Design to Instruct: Lessons for Training through Involving Teachers in Design.

Jane Hughes & M. Angela Sasse

Abstract

Multimedia and communications technology has great potential for education, provided that teachers are given training that enables them to make full use of it. Access to training is often a problem and, with very new technologies, there is a shortage of trainers who understand both the technology and the needs of teachers and learners. This paper reports on a project to involve users in the design and development of a real-time multimedia conferencing system for language tutoring and explains how it has yielded information about the kind of training needed to use such systems effectively. It has also generated training materials and shown that teachers can shape the technology and gain the confidence to transfer their expertise to others.

Introduction

The meeting of multimedia and telecommunications promises an explosion of new computer-based applications for education. Only a few years ago it was assumed that packet-switched networks, such as the Internet, could not be used to transmit real-time audio and video because of the delays inherent in shared networks. However, improved compression techniques and the advent of multicast [1] in the 1990s have changed this. It is now possible to take part in real-time, multi-party videoconferences, from a desktop computer. Educational institutions at all levels are connecting to computer networks and the potential to enrich student learning is huge. Yet, whilst university research staff hold “live” project meetings with partners all over the world, or watch lectures transmitted from conferences they cannot attend, teaching staff are only beginning to explore the possibilities of this new technology.

Barriers to the acceptance and use of computer-based technology in schools and colleges are well documented. There may well be resistance to change amongst both management and teachers but underlying this are a number of other factors. Williams (1994) describes vividly the pressures on teachers and institutions which limit access to training. Bates (1995), however, suggests that the greatest obstacle is the unfamiliarity of the technology, which erodes teachers’ confidence. He too points to inadequate training as the key underlying cause. He states that the problem is not simply ignorance about how to use a piece of technology but, more importantly, the “lack of an appropriate conceptual framework to guide the use . .”. This problem is particularly acute in the Higher Education sector, where few of those who teach have a foundation in educational theory or instructional design. He warns that, without such a framework, people will fall back on the methods that were used to teach them – a strategy wholly inadequate where social and technological change are rapid.

To make effective use of information technology, teachers must learn more than just how to make the system function; they need to use it to communicate with their students and to deliver appropriate and effective courses over a period of time. Whatever conflicting theories exist about teacher training, it is generally accepted that those who deliver it will be able to draw upon extensive practical experience within a rigorous theoretical framework. It would therefore seem reasonable that teachers should be introduced to new technology by experts from their own

[1] An efficient way of routing traffic over networks, that makes multi-way conferences possible at reasonable cost and without causing undue network congestion. See Macedonia &Brutzman (1994).

domain, rather than by experts from the technology domain who might be seen to have different objectives and not to understand what teachers and students need. At present, however, not enough teachers have the relevant technical knowledge and even fewer combine it with appropriate practical experience. In the case of very new technologies the shortage is even more acute and even the content of a training programme may be hard to determine.

The work presented arises from research into the design of real-time multimedia systems for education. Approaches derived from the participatory design movement (see next section) have enabled teachers to play an important part in developing a language tutoring system. The paper discusses the techniques used and considers the implications of the work for staff development and training. The aim was to make the technology acceptable to teachers and to empower them to shape it to their needs.

The technology and the teaching

ReLaTe[2] is a real-time system for small-group distance language learning, incorporating audio, video and a shared workspace. It uses ordinary computer workstations and runs over the multicast backbone of the Internet. Separate tools for each medium (audio, video and shared workspace) are combined into a single window (see Fig. 1). This interface was designed for small group tutoring but it could be configured for larger groups.

On the left is the video area, where four images can be displayed with a choice of size. Participants select the image to be displayed in the larger rectangle by clicking on the name bar above it. The shared whiteboard on the right allows displayed text and graphics to be seen and annotated by all, and all may write or draw on it simultaneously. Users can replace this with a shared text editor by pressing a button. The audio controls allow users to adjust incoming and outgoing volume.

[2] ReLaTe (Remote Language Teaching over SuperJANET) is a joint project between University College London and the University of Exeter in the UK. It has been funded by British Telecom and the UK Joint Information Systems Committee (JISC). The project is a partnership between computer scientists and teachers, to investigate the feasibility and implications of teaching foreign languages over computer networks. See Hughes & Sasse (1997).

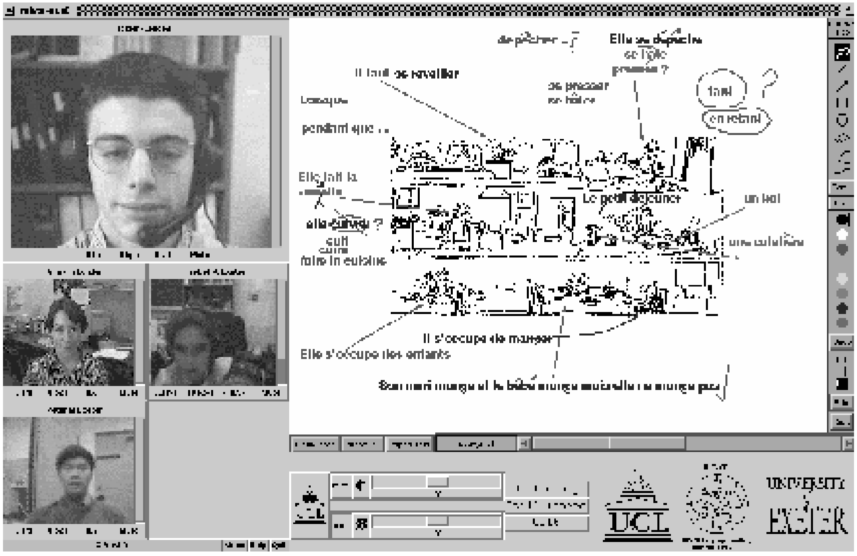

Figure 1: The ReLaTe System in action.

The prototype system has been used in 10 and 22-week field trials, with language learners at levels from beginner to advanced. Five courses were run, in three different languages. In all of them, the tutor and students were geographically dispersed, sitting at individual computers at various locations in London and Exeter, connected via the UK academic network. Five tutors and twelve students participated.

Why use a Participatory Design Approach?

Participatory design (PD) originated in Scandinavia as part of the workplace democracy movement (Clement & Van den Besselaar (1993)) but the term now covers a continuum of user-centred design approaches. Carmel et al (1993) give an indication of the range. Participatory design was used here for a number of reasons. First, Grudin (1991) has stated that a participatory approach is best suited to emerging technologies. Secondly, participatory methods have traditionally been used for custom-built systems and this is exactly what was required in this instance. There was no existing blueprint and the application being designed was very specific. A PD approach also suited the work situation because teaching and learning can not be described adequately in terms of data flow and task analysis, favourite designers' tools. Another factor was the participatory design view that work practice changes should run alongside the development of computer systems. This fitted the philosophy of the project. PD also tackles cross-domain communication problems, certainly relevant in this project, where computer scientists and language teachers would be working together. The PD view is that development is co-operative, learning a two-way process, rather than one where "expert" technologists transfer information to "inexpert" users. Finally, PD evolved to combat the idea that technology was a threat to the worker, instead regarding its introduction as a way to improve the quality of people's working lives.

PD Principles and Practices

Participatory design does not follow a rigid formula. Instead, researchers have developed a repertoire of tools and techniques to promote equal participation of users in design. Whatever techniques are used, a few broad principles underpin most work. Apart from those mentioned above, practitioners believe that concrete experience is needed to inform abstractions (so developers must gain workplace experience and users experience with the proposed system) and that requirements will not be understood fully until users have worked with a system for an extended period.

A taxonomy of PD techniques is presented by Muller et al (1993). Wall & Mosher (1994) and Kensing & Munk-Madsen (1993) also provide reviews of methods. Those most commonly used are unstructured interviews, formal and informal group discussions, future workshops, graphical techniques to aid both visualisation of the future system and representation of the workplace, and prototyping. Prototyping is a particularly important tool and includes everything from storyboards, through Hypercard-based simulations to working systems. Often low-tech artifacts, such as plastic tokens or post-it notes (Muller (1991)) are used, in order to diminish the intimidating effects of unfamiliar technology. Observation of the workplace and of prototype systems in use is essential. Extensive records are kept, including audio and video tapes, notebooks and illustrations. These may form the basis of further discussion and in certain cases may be re-used in future projects.

Methods used in the project

Central to this project was the creation of a fully functional prototype, and its use in field trials. The working demonstrator system described above was constructed and used in trials over a period of nearly two years. Because there are separate tools for audio, video and shared workspace[3] and these are configurable, it is possible to construct conferencing systems to support a range of different applications. In effect, the team had access to an immensely flexible rapid prototyping tool. Not only the appearance of the interface can be configured, but also modes of

[3] In this case the tools used were vic [Jacobson 1994], rat [Hardman et al 1995], wb [Jacobson 1993] and nte [Handley 1996]. The references are to technical papers.

communication[4]. New elements can also be attached as required. The process has been cyclical, with the activities described below feeding into development of the system, which in turn led to further associated activity.

Observation was a key part of the process. Teachers were observed both using the system and in the classroom, by developers and education experts. Results were recorded digitally and on paper and formed the basis for discussion and written reports. At key times, questionnaires and rating scales were administered, and unstructured interviews were conducted with students, tutors and other language teachers outside the project. A variety of group activities were constructed, using focus groups to draw out participants' opinions and experiences, and workshops to encourage thinking about future developments. The composition of groups was varied (with teachers and students sometimes together and sometimes apart). Digital recordings of lessons were made using a conference recorder developed at UCL[5]. This has proved to be a non-intrusive way of obtaining a full record and has produced a huge amount of data, the use of which is discussed in the next section. Alongside this, written and spoken comments from teachers and students have been analysed with a view to determining which terminology and concepts cause difficulty and to evolving a shared vocabulary for discussing certain aspects of the system, particularly audio and video quality.

Results

As reported by Watson & Sasse (1996) and Hughes & Sasse (1997), the efforts to involve users in design have resulted in a system that is fun and easy to use. The highly interactive tutorials were agreed to be rewarding, intense and involving. Language teacher assessors particularly applauded the fact that reading, writing, listening and speaking skills were developed together. Although the system was designed for teaching foreign languages, the lessons for teacher development are probably applicable to other subjects.

The uncertainties of computer networks

Teachers have gained a greater understanding about the nature of networked systems and, in particular, appreciate the uncertainties involved and have learned to plan for them. Conditions in shared networks can fluctuate rapidly. Congestion may cause sudden changes in video or audio quality which can be alarming but are usually transitory.

[4] An example,in our configuration, is the choice of full duplex audio rather than the default push-to-talk mode. See Watson & Sasse (1996)

[5] UCL – University College London

Once teachers experience this a few times, they learn to plan for a change in activity in the same way that they are flexible in a classroom. Training can prepare them for this.

Another disconcerting factor is that in such systems some things can not be known for certain. Failure to reply to a question, for example, can be the result of local equipment failure, software failure, a problem with a link in the network or simply the fact that someone has taken off a headset or muted a microphone. Teachers gradually learned how to eliminate the most obvious causes but the problems this can cause should still not be underestimated.

New ways to communicate

Superficially, a lesson over the ReLaTe system is similar to one in a classroom, but computer-mediated communication, even with the aid of audio and video, differs from face-to-face communication. Teachers found it harder to sense the group atmosphere or to detect when the class needed a change of activity. They did, however, feel they could still sense an individual student's personality, and they talked of their students in the same ways as do face-to-face teachers. During lessons, more overt signaling was needed. Teachers could not use eye contact to indicate which student was to speak and, similarly, students could not convey by body language that they would rather not be asked a particular question. Remembering that others could see only what was within the camera's field of vision also had to be learned; words were needed to explain actions that would be obvious in a face-to-face situation. The interplay between communications channels (the whiteboard, audio and video) was also slightly different from that between the face-to-face equivalents.

Managing a new environment

Teachers developed experience and greater understanding about managing this unfamiliar working environment. They were constrained physically by the equipment and cut off from external stimulus by headsets. This made it harder to retain a sense of time and to pace a lesson. The trials also raised questions about the boundary between technician and teacher. To have full control over resources, the tutor must understand a computer file system and the formats in which material can be stored. Finally, the physical environment must be managed, so that there is sufficient space for equipment and the lighting enables people to see one another but is not uncomfortably hot or bright.

Preparing teaching materials

It is easy to assume teachers know already how to choose material and devise activities around it. Here, as elsewhere, some lessons had to be unlearned. For example, incorrect assumptions were sometimes made about translating from classroom to system domain. Although the shared workspace usually performed the function of a photocopied worksheet, there were times when the camera was a better means of showing things to students. Similarly, the camera was not always an adequate substitute for holding an object up for the class to see. Material may have to be altered in order to display it effectively on a computer screen and if teachers are not to be over-reliant on technical support they must learn to use machines, such as a scanner.

Adding new elements

As in most courses, a varied mode of delivery was needed. Teachers began to explore ways of integrating this real-time communication with e-mail and the world wide web. Having to learn more new applications was a burden, however, and this was a good example of the extra cognitive load imposed on both tutors and students by the use of multiple software applications.

Training materials

Because these lessons were observed and discussed so fully there is now a large store of materials, including recordings of lessons, images of whiteboard and text editor screens, audio recordings, observers' notes and reports. These have already proved an effective stimulus to discussion but they could also be used in training. It is hard to give teachers the time to watch their colleagues at work. Recordings offer them the chance to observe lessons in their own time. It is also possible to interact with the recordings, adding commentary or questions, and to transmit them to remote locations. Trainees need not all be on one site.

Terminology and concepts

There was an evident gulf between teachers and developers here. It also proved to be important. Language is so closely linked with thought that one cannot assume teachers do not need to know any technical terms. The teachers found it difficult to discuss aspects of audio and video quality and to understand the factors that affected these. This made it harder for them to explain what changes were needed and had implications for their use of the system, especially understanding its limitations. It also made some functionality inaccessible to them. The interface was specifically designed to hide complexity but some of the hidden, more advanced features are useful. Unfortunately, they are labeled in technical terms. It is in this area that the meeting between designers and users is so crucial; the implications for interface design are as important as those for training.

Conclusions

As suggested in the previous paragraph, not all issues should be tackled by training the teachers; system and interface design must minimise usability problems too. Some of the findings from this study have fed into work on redesigning the interfaces to a number of conferencing tools. The main idea behind this re-design is that users bring to their work with a computer system a model based on general knowledge and knowledge of the task they are trying to perform. This model is not always appropriate; misconceptions embedded in it can lead to problems in using the software. Interface designers can facilitate the acquisition of a more appropriate user's model by choosing labels compatible with the user's general knowledge and the tasks to be performed. Clark & Sasse (1997) provide an example of successful application of this theory of conceptual design to an Internet tool.

The main message, however, is that training is needed if teaching and learning with such systems are to be optimised. Such training must extend beyond operating the controls and it is likely that a good programme would cover all the areas outlined above, as well as a number of discipline-specific topics. Some areas can be dealt with quickly but many require discussion and reflection, which takes time.

Not surprisingly, time is also needed to develop expertise in teaching with such systems. However, this should not be seen as discouraging. Reasonable performance can be achieved quickly, provided the system is designed appropriately. As was found with the use of a "minimal manual" by Anderson et al (1993), it is not necessary or advisable to introduce every possible function at the start. A more gradual approach is particularly well-suited to in-service training. In higher education, even new recruits must learn while in service, whilst in every sector established staff need training to respond to change. Also encouraging is the fact that small-scale, discipline-specific trials can generate material that is more widely useful in staff development.

Teachers are highly skilled experts. They can be introduced to new technology without being made to feel de-skilled or lacking in competence. The teachers in this project have shown that they can participate in system design and shape the technology to suit their needs. In follow-up work, it is also the teachers and students who will select the materials to train others. However, an unexpected benefit is that members of the development team have gained greater understanding of the teachers' domain. Perhaps this will enable them, too, to train teachers effectively?

Finally, the focus of this paper has been on the tutors' learning. At times, the literature (such as Williams (1994)) suggests there is a conflict between evaluating educational software as a tool for the teacher and evaluating it as a tool for the learner. In the type of real-time system under discussion, however, the participants interact throughout. The approach in this project has been to involve students fully and the expectation is that they too will benefit from further work in this area. Interestingly, they regarded the fact that their teachers were, like them, learning to use the technology as a positive and equalizing factor.

Acknowledgements

The authors wish to acknowledge the following contributors to the work reported here:
At UCL: Vicky Hardman, Isidor Kouvelas, Anna Watson, Louise Clark, Colin Perkins, Kristian Hasler.
At Exeter: John Buckett, Mark Pack and Gary Stringer.

Special thanks are due to teachers and students from the Foreign Language Centre at the University of Exeter (Director Elizabeth Matthews) and the UCL Language Centre (Director Dolores Ditner) and to the expert assessors, Marie-Danielle Chamary, Marc-Georges Nowicki and Dr. Carmo Ponte.

References

Anderson, A., Knussen, C.L., Kibby, M.R. Teaching teachers to use Hypercard: a minimal manual approach. *British Journal of Educational Technology.* Vol 24 (2) 1993

Bates, A.W. (Tony) (1995) Technology, Open Learning and Distance Education. *Routledge Studies in Distance Education.* Routledge 1995

Carmel, E., Whittaker, R. D., George, J. F. (1993) PD and Joint Application Design: a Transatlantic Comparison. in *Communications of the ACM*, special issue on Participatory Design, June 1993

Clark, L. & Sasse, M.A. (1997) Conceptual Design Reconsidered - The Case of the Internet Session Directory Tool. *Proceedings of HCI'97*, Bristol, August 12-15. Springer.

Clement, A., & Van den Besselaar, P. (1993) A retrospective look at PD projects. *Communications of the ACM*, special issue on Participatory Design, June 1993.

Grudin, J. (1991) Obstacles to User Involvement in Software Product Development, with Implications for CSCW. *International Journal of Man-Machine Studies*, 1991, 34(3)

Hardman, V., Sasse, M.A., Handley, M.J., Watson, A. (1995) Reliable Audio for Use over the Internet. *Proceedings of INET'95*, June 1995, Honolulu, Hawaii.

Hughes, J. & Sasse, M.A. (1997) Internet Multimedia conferencing – Results from the ReLaTe project. *Proceedings of the ICDE Conference*, Pennsylvania, 1997

Jacobson V. (1994) VIC manual pages, Lawrence Berkeley Laboratory (LB.L), 2 November 1994

Jacobson, V. (1993) WB README file, Lawrence Berkeley Laboratory (LBL), August 12th 1993.

Handley, M.J. (1996) Handley, M.J. Network Text - A Scaleable Shared Text Editor for the Mbone. UCL Computer Science Department Research Note, RN/96/13.

Kensing, F. & Munk-Madsen, A. (1993) PD: Structure in the Toolbox. *Communications of the ACM*, special issue on Participatory Design, June 1993.

Macedonia, M.R. & Brutzman, D.P. (1994) Mbone provides Audio and Video across the Internet. - *IEEE COMPUTER*, April 1994.

Muller, M.J., Wilman, D.M., White, E.A. (1993) Taxonomy of PD Practices: a Brief Practitioner's Guide. *Communications of the ACM*, special issue on Participatory Design, June 1993

Muller, M.J. (1991) PICTIVE – An exploration in participatory design. *Proceedings of CHI'91*

Wall, P. & Mosher, A. (1994) Representations of Work: Bringing Designers and Users Together. *Proceedings of PD94*, the Particpatory Design Conference, October 1994
Watson, A. & Sasse, M.A. (1996) Assessing the Usability and Effectiveness of a Remote Language Teaching System *Proceedings of ED-MEDIA'96*, Boston, July 1996
Williams, Marian G. (1994) Enabling Schoolteachers to Participate in the Design of Educational Software. *Proceedings of PD94*, the Particpatory Design Conference, October 1994

Jane Hughes is a Lecturer in Educational Technology in the Higher Education Research and Development Unit, University College London, 1-19 Torrington Place, London WC1E 6BT. E-mail jane.hughes@ucl.ac.uk.

Angela Sasse is a Senior Lecturer in the Department of Computer Science, University College London, Gower Street, London WC1E 6BT. E-mail a.sasse@cs.ucl.ac.uk.