" we suggest that this is a reaction against the idea of manuals in general as well as the tedious chore of writing down all the relevant information. In contrast, the reaction towards helping someone is positive, because they have specific and personalized requests for assistance. It should also be noted that whereas Kylie uses the phrase "*writing* a manual" she refers to "*showing* them" when offering assistance. We feel this is an important distinction which we will touch on again later in the paper when considering the value of demonstration.

In summary, our aim is not to focus on the development of a database or manual of local practice, although such a database may be produced as a by-product of technologically-mediated support. The primary function of a collaborative support system is to provide technological support for local collaborative assistance or mutual aid. In the application of technology to this area we are not trying to undermine existing face-to-face support. We believe technology can make an important contribution to the collaborative learning process. At the heart of the process, however, it is still people helping people.

5.1. The Prime Objective is Participation

User support is essentially a discretionary function. In order to make an effective contribution to end user skill development a support system has to be more than just useful and usable - above all *it has to be used*. Our prime objective should always be to encourage use of the system and participation in its continuing development. To this end we have to be aware of the subtle constraints and inducements acting on the end user in the workplace. We do not think it is sufficient just to demonstrate that one type of user support is better than another by some form of contrived testing in an artificial environment. The real issue is not whether one example of user support is better than another but whether users will freely choose to use any example when not constrained by the test situation.

We want to extend the technology of the workplace to utilize and develop collaborative support. The driving force in the use and development of such a system is cooperation. Although cooperation is ubiquitous and intrinsically rewarding it is also a very fragile phenomenon. In the development of computer-mediated collaborative support systems what appears to be required is sensitive design and restrained use of technology. We suggest that a local and personalized focus on collaborative problem-solving appears to be a promising starting point for a collaborative support system. To facilitate this we need a method of local interaction between end users that provides a rich demonstration of possible problem solutions while requiring the minimum of end user time and effort.

5.2. Principles of collaborative visibility

In the study of Computer Supported Cooperative Work (CSCW), with its focus on the social processes involved in group interaction, there is a growing awareness of the significance of the "invisible" aspects of joint activities (see for example [8]). These invisible aspects are often vitally important habitual practices and other tacit understandings that people take for granted and are rarely aware of. Hutchins [9] has drawn attention to what he terms *open tools,* such as navigation charts. He suggests that the design of tools can affect their suitability for *joint use* or for *demonstration*. When a person is performing some activity the interaction between that person and a tool may or may not be open to others depending on the nature of the tool. Open tools provide opportunities for observation of tool use and may contribute to the general spread and development of more expert practice.

It has to be said, however, that desktop computers are not inherently open tools but instead are private tools - it is not easy to observe the interaction of a user and a computer other than in a very trivial way. In the workplace we start from a position where computer skills are largely invisible. Our goal is to make computer systems into open tools. To this end we have attempted to identify a number of generic *collaborative visibility* design principles. Collaborative visibility involves the revealing of both activity and the collaborative context:

(a) Make tool use visible

Making tool use visible to others is the fundamental objective in our quest for collaborative learnability. However, the idea of visibility goes far beyond the simple notion of being able to physically observe something. We believe that one of the most important methods of making tool skills visible is demonstration.

(b) Make the end-products of tool use visible.

Under certain circumstances the end products of tool use can be a quick and useful summary of what it is possible to do, in a given situation. They provide a kind of indirect visibility of activities.

(c) Support interactive discourse about tool use

Participation in the practice of a group is more than just the passive observation of someone doing something. Making something visible also involves revealing the meaning.

(d) Allow control of visibility

Control over the revelation of practice by the practitioner creates the opportunity for conditions of confidence and trust to develop and for practice to be revealed in meaningful ways.

(e) Allow capture and storage of examples of tool use

One of the inherent advantages of technological communication is that it creates the opportunity for the capture of interaction, thus extending visibility by making it possible to relay this information to other places or other times.

6. The Importance of Demonstration

Ask a colleague how to do something using a computer and they will invariably demonstrate it to you. You are invited into a position where the events occurring on the screen can be clearly seen, or the demonstrator takes over the controls of your computer. The demonstrator then goes through the relevant sequence of interactions between the user and machine. The significant events are emphasized usually by verbal commentary and you are also able to ask questions if a point is missed or needs clarifying. The demonstration, of course, has a long history. For example, it was an essential part of traditional apprenticeships. Recently there has been a certain amount of research interest in animated demonstrations as a method of instruction for computer skills (for example [10,16]). Some software packages now include animated demonstrations in guides to the system or as a part of on-line help.

The face to face demonstration can be a very important method of *communication between users* about the way to do things with tools. Demonstrations provide a special kind of visibility. They are not just a slice of everyday activity made visible, they have specific characteristics that make them useful for learning. A demonstration is *focussed* on a specific area, it is normally the answer to a query but could be an illustration of a problem or something similar. It is not just a randomly selected sequence of normal activities. A demonstration normally takes place under time constraints, therefore it tends to feature only events that are significant. It is a *condensed* version of significant events. The same sequence of events when performed during normal working conditions may have long time intervals between them. A demonstration is usually *interactive*. This interaction between the demonstrator and the learner is often very important. Interaction about the problem or the solution may change the whole direction of the demonstration. An active demonstration is also very useful for the demonstrator because it *aids recall* of the interaction between the user and the machine. Even experienced computer users find it hard to recall the sequences of interactions that occur between themselves and an application, when that particular application is not in front of them [14]. A demonstration normally also has a verbal *commentary* that serves to highlight significant events. A running commentary is not normally a feature of the average user's work activities.

Demonstrations can take many forms. We have concentrated on the face to face demonstration as the ideal model, but it has the significant disadvantage that this is usually only between two people and after the actual event the demonstration is lost. Technologically mediated demonstrations can be either synchronous, allowing real-time interaction, or asynchronous where a demonstration is recorded for later viewing. We can also envisage some kind of virtual reality demonstration taking place in cyberspace. At the other extreme most printed software tutorials incorporate static text and graphics demonstrations of how to interact with the system. We believe demonstrations are more effective in a situation where there is some measure of mutual understanding and mutual commitment along with shared or similar problems. This is the situation we have attempted to address with the MutualAid model.

7. Outline of the MutualAid Model

We use the term *collaborative support system* as a generic name to cover a whole range of possible technological systems designed to incorporate collaborative learnability. Our *MutualAid* model provides a generic basis for the development of specific systems utilizing different forms of technology. MutualAid1 (MA1) is one such system. It is a group-based asynchronous information sharing system focussed mainly on developing computer tool skills. This particular system utilizes recorded demonstrations as the principal means of communication and a collaborative forum and database to support discussion and subsequent storage of local knowledge. We emphasize that the MutualAid model is deliberately *minimal* and the MA1 system is a minimal prototype. There are a number of reasons for this minimalist design approach:

- The generic MutualAid model can be the basis for various systems utilizing a wide variety of representation, communication and storage forms. For example, we are currently investigating the feasibility of utilizing electronic mail, the World Wide Web and various groupware products as the basis for, or a part of, a MutualAid system.
- We feel that collaborative support systems ought to be developed collaboratively, if possible. We therefore consider our system as a minimal starting point for evolutionary development by participatory design contributions and contextual evaluation.
- Even when the system design has been developed and stabilized we hope that the form will still be considered as minimal. If users are having problems mastering computer-based tools the last thing needed is more technological complexities "thrown at them".

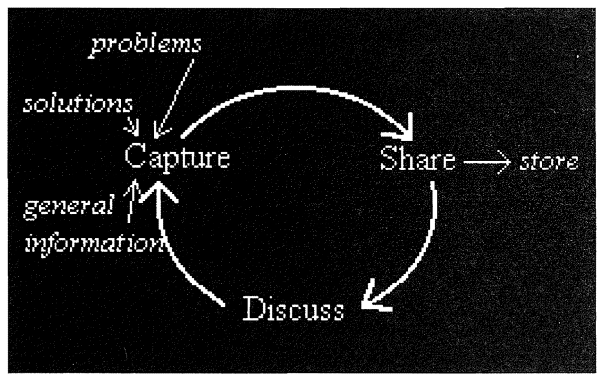

Figure 1. The basic MutualAid model.

7.1. The Conceptual Model

The basic conceptual MutualAid model is represented in (figure 1). It is as a cyclic process of capture, sharing and discussion. Problems, solutions, related experience and the discussion process itself are captured and made visible within a support group. The various contributions are stored for reference purposes. The driving force of the system is the collective resolution of everyday dilemmas (or innovative opportunities) associated with the use of complex tools in variable work activities, where time and quality are critical elements.

7.2. The Functional Model

The MutualAid1 system is a particular instance of a system based on the MutualAid generic model. Although this instance may be a valuable way to illustrate the general model we can derive many other specific instances from the basic model. The MutualAid1 system can be considered as three functional elements, the multimedia recorder tool, the problem-solving forum and a local database. The multimedia recorder tool is the principal method of capture in the system, although textual, graphical and other forms of representation can also be used. In order to facilitate the sharing of problems, solutions and general information and supporting the subsequent interactive discussion; we need some method of making these issues more visible or public within the support group. This is the function of the problem-solving forum. As a by-product of these processes a local database of practice is built up.

7.2.1. The multimedia recorder tool

Although it is possible to create textual and graphical descriptions of problems and solutions using the system, by far the most important method of communication is the recorded animated demonstration. There has been little if any published research into animated demonstrations *created by end users*. The recorder tool[2] allows the user to record an animated demonstration of screen events with a coordinated verbal commentary. The recording can be based on an individual demonstrating some aspect of tool use or it can be a recording of an interactive demonstration, that is, what we have termed a face to face demonstration between two people, with both sets of comments recorded. The interface to this tool is very simple, it looks and works in a similar way to the controls of a cassette recorder. The user selects record, demonstrates the relevant sequence of screen actions, while optionally making comments, and then selects stop. This creates a file, which can be played back to check the demonstration. The recording can then either be saved or wiped. Recorded demonstrations can be activated by loading them into the recorder or player and playing them. A simpler method of activating them is to embed them in a document as an icon using Microsoft Windows Object Linking and Embedding (OLE™). Recorded demonstrations can be embedded in a variety of document types. Documents with embedded demonstrations can be submitted to the forum or demonstrations can be added to existing documents already in the forum or database.

It should be emphasized that the users of this system are expected to be relatively inexperienced end users, we therefore want a method of operation that invites participation rather than adding to their learning burden. Perhaps the principal advantage of the recorded demonstration is that it can convey a great deal of information and yet is both easy and quick to create. We have found that a 30 second long recorded demonstration can convey a considerable amount of information and takes only a short time to record and embed in a document.

7.2.2. The problem-solving forum

The term forum (i.e., a place of public discussion) reflects both collective visibility and interactivity. The forum is the place where interaction, which is a vital part of the sharing and development of practice, takes place. It is also the way in which the local database grows and develops. Problems, solutions, experiences and general comments are made public within the group by inserting them as documents in the forum space, inviting responses from other members of the group. Responses to issues in the forum may take the form of new documents or may be additions to the original document. Recorded demonstrations can be added to existing documents very simply. To support the forum in Microsoft Windows we use a program group as a form of noticeboard on which documents can be posted. When discussion on an area in the forum has reached a reasonably stable state, the relevant responses are transferred (by the support person) to the permanent local database.

7.2.3. The local database

The local database is a collaboratively created collection of documents containing embedded multimedia demonstrations and other related pieces of information. Because it is locally created it can contain local methods of manipulating local information. For example, a part of the database devoted to using a spreadsheet can incorporate demonstrations of actions on the actual spreadsheet models used in the workplace. Initially, of course, the database is empty. It may be necessary for the support person (or the group as a whole) to submit a range of documents with embedded demonstrations covering the basic skills used by the group in order to achieve some initial "critical mass". The principal value of the database is that it provides a permanent record of local solutions to local problems which can be browsed or accessed quickly by members of the group. The demonstrations should be easy to access as a reference when someone needs to be reminded of the way to do something. Consequently speed and ease of access are important structural issues in the organization of the database. The central question in the organization of this local database is not how *we* should structure it, but what methods of structuring do we need to provide for the users.

We are currently testing this system in the introduction of spreadsheet software into an administrative department and also in a limited form in a course on computer skills for undergraduates.

8. Further Research

Just as a computer user may be isolated from the practice of others, a particular group of users may be isolated from the practice of other groups. There are obvious advantages in sharing solutions to common problems on a wider scale than the local support group.

[2] We saw the value of such a recorder tool, we then discovered Lotus ScreenCam™, an inexpensive multimedia screen and sound capture utility for Microsoft Windows™.

However, various problems can be expected to arise when we attempt to communicate practice across group boundaries. Our intuitive approach to this problem has been to suggest a support persons' support group - a group where expertise can be developed at a more specialized level. The learning robustness of the support person or gatekeeper may be able to compensate for the initial lack of cohesion and understanding in the group. This intuitive approach requires further investigation.

The MutualAid system has been designed as an asynchronous system. This has technical and operational advantages, but there may be times when synchronous communication is essential. For example, problems may have to be resolved urgently or certain kinds of practice may only be able to be communicated interactively. We need to investigate the integration of a synchronous communication facility, paying close attention to the subtle constraints and inducements acting on its use.

9. Conclusions

This paper has discussed a collaborative approach to the problem of computer skill development in the workplace. We have presented a theoretical direction - design for collaborative learnability - and have illustrated this approach by outlining our MutualAid model and MA1 system. We have emphasized the central importance of demonstration and have utilized multimedia demonstrations as the principal form of representation in the system. We believe an approach based on principles of collaborative learnability and taking into account the potential of technologies such as computer-mediated communication and multimedia can transform individual computer-use dilemmas into collaborative learning opportunities.

Acknowledgments

The work reported in this paper has been funded in part by the Cooperative Research Centres Program through the Department of the Prime Minister and Cabinet of Australia.

References

1. M. S. Ackerman. Augmenting the organizational memory: A field study of Answer Garden. In the proceedings of CSCW'94, pages 243-252, 1994.

2. L. J. Bannon. Helping users help each other. In D. A. Norman and S. W. Draper, editors, User Centered System Design, pages 399-410. Lawrence Erlbaum, Hillsdale, NJ, 1986.

3. J. Seely Brown, A. Collins, and P. Duguid. Situated cognition and the culture of learning. Educational Researcher, 18(1):32-42, 1989.

4. C. V. Bullen and J. L. Bennett. Groupware in practice: An interpretation of work experiences. In R. Kling and C. Dunlop, editors, Computerization and Controversy, pages 257-287. Academic Press, San Diego, CA, 1991.

5. R. T. J. Eales and J. Welsh. Learnability through working together. In the proceedings of OZCHI'94, (Melbourne, 28th November - 2nd December 1994) , pages 27-32, 1994.

6. M. Frese, F. C. Brodbeck, D. Zapf, and J. Prumper. The effects of task structure and social support on user's errors and error handling. In D. Daiper et al., eds, The proc. of INTERACT'90 , pages 35-41, 1990.

7. R. E. Granda, R. Halstead-Nussloch, and J. M. Winters. The perceived usefulness of computer information sources: A field study. SIGCHI Bulletin, 21(4):35-43, 1990.

8. J. A. Hughes, I. Sommerville, R. Bailey, and D. Randall. Designing with ethnography: making work visible. Interacting with Computers, 5(2):239-253, 1993.

9. E. Hutchins. The technology of team navigation. In J. Galegher, et al., editors, Intellectual Teamwork: Social and Technological Foundations of Cooperative Work , pages 191-220. Erlbaum, Hillsdale, NJ, 1990.

10. M. P. Kerr and S. J. Payne. Learning to use a spreadsheet by doing and by watching. Interacting with Computers , 6(1):3-22, 1994.

11. T. D. Koschmann, A. C. Myers, P. J. Feltovich, and H. S. Barrows. Using technology to assist in realizing effective learning and instruction: A principled approach to the use of computers in collaborative learning. The Journal of Learning Sciences, 3(3):227-264, 1994.

12. J. Lave. The practice of learning. In S. Chaiklin and J. Lave, editors, Understanding Practice: Perspectives on activity and context, pages 3-32. Cambridge U. P., Cambridge, UK, 1993.

13. J. Lave and E. Wenger. Situated Learning: Legitimate Peripheral Participation. Cambridge U. P., Cambridge, UK, 1991.

14. J. T. Mayes, S. W. Draper, A. M. McGregor, and K. Oatley. Information flow in a user interface: the effect of experience and context on the recall of MacWrite screens. In D. M. Jones and R. Winder, eds, People and Computers IV - The

proceedings of BCS-HCISIG 1988, pages 275-289, 1988.

15. E. Mumford. Sociotechnical systems design: Evolving theory and practice. In G. Bjerknes, P. Ehn, and M. Kyng, editors, Computers and Democracy: A Scandinavian Challenge, pages 59-76. Avebury, Aldershot, UK, 1987.

16. S. Palmiter and J. Elkerton. Animated demonstrations for learning procedural computer-based tasks. Human-Computer Interaction, 8:193-216, 1993.

17. T. Robertson. "We can do it better": Communication and control of work practices. In the proc. of OZCHI'94, pages 295-300, 1994.

18. A. Schutz and T. Luckmann. The Structures of the Life-World. Northwestern University Press, Evanston, IL, 1973.

19. Y. Waern, N. Malmsten, L. Oestreicher, A. Hjalmarsson, and A. Gidlof-Gunnarsson. Office automation and users' need for support. Behaviour and I.T., 10(6):501-514, 1991.

20. T. Yamakami and Y. Seki. Knowledge awareness in asynchronous information sharing. In D. R. Vogel et al., editors, Local Area Network Applications: Leveraging the LAN (Proc. of the IFIP TC8/WG8.4 Working Conference), pages 215-225. North-Holland, 1993.

Authors' Addresses

R. T. Jim Eales and Jim Welsh: Department of Computer Science, University of Queensland, Brisbane, 4072, Australia. {eales, jim}@cs.uq.oz.au.