

**LANCASTER
UNIVERSITY**

**Computing
Department**



Situated evaluation for cooperative systems

Michael Twidale, David Randall[†] and Richard Bentley*

Centre for Research in CSCW

Research report : CSCW/11/1994

[†] Dept. of Interdisciplinary Studies, Manchester Metropolitan University

* GMD, Germany

©University of Lancaster 1994. Copying without fee is permitted provided that the copies are not made or distributed for direct commercial advantage and credit to the source is given. For other copying, write for permission to:-

Computing Department, Lancaster University, LANCASTER, LA1 4YR, UK. Phone: +44-524-593041; Fax: +44-524-593608; E-Mail: cscw-info@comp.lancs.ac.uk

Situated evaluation for cooperative systems

Michael Twidale, David Randall, Richard Bentley

Computing Department
Lancaster University
Lancaster LA1 4YR
UK

Tel: +44-524-65201 x3112
E-mail: mbt@comp.lancs.ac.uk

Department of Interdisciplinary Studies
Manchester Metropolitan University
Manchester M15 6BG
UK

Tel: +44-247-3037
E-mail: D.Randall@mmu.ac.uk

GMD FIT
Schloss Birlinghoven
53754 Sankt Augustin
Germany
Tel: +49-2241-14-0
E-mail: bentley@gmd.de

ABSTRACT

This paper discusses an evaluation of the MEAD prototype, a multi-user interface generator tool particularly for use in the context of Air Traffic Control (ATC). The procedures we adopted took the form of opportunistic and informal evaluation sessions with small user groups, including Air Traffic Controllers (ATCOs). We argue that informal procedures are a powerful and cost effective method for dealing with specific evaluation issues in the context of CSCW but that wider issues are more problematic. Most notably, identifying the ‘validity’ or otherwise of CSCW systems requires that the context of use be taken seriously, necessitating a fundamental re-appraisal of the concept of evaluation.

KEYWORDS

Evaluation, Ethnographic Observation, Rapid Prototyping, Multi-user Interface Design, Air Traffic Control.

INTRODUCTION

Few practitioners in CSCW would wish to contest the importance of evaluation work. In principle, evaluation should be a significant check of a system’s capacity to deliver what is required of it. As Grudin [12] has noted, the evaluation of CSCW systems is especially difficult for a variety of reasons, including the effect on performance of the behaviour and personalities of other group members, the effect of social, motivational, economic and political dynamics and the importance of time, in that group interactions may unfold over days or weeks.

Grudin cites the difficulty of evaluation as just one contributory factor in why CSCW systems fail to deliver the benefits intended. Indeed, we may regard all his case studies as examples of a failure to adequately determine what is being evaluated, when it is appropriate to evaluate,

and what methods are likely to prove suitable when the focus of evaluation moves from system functionality to system use [1, 21]. These problems became evident to us through our experience of evaluating the MEAD system [3, 5], a rapid prototyping tool developed as part of a project investigating multi-user interface design in the context of Air Traffic Control [4]. Although, as we recount below, we believe our evaluation studies contributed useful information concerning the effectiveness of MEAD, in the course of the studies we became increasingly conscious of their limitations. Our problems in finding an acceptable basis for ‘validating’ the system led us to suggest there is a pressing need for the reappraisal of evaluation philosophies and techniques. In particular, the view that evaluation should be regarded principally as a summative process which takes place at a given stage in the software life cycle and which yields ‘objective’ results is, we believe, deeply problematised by CSCW’s interest in the ‘real world’ context of use. Rather, we argue there is a need for practitioners to identify common experience of problems in the evaluation of CSCW systems.

Our aim in this paper is to begin this reappraisal process by recounting our own practical experience of evaluation work, primarily in the context of MEAD, but drawing also on other research work in which we have been involved. We aim to identify approaches that resulted in useful information about the performance of MEAD and its relevance to the concerns of those who might use it, and perhaps most importantly, to specify the limits of these approaches. We do so in the hope that we will prompt others to engage in a similar exercise, with the ultimate aim of uncovering evaluation measures which relate directly to CSCW problems.

WHAT IS EVALUATION?

The issue of evaluation was raised fairly late in our Air Traffic Control (ATC) project. This happened largely because we had presumed, somewhat naively as it turned out, that as such exercises tend historically to occur late in the cycle, it was advisable to allocate specific time towards the end of the project for our own activities. This assumption was profoundly challenged, as we shall see, in the course of the work. Our initial concern was to identify the different ways in which the primary question of ‘what is

evaluation?’ can be answered. Our review suggested rather more competing viewpoints than we had expected, reflecting the differing intellectual backgrounds and histories of those who had been involved in this kind of activity. Some of these are listed in Table 1.

1.	An assessment of the overall effectiveness of a piece of software, ideally yielding a numeric measure by which informed cost-benefit analysis of purchasing decisions can be made.
2.	An assessment of the degree to which the software fulfils its specification in terms of functionality, speed, size or whatever measures were pre-specified.
3.	An assessment of whether the software fulfils the purpose for which it was intended.
4.	An assessment of whether the ideas embodied in the software have been proved to be superior to an alternative, where that alternative is frequently the traditional solution to the problem addressed.
5.	An assessment of whether the money allocated to a research project has been productively used, yielding useful generalisable results.
6.	An assessment of whether the software proves acceptable to the intended end-users.
7.	An assessment of whether end-users <i>continue</i> to use it in their normal work.
8.	An assessment of where the software fails to perform as desired or as is now seen to be desirable.
9.	An assessment of the relative importance of the inadequacies of the software.

Table 1: Definitions of evaluation

The problem of competing definitions is of course compounded by the fact that evaluation activities may well be directed at more than one of the above definitions. Adding to the complexity of the evaluation task is the multitude of techniques that can be used. These can be classified in a number of orthogonal dimensions including:

- summative <-> formative
- quantitative <-> qualitative
- controlled experiments <-> ethnographic observations
- formal and rigorous <-> informal and opportunistic

Although it is quite possible to discover evaluations in the literature that fit into each of the possible dimension combinations, in practice we may conflate these dimensions to a single dimension with the two most common styles of summative-quantitative-controlled experimental versus formative-qualitative-opportunistic having an oppositional relationship. For convenience we shall refer to these as formal and informal techniques.

The influence of software engineering can be discerned in at least three of the definitions mentioned, in that definition

(2) broadly corresponds to the software engineering concept of verification, definition (3) to validation, and definition (6) to acceptance testing. Amongst the numerous evaluation techniques, it is clear that a scientific paradigm influenced many of them. The influence of the paradigm is apparent in the tendency to regard formal techniques as somehow more ‘proper’ because the ‘proving’ of the system renders it ‘complete’. Certainly, the summative evaluations characteristic of psychology, education and HCI research belong to the scientific paradigm. Part of computing also belongs to that paradigm in that it draws on these resources, but it equally draws on the paradigm of engineering. That these are different is illustrated by the fact that within the engineering paradigm, and unlike in science, proof by construction is permissible. Thus the historical relationship between engineering approaches and ‘scientific’ evaluation is not a necessary one.

In our view, both the scientific origins of formal evaluation and the more pragmatic combinations of formal and informal approaches associated with commercial systems development may be inappropriate in a context where a new paradigm is struggling to be recognised [16, 22]. In this respect, the surfacing of doubts concerning the relevance of historically-evolved and discipline-specific procedures for evaluation parallels the concern over requirements analysis which is characteristic of the CSCW community.

Formal techniques tend to be the most frequently advocated in the academic research literature, arguably because some ‘fit’ between science and engineering is commonly assumed. It would appear that controlled experiments are generally perceived to be more objective and to offer the potential of reproducibility. (Strangely however, unlike in the natural sciences, it seems that evaluation experiments are rarely reproduced.) Associated with their ‘scientific’ status, experimental techniques can additionally be attractive to developers and their clients as they offer the promise of risk reduction precisely because they proffer measurements of productivity or effectiveness. On the other hand, there are a number of disadvantages associated with formal techniques, not least their cost in terms of materials, equipment usage, acquisition of subjects, and the labour of the experimenters in planning, administering and analysing the results. A related problem is that the whole activity takes a great deal of time which may not be available within the deadlines of a project. Despite these problems, the immense literature on evaluation in the interlinked fields of education research, psychology and HCI provide only a minority of cases in which more informal techniques have been advocated, mainly for rapid formative evaluations [2, 19, 20].

Our interest in the first instance lay in the possible value of using formative, incomplete, and ‘subjective’ evaluation as an alternative. We were and are satisfied that they are appropriate, given that our concern was with obtaining information that would allow us to change the system, rather than with definitive information about the system as it stood. Such informal evaluations have been used in the engineering approach of rapid prototyping or iterative

design for the development of both general computer systems and CSCW systems [7]. We were conscious that the presumed benefits of objective results do not seem to accrue universally, even when the systems under consideration have not been historically regarded as CSCW systems (see for instance Vincent [31] for a discussion of productivity and office automation), and were interested in what value might be derived from a more informal stance. The fact that scientific testing does not, on the face of it, universally confer the benefits that one might expect was reassuring given our own opportunistic approach, though at the time we were not in a position to understand quite why this might be so.

However, and we stress that at the outset we had little purchase on these issues, radical critiques of evaluation have recently become available. Bannon [1], for instance, argues that design, use and evaluation should be viewed not as distinct activities, but as being necessarily interwoven. Evaluation can be understood as a process which should saturate and be constitutive of the design process precisely because the ‘context of use’ is central to the analysis of CSCW systems. If one accepts such an argument there are immense implications for the issues we have mentioned above and we draw on this and on the similar arguments in [21] to argue that while informal evaluation procedures have a value for specific purposes, they constitute only modest revisions of the status quo. Identifying the ‘completeness’ or ‘validity’ of MEAD turned out to be quite beyond the scope of our chosen methods, largely because we had not at that stage taken the step of imagining what such a ‘situated’ evaluation might look like, and the analytic purchase that shifting the focus of evaluation from the computer system to the socio-technical whole might bring. Our discussion concludes with an examination of this issue.

EVALUATING MEAD

We re-iterate here that these issues were only vaguely glimpsed when we began the process of deciding how to undertake an evaluation. At the outset our problem was to decide which, if any, of the purposes cited in Table 1 were relevant to our concerns, and which of the associated methods, if any, should be adopted. Our stance throughout the exercise was a pragmatic one, largely because it was unclear how the ongoing collaboration between sociologists and computer scientists [4, 17] would affect our evaluation procedures. Nevertheless, we shall argue in the light of our experiences that the ‘objectivist’ stance associated with validity and acceptance testing in particular, and the attendant mathematisation of technique, is misconceived.

The problems we encountered led us to view informal evaluation techniques as useful given an orientation towards ‘incompleteness’ rather than ‘completeness’, although no objective conclusions about system validity or acceptance are possible. This implies that summative strategies, aimed at providing such objective conclusions, are likely to prove inappropriate in domains where the context of use may vary in significant ways, and these domains are likely to be

those in which CSCW has an interest. We return to these issues in the latter part of this paper.

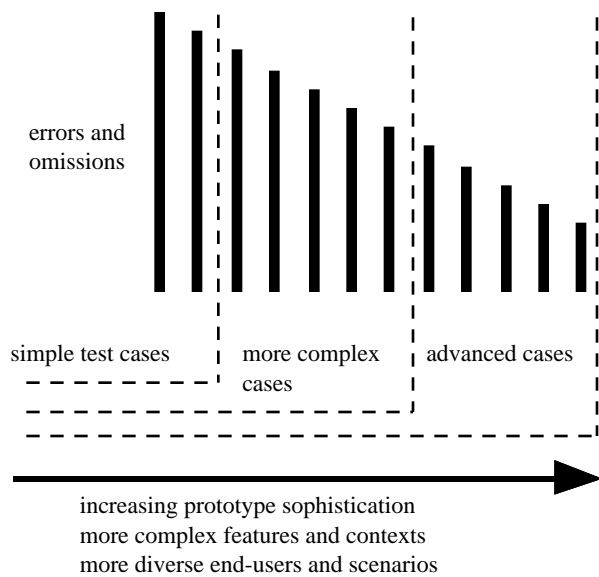
At the outset our problem was to decide on appropriate purposes and techniques. We drew substantially on the literature on evaluation in education research, particularly on research into computer based learning systems and Intelligent Tutoring Systems. Many of our concerns had already been debated in this context and a number of approaches proposed [24]. An important example is the distinction between evaluating the power of the software, the effect it has on learners in experimental conditions and the degree to which the software continues to be acceptable in the intended environment. The latter may be characterised as, ‘do the teachers go on using it after the evaluators have left and it is no longer a novelty for the students?’

As a matter of practical purpose, and given that our project had been from the start concerned with the potential value of ethnographic insights for systems design, we also decided that our methods should at least in part utilise the domain knowledge of the ethnographer. Given the impressionistic and informal nature of ethnographic data, we felt that a collection of informal methods would be suitable, although we had no firm convictions at that point as to the relative merits of the various techniques available. Our reasoning was in part also due to the difficulty of deciding what criteria might be used to determine ‘completeness’ in a tool used to support rapid prototyping, as our aim was not to prove the completeness of the system, but to identify incompleteness. In other words, we wanted to be able to say something about what the system failed to do, and perhaps what should be done about it. We opted to undertake a formative evaluation with a view to identifying problems with the system, their relative importance, and discovering how such a prototyping tool might be used.

We were conscious that much of our argument elsewhere, and that of others in the CSCW community [26, 15] has been for a naturalistic approach to the study of work, on the grounds that such approaches have much to tell us about the ‘situated’ nature of the collaboration. By implication, a similar approach to evaluation work would confer many benefits about the actual use of new systems. However, for us, the prospect was infeasible, since it would have been difficult to set up, costly, and time consuming to do, particularly for continuing formative evaluation of an unfinished prototype. Fortunately a characteristic of formative evaluation means that less ‘authentic’ studies may yield valuable preliminary results. That characteristic may be phrased as, “all problems with a system scale up and out; any success may not”, and is particularly true of a system’s user interface (Figure 1). So, to take the extreme case, if even a system developer finds a feature of the system clumsy or confusing to use, we can be reasonably confident that the same will be true for a novice end-user.

The converse of this characteristic is obviously untrue and has been noted by HCI evaluators and used as evidence for the importance of evaluation with end-users [9]. Although what such evaluators say is quite true, they miss the point

that as part of ongoing formative evaluation, less authentic testing can be a valuable way of eliminating grosser system errors in a more economical manner. If those errors remain until later, not only will the delay mean that they are more difficult (and costly) to rectify, but the gross errors will completely swamp the interaction making it unlikely that more subtle but still significant system errors will be observed at all, as shown in Figure 1. In this regard, our approach to evaluating MEAD was informed by our earlier work on the design of the user interface for the Designers' Notepad - a system to support the early stages of design [29, 30]. Here, initial interface testing was performed by the system developers who undertook authentic tasks such as preparing talks or papers. This revealed some problems, including lack of functionality, which were rectified. Subsequent testing involved end-users increasingly remote from the development group including project members not involved in systems development, postgraduate computing students, visiting academics, colleagues from Sociology, undergraduate computing students, and undergraduates with minimal computer experience. The advantage of this approach is that it does not take large numbers of subjects to reveal problems with a developing system, and so one can discover what is wrong quickly and cheaply.



Subtle errors/omissions are unlikely to be detected until less subtle errors are rectified. Simple test cases do not involve some activities and so cannot detect problems concerning them.

Figure 1: Scaling up

Although it was not possible to locate the MEAD prototype in the actual work situation, we made some attempt to maintain authenticity by providing tasks which subjects would regard as authentic tasks to perform. In this manner we hoped to avoid the inherent danger of MEAD's

designer embodying his possibly incorrect beliefs about the activities to be supported by the tool - if the subjects are asked to perform tasks which relate to their particular concerns, the tool will be directly challenged as to whether it can support those tasks. Equally, we were persuaded that in depth observation of a small number of individuals would enable us to discover something of the kind of problems they would encounter in using the system, with no suggestion that we might discover all relevant problems.

Our interest lay in revealing the kinds of misconceptions that end-users may have. In the context of a summative evaluation, a possible objection to this would be that the small number of end-users might be unrepresentative. However, in one sense no end-user is unrepresentative in that all end-users' viewpoints and requirements reflect a context in which the system may have to function and, in the first instance, we were concerned to identify some and not all ways in which the system might under-perform or cause problems. In this respect we were influenced by a study reporting problems that arose with Cognoter [27]. This study was very much in the style we chose to adopt, involving authentic tasks and in depth small-scale study (two groups of three each having two 2-hour sessions). Despite the limited size of this study, major problems with the system were discovered and generalised.

A further problem we had to contend with concerned the relationship between MEAD's user interface and underlying features of the software. A poor user interface can distort the performance of any system by confusing or distracting end-users. Likewise, if a system has a particularly good interface, much of the improvement in performance may be due to how that interface supports an end-user, freeing resources to concentrate on the task in hand and acting as a supportive environment, thereby reducing the number of issues to be maintained in working memory [28]. One must therefore be careful to correctly ascribe both advantages and disadvantages of a system, as performance variations may result more from changes to the interface than from underlying software features that purport to support the collaborative nature of a task more effectively. In an ideal world one would have the time and resources to undertake work to determine these relative effects, but this is a luxury available to few issues and projects.

The development of the MEAD multi-user interface prototyping system was informed by a series of ethnographic studies of ATC [4, 14, 25]. These studies fed into the functionality provided to support rapid prototyping of novel ATC interfaces. Hence the evaluation of MEAD was intended to assess the extent to which the system supported the process of such prototyping, and the system's effectiveness was to be determined by the speed and ease with which different interface designs could be implemented and iteratively evaluated and refined. (Note the recursiveness of the evaluation problem in this context: partly we are evaluating how our system supports evaluation!) In the rest of this section we describe the preliminary evaluation of a completed prototype of MEAD.

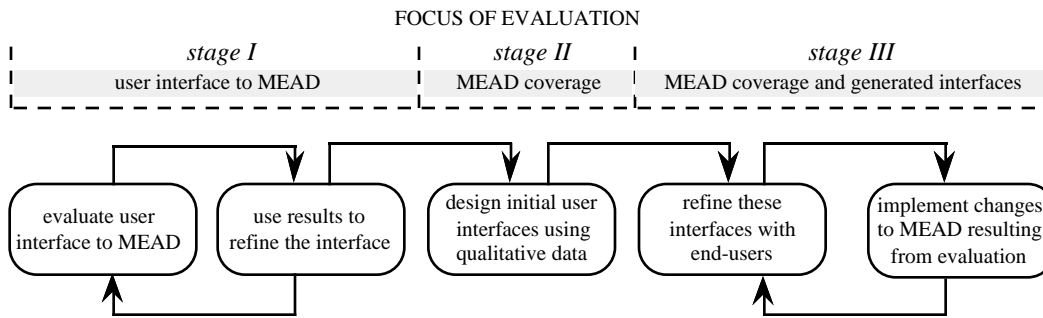


Figure 2: Stages of the evaluation process

What to evaluate

The nature of the MEAD system meant that it could be evaluated in three distinct but interconnected ways:

- *The user interface to MEAD*
(The ease of learning and use of the tools provided for interface developers)
- *The coverage of MEAD*
(The ability to generate the kinds of interfaces required by end-users)
- *The user interfaces produced by MEAD*
(The usability of the interfaces that can be generated)

All three aspects contribute to the system's overall effectiveness and are closely related. However, we first needed to ensure the system did not fail to support end-users' requirements due to the inflexibility of its user interface. Other aspects contributing to effectiveness could only be investigated once it had been established that the tools MEAD provides could be used flexibly. This removal of potential 'swamping effects' of errors in the MEAD user interface was the focus of stage I of our evaluation process (Figure 2). Stage II involved the design of initial user interfaces for ATCOs and in stage III these interfaces were refined with end-user participation.

The evaluation was intended to be formative and opportunistic, merely to discover some of the most visible issues for future work. Even for these purposes, it would be worthwhile to undertake simulations of realistic controlling situations and the environment in which they take place. Such simulations were however beyond the budget and available time of the project. Even for informal ethnographic studies we had to bear in mind the cost and scarcity of end-users (ATCOs). For this reason, stage I of the study involved undergraduate computing students. The scaling up and out argument presented above means that any problems identified and rectified by this study could be found at a cheaper cost than by using an authentic ATC activity.

Stage I: Evaluation of the MEAD user interface

For the purposes of this study, aimed at discovering problems with MEAD's interface, two undergraduate computer science students acted as interface developers. The students sat side by side working at separate machines and

were encouraged to discuss problems they were having; it was hoped that this arrangement would establish a "constructive interaction protocol" [11] where the emerging conversation between the students would give information about what they are trying to achieve and problems they are having. To focus attention on MEAD's interface, the session was based on an old course work exercise very familiar to the students - the design of a user interface for a simple Theatre Booking system. Help was only provided when the students could not see a way forward.

Discussion

The students completed the exercise comfortably. The majority of the problems encountered were either with unfamiliar concepts or with aspects such as the wording of button labels and menu entries. The students rarely asked for detailed help about what to do, but rather wanted to know how the various features of the tools could be accessed and manipulated. The study did reveal usability problems with the system, causing it to crash with certain action sequences. This problem arose due to a lack of error checking caused by assumptions made by MEAD's designer about how the system would be used; an example of system errors due to the designer's implicit assumptions. This simple study shows the relative ease with which many of these errors can be detected by informal testing. Following the study additional error checking was added to MEAD and some menu options and button labels which had caused confusion were re-worded.

Stage II: Design of the initial ATC interfaces

Using the results of social studies of ATC work, interfaces were prototyped with MEAD that offered slightly different facilities to those already in use by ATCOs. Currently ATCOs use two main information displays. Firstly a radar shows current aircraft position as *blips* with *datablocks* next to each blip giving identification numbers and current height. Extra information can be displayed; for example, a *track* of three dots for each aircraft can be displayed indicating its past course, speed and heading. The second display is a set of paper *flight strips* held in racks in front of the controller. Much of ATCOs' work involves positioning and writing on these strips. All controlling decisions, such as instructing aircraft to climb, descend or change heading, are recorded on the strips in ink. This information is then available to different members of a team

of controllers. The public visibility of information is vital in controlling work; even during periods of heavy traffic density when flight strips often overflow the bays, they are still positioned where they can easily be seen by all team members.

Electronic flight strip display

While replacing the paper strips with electronic strips 'under glass' would have many advantages [25], it would also have problems such as limited screen space. Some previous attempts to produce electronic strip systems have provided a scrolling mechanism to allow access to strips which cannot fit on the screen (e.g. [23]). Social studies of ATC work have stressed the importance of strip visibility, and scrolling is therefore not suitable for an electronic strip display. Our studies of ATC did however show that most of the information on the strips is provided for use by one member of the controlling team - the ATCO in direct contact with the aircraft or the *radar controller*. This information is not required at all times, so we used MEAD to prototype an initial electronic strip display that allowed the radar controller to 'fold' strips to smaller versions that displayed only the information potentially needed by other team members.

Augmented radar display

The data block on the radar screen shows only a small amount of information about each aircraft, and ATCOs must often relate a radar blip with the aircraft's corresponding strips. As the strips are currently paper, there is no link between them and the electronic radar display, so the task of locating strips involves visually scanning the strip racks. Our studies suggested a need to display information on the radar screen that is currently only presented on the flight strips. Using MEAD an augmented radar was designed that allows blips to be expanded to alternative representations displaying more information.

Stage III: Refinement with ATCOs

Both the electronic strips and the augmented radar were based on existing ATC displays, but were modified in the light of information from social studies of ATC. The next stage of the evaluation was to refine these initial designs with the participation of real end-users. Our approach was similar to that sometimes described as cooperative prototyping [6], where feedback from end-users informs immediate prototype refinement. To support this, the ATCOs were not given information about what MEAD could or could not do so they would not feel constrained in discussing the features they wanted in the evolving interface (and thereby allowing identification of features missing from and inflexibly realised in the existing version of MEAD). For the sessions MEAD's designer acted as the interface developer with access to the system's development environment. (Again the scaling up argument makes this appropriate for finding problems - the next evaluation stage would require someone other than the designer to take this role).

The first ATCO session

The session began with a demonstration of the initial display designs developed in Stage II. The ATCO chose the augmented radar as a starting point based on the flexibility with which information could be added and removed. To explore a novel arrangement of information, the ATCO suggested we expand the augmented radar to present all the dynamic, constantly changing information about aircraft, such as heading, position, coordinated height and so on, while restricting the strips to static, reference information such as aircraft type and expected route. This division would allow investigation of different patterns of sharing, as team members could each have personal displays configured to their own tasks, but could all share the static, reference information display.

The ATCO wanted to keep the track facility from existing radar displays that super-imposes three dots on the screen to show previous aircraft position. The system could not provide this as it requires displaying a function of the position history for each aircraft; there was no way to specify that the position of graphical components on the screen should be based on functions, as in graphical constraint systems such as Rockit [18]. Instead it was decided to try and modify the radar blips to show only the dynamic information required by ATCOs. This revealed more system limitations; an early design decision meant that in order to modify the radar blips a shutdown and re-start process was required. This process took about a minute and if needed several times in a session, was far too long to maintain the interest of end-users as well as being tedious for the interface developer.

During one shutdown/re-start pause, the ATCO discussed a facility provided by some radar displays that automatically indicate if an aircraft is equipped with an on-board Traffic Collision Avoidance System (TCAS). ATCOs find this useful as TCAS can cause aircraft to deviate from their course without ATCO instruction. This facility was added to our display so that a 'T' was shown if a represented aircraft was TCAS-equipped.

Discussion

The shutdown/restart problem arose from a difference in approach to interface development taken in this session to that used by MEAD's designer during system development. During this session a large part of the interface refinement required experimentation with different layouts, sizes, colours and so on - the ATCO was keen to try many different configurations to find the most acceptable, requiring many small changes to the interface definition. This was not the approach used by MEAD's designer for the initial displays, where there was an understanding of what was to be developed from the start, and the definitions were completed before the results were displayed on the screen. A further distinction was the designer's ability to see how a display would appear by examining its definition; the ATCO did not want to look at the definition and was only concerned with how it looked as part of the display on the screen.

It is perhaps surprising that this problem was not discovered before, during system development or when the initial displays were developed. What is more surprising is that the problem was not caught in stage I of the evaluation, which focused on just this sort of problem arising from incorrect assumptions on the designer's part regarding how MEAD would be used. With hindsight, it was obvious that the reason for this lay in the designer's formulation of the theatre booking task performed by the students. This problem is summarised by Twidale [28]:

There are inevitably implicit assumptions about the nature and style of use (and about the user, task etc.) in the design of the tool. In making up test problems, developers are in danger of incorporating the same assumptions. Thus the study will fail to reveal them.

By trying to simplify and structure the task to be performed by the students, MEAD's designer imposed the same process of interface design as had been assumed during the system's development.

Although the results of this session seemed overly negative, it was still possible to prototype alternative information representations and, in particular, incorporate the TCAS facility described by the ATCO. In the light of these results MEAD was modified to resolve the shutdown/restart problem and it was hoped that the system could now be used in another session to support the rapid prototyping process more effectively and build a more realistic user interface.

The second ATCO session

A different ATCO participated in the second session, which commenced with another demonstration of the initial display designs. Based on the idea of strip 'folding', it was decided to prototype an electronic strip interface similar to one the ATCO had seen during a secondment to the UK ATC development centre. The interface resulting from this session, which closely reflected the ATCO's requirements, is illustrated in Figure 3. These requirements were based on a series of rough sketches quickly made by the ATCO, which were then translated into preliminary strip and display designs and iteratively refined.

A detailed description of the session which resulted in the interface shown in Figure 3 is not appropriate for this paper and interested readers are referred to [3]. It is however worth briefly describing the development of one feature of the interface that is characteristic of the process of cooperative prototyping MEAD supports. A facility was added to the new strip and radar displays that allowed aircraft to be *hooked* by an ATCO, causing their flight strips and radar blips to highlight and removing the need for visual scanning of strip racks to find all an aircraft's strips. Using MEAD it was possible to experiment rapidly with different methods of hooking and highlighting before the final strategy was selected.

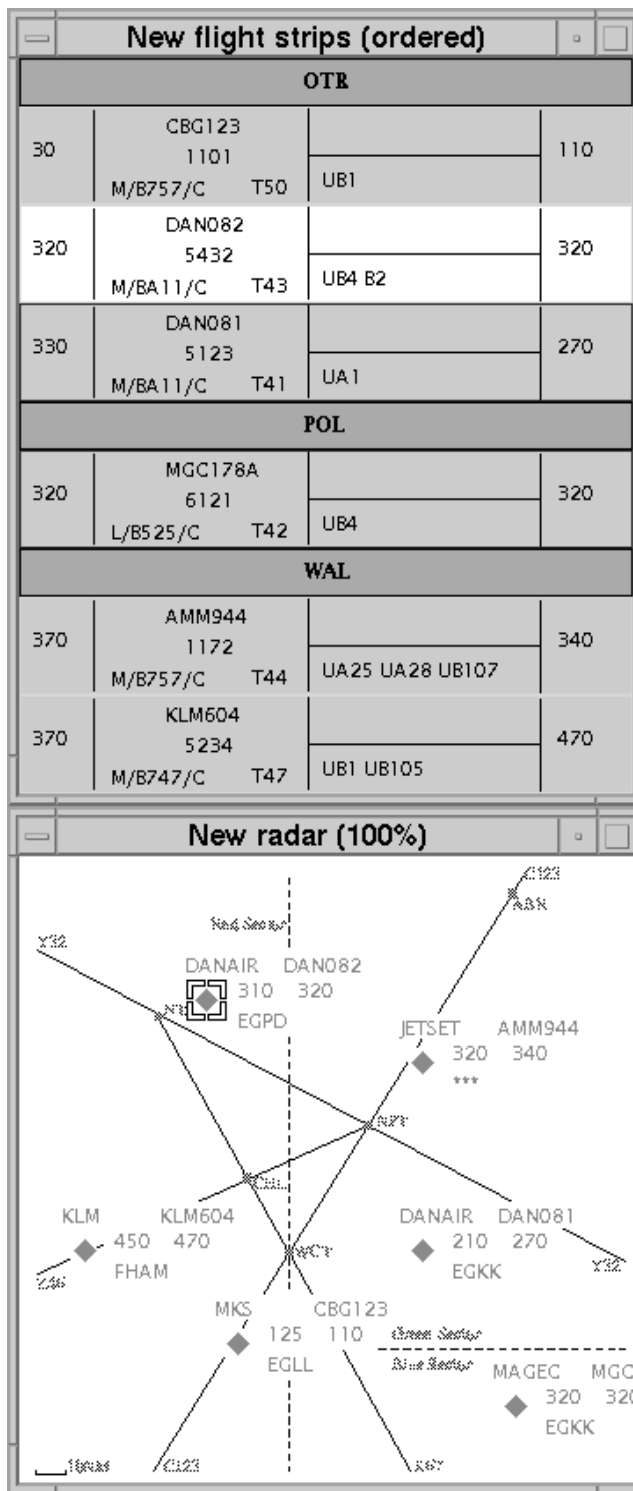


Figure 3: The interface developed with the ATCO. The aircraft 'DAN082' has been *hooked* and its flight strip and radar blip are highlighted

Discussion

The interface developed during this session closely reflected the ATCO's requirements. It was possible to experiment with different styles of presentation and interaction until the

ATCO was satisfied, and correction of the shutdown/re-start problem encountered in the previous session heightened the sense of end-user participation in the development process. By removing this problem a number of subtler and less severe problems were discovered. For example, the tool MEAD provides for designing information representations such as flight strips and radar blips was lacking in functionality that would have further accelerated the process of display refinement. These smaller errors would not have been discovered if the much grosser effects of the shutdown/re-start problem had not first been eliminated.

LIMITS OF THE EVALUATION

The evaluation of MEAD was based on a number of small, formative studies of the system in use in limited contexts. In certain respects this decision was justified by the results gleaned from the studies. In particular, the highlighting of specific system features that needed attention is unlikely to have come out of a formal analysis, where overall effectiveness tends to be the focus of the evaluation. Our informal approach revealed the grosser system errors which could be rectified before more time-consuming, costly evaluations were performed. Thus, the initial evaluation with undergraduate computer scientists had little relevance to the issue of whether such a system could be used to generate usable interfaces for ATCOs, but did reveal some significant problems that could be rectified. If these problems had been carried into the ATCO studies they would have made it difficult to assess the ability of MEAD to produce interfaces for ATC. There is little doubt that these exercises fulfilled the function of evaluation definition 8 in Table 1.

During the second session with an ATCO, MEAD was used to generate a realistic prototype of a new user interface. This was iteratively refined using MEAD's development tools with the expert end-user critically assessing each new version. The final version of the user interface prototype incorporated both the required presentation of the information and some behaviour details to allow selection of flight strips and radar blips. By this stage we felt that we had gone some way towards meeting definitions 3 and 6 from Table 1. If nothing else, the study illustrates how informal evaluation techniques can yield a substantial amount of information at relatively little cost and in a short space of time. Further, the overall impression of the system given by end-users was a positive one.

Nevertheless, it was on further reflection and particularly on analysing the transcripts of our discussions with ATCOs that we began to realise that, while our evaluations had provided useful and usable results, our procedures had been very limited in their scope. We were satisfied that we had identified ways in which MEAD was incomplete, and that in these circumscribed settings it was capable of doing useful work. Indeed, discussions with ATCOs who were actively involved in the development of new ATC interfaces gave further support to the idea that MEAD had potential value for ATCOs as a rapid prototyping tool. Thus:

ETHNOGRAPHER: "I was quite interested in what you said, about how that would have been extremely useful to you right at the start, and you termed this two phase prototyping, when you play and try out all the things you might want to do ..."

ATCO: "Yeah ... 'cause when you've not seen anything before and you have people with lots of imagination and you come up with hundreds of ideas, you've got to reject a lot of those ideas unless you can look ... you can't afford to put them onto a large scale simulator 'cause of time and cost you lose ideas before they ever get off the drawing board ... something like this would have been ideal for us because it would have helped the deadline as well because we have always been up against that ... and we never managed to get ahead ... it was always 'go firm' too quickly ... behind you is coming the people who are going to write the reqs ... they've got to know what you want ... you've got to make the decisions before you're ready to ... we didn't have anything like this ... a system was bought which took hours to change a track data block from green say to yellow ... to me as someone who's been involved in simulations for some time now you want to be able to say to the programmer 'well, we'll take a quarter of an hour break now' and just change the colour and we'll start again ... you can move along fairly quickly what we were into was, you write out a form and it would be changed days or even weeks later ... totally unacceptable at that stage ... that comes later when you're trying to stabilise ... you don't need to write it down 'cause you're not affecting the software ... you don't need to trace what you've done ... you're using a tool which is there to be changed"

However, the nature of our evaluation method necessarily focused attention on the problems with MEAD rather than conclusively demonstrating its effectiveness. It was, for instance, clear that the exercises in no way enabled us to assess continued acceptance of the system, or whether in fact the interfaces that the tool was capable of generating would prove generally acceptable to ATCOs in the course of their work. This has caused us to consider the problems evaluating systems-in-use might present, and aroused our interest in the idea of 'situated' evaluation.

Our discussions with the ATCOs provided many oblique references to the kinds of problems we were interested in. For example, it became increasingly clear that although there had been a series of trials of proposed interfaces for actual use in the ATC environment, little confidence could be expressed about their acceptability to the ATCOs. One of the most significant aspects of this was that the proposed interfaces were designed for use in a context where the work itself was likely to change. Hence:

ATCO: "we can see crew chiefs and assistants being pushed aside and ultimately controllers as well ... and you haven't got the assistants to do the writing and say there's something to work on when there's no paper strips ... I suspect they'll just say well you close the system down but what happens while you're actually

doing it I'm not so sure ... that's the message I get anyway ... and aircraft can't park ... there's something else I want to say about teamwork actually ... maybe I've given the impression that teamwork is dead ... one must bear in mind that [ATC] does involve teamwork ... the tactical and planning controllers ... but they each have their own jobs to do and we've been alluding to something else ..."

In other words, the 'validity' of the proposed interface might depend at least in part on the success or failure of new working practices. Thus 'acceptability' seems to be intricately bound up not only with the functionalities of the system but also with the potential re-design of work practice. This implies that a 'situated' evaluation would need to address not only the capacity of the 'system', but the flow of work around it.

On the face of it, such a focus seems to have much in common with currently deployed methods of organisational change, including Workflow management, Business Process Re-engineering (BPR) [8], and Total Quality Management (TQM) [10]. However, leaving aside the actual or potential value of such methods, we believe that 'situated' evaluations which deal with work and systems are nevertheless quite distinct from them. Space precludes close examination of the foundations of these managerial philosophies, but we believe they share concepts of 'process', and implicit or explicit mechanisms for monitoring 'process', which ignore the very contingencies, interruptions, and problems which arise in use, and which a 'situated' evaluation might identify. In this vein, the emphasis on teams in the TQM literature apparently parallels CSCW interest in teamwork. Our work on ATC has already highlighted teamwork as an important component of effectiveness, and the following extract reaffirms that importance:

ATCO: "teamwork shifts aeroplanes basically ... it all flows because of people who know when to point the finger at the right time and make the odd comment ... and it all flows very well ... but you cannot teach that ... it relies on something which you can't write down ... you can't design an Air Traffic Control system around it ... or at least to do it ... but you can write it out so its not necessary ... except when you've got that big black cloud coming across, or whatever ... well is it a human problem or a system problem when you take the flexibility ... the flair .. out? You still need to train ATCOs for when things go wrong ..."

Several points can be made about this remark. Firstly it suggests the elusiveness of teamwork is such that the specification of teams associated with TQM might be improved by a better understanding of teamwork and the use of technology. Secondly, and crucially for evaluation processes, a continuing uncertainty about the relationship between prospective teamwork and system elements is also evident, despite the fact that the design and implementation of new ATC systems are well under way and due for introduction in 1996. This has to do with a whole gamut of

imponderables including training, transition time, responses to new working arrangements, and individual and group capacity to learn the new system. Thus, where we were confident that MEAD was a usable system for generating interfaces that were envisaged by ATCOs on the prototyping team, there was no means to determine whether in the end a 'workable' system would result in the absence of detailed knowledge about such factors. If, as we believe, ultimate decisions about the appropriate interface for use in the real working environment depend for their validity on a complex interrelationship between system functionality and the social organisation of work, evaluation work will need to contend with the intricacies of change and change management. The 'situated' evaluation of CSCW systems should, we feel, be oriented towards the reduction of uncertainty through examination of these areas.

The total elimination of uncertainty, however, is not possible. A striking feature of our conversations with ATCOs was the considerable uncertainty about the outcome of the introduction of new technology and practices. This was felt even by those people who knew the workings of the current system from the inside. For example, during our discussions the issue of the accuracy of Flight Plans (filed by the airlines prior to aircraft take-off) and the flexibility of the current system was raised:

ETHNOGRAPHER: "but presumably you'll still get input errors ..."

ATCO: "well they'll be corrected centrally and from that point on they'll be correct for the area they're going to fly through ... in theory all flight plans should be correct ..."

ETHNOGRAPHER: "is that plausible ..."

gap ...

ATCO: "it is ... but what it does is take the flexibility out the system ... the flow controller will do it all at the moment we can say Bristol looks busy, we can say right let's reroute ... I don't know if we'll retain that if we lose that we're in deep trouble ... it is possible I suppose that every flight plan will be correct there aren't actually that many major errors in flight plans, not as many as there used to be ... the wings are less useful than they used to be but its exactly because they're losing their skills ... I mean it'll happen with the ATCOs down the line somewhere and it comes back to the human or system problem ... whatever the problem is in your less than perfect system, a big black cloud or whatever, do you train them specifically for those situations, whereas now you don't have to 'cause with the expertise we've got we can cope ..."

That is, technological and procedural changes have already considerably reduced the level of error in Flight Plans with a concomitant reduction in the usefulness of the assistants or 'wings'. What cannot be known with any degree of certainty is what the consequences of errors, when and if they arise, will be in a situation where experience of dealing

with error has been attenuated. It is not as if the evaluation methods we are advocating unequivocally resolve problems of this nature, for we do not believe they can. They do, however, draw attention to their existence as problems and as such constitute resources for decision making which are not available from process-driven models.

CONCLUSIONS

Of course systems have to be evaluated, and we are not trying to suggest otherwise. We are confident that our informal procedures were a powerful, cost effective means of evaluation, given a constrained view of what that process should involve. Anecdotal and impressionistic evidence of specific cooperation activities and breakdowns in the use of specific systems in specific situations can contribute to knowledge about the general issues of collaborative working and the potential role of new systems. The procedures were cheap, relative both to more formal evaluations and to the costs of software development and were capable of informing design intuitions. The number, complexity and varying importance of the design decisions which are made during development of a computer system suggest there is a case for specific evaluation procedures targeted at particular kinds of problem, and in effect this is what we did with our small-scale evaluation studies.

However, we are aware that these procedures alone did not begin to address the issue of the situated character of work. We came slowly to the view that our discussions with ATCOs raised issues of very considerable importance for evaluation. Not least the uncertainties discussed above - articulated in the course of discussion between ATCOs and an ethnographer with considerable experience of their work - which concerned working with systems rather than systems in isolation. The salient issue, both in principle and in practice, should be how we determine whether systems can be said to 'work' or not.

The issue of 'validation' is very much the focus of our argument here. The fundamental problem lies in the combination of assumptions concerning purpose, timeliness, and method that typically (and our evaluation of MEAD is no exception) surround the evaluation process. More specifically, the idea that evaluation should occur late in the development process, should be concerned with machine or software functionality, and should concern itself with 'objective' results, sits strangely with the concern for the social organisation of work that characterises CSCW enquiry. We were led to question whether systems for use in cooperative work environments can indeed be evaluated for validity in isolation from the work. Significant doubt must be cast on the notion that we can 'validate' a system at a given point in the project if we accept that the use of systems is not completely determined by the functionalities designed into them. There may in principle be a vast range of reasons why usage may vary even within a single organisation. Systems put in place may initially fail because they do not resonate with existing practices. Training failures, the prevalence of 'fear and loathing', the breakdown of new organisational processes and so on may

all impinge on the speed with which systems become 'usable'. Equally, tried and trusted systems may begin to fail as changes in the environment begin to impact upon them.

To some extent, and we have many reservations, the relationship between the design of technologies and the design of work is addressed in the BPR literature. As Davenport [8] puts it:

The term process innovation encompasses the envisioning of new work strategies, the actual process design activity, and the implementation of change in all its complex technological, human, and organisational dimensions it implies a strong emphasis on how work is done within an organisation, in contrast to a product focus's emphasis on what.

That is, Information Technology design and organisational change are inextricably linked to one another. Systems innovation must associate with a set of assumptions about how 'improving work' is constituted. This shift in emphasis away from the system as technical artifact towards the system at work has a number of important implications for the evaluation process, implications which place ethnographic insights of the kind that the ATC project has utilised at its centre, at least if system 'validity' or 'acceptance' is the problem being addressed. In order to gain a purchase on 'validation', evaluation work will need to focus increasingly on the examination of the relationship between the system, existing work and organisational practices, and the re-design of both.

In CSCW, this places evaluation at the core of all the design activities which normally precede it, but which we argue is better conceived of as surrounding it. This is no easy matter. As Davenport points out, changes in work activity may take years to manifest, and the impact may not, even if apparent, be straightforwardly measurable. If true, it indicates that evaluation must be extended not only into the whole of the conventional design process, but also well into the system's useful life. That is to say, evaluation work will have to be conceived of not as something separate from other stages in the design process but as a necessary feature of all design work. Further, substantial re-conceptualisation of the notion of the 'system' and its boundaries will be necessary if we are to be serious in our attempts to evaluate use.

With the admitted benefit of hindsight, we came to feel that all of the ethnographic work undertaken during the course of the project can and should be regarded as ongoing evaluation, proving useful in various ways at different stages of the design of MEAD, and in principle in systems development at large. In other words, there are good reasons for regarding evaluation as extending well beyond the point of closure associated with both formal and informal techniques. Indeed we would say that it should be undertaken throughout the entire design process and well beyond. However, the relationship between these activities and more specifically located evaluations needs clarification.

Hence there is a need to devise mechanisms which both identify the specific evaluative purposes of ethnographic enquiry, its timeliness, and mechanisms to enable us to cope as well as possible with parallel activities.

We began our evaluation procedures with a considerable cynicism concerning the use of controlled experiments which, we felt, could not take the situated nature of work into account. Such experiments take place in a laboratory, involve ingeniously designed, but consequently artificial and de-contextualised tasks, and due to the usual constraints on obtaining subjects, are generally short term and specifically located [12]. We felt at that time that informal and opportunistic evaluation work might yield useful results without the need for the baggage carried by formal procedures. We came to the realisation that whilst our methods presented us with considerable information concerning 'incomplete' aspects of the system, they gave us no purchase on what 'completeness' would look like. The highly dynamic and variable nature of much co-operative work and the organisational context in which it takes place means that both formal and informal methods are subject to limitations imposed by the assumptions that inform them, and led us to conclude that there is an urgent need for triangulation [13] of the results, scope and limitations of evaluation activity in the context of CSCW enquiry. We would encourage other researchers to report their findings in different situations, so that a growing set of case histories can provide us with a means to assess the lessons, value and purpose of evaluation techniques which relate the system to its use.

ACKNOWLEDGEMENTS

Our thanks are due to our partners in the ATC project and colleagues at Lancaster, notably John Hughes, Dan Shapiro, Pete Sawyer, Ian Sommerville and Tom Rodden. We would also like to record our gratitude to Jacqui Forsyth for helping us turn the wildly idiosyncratic styles of three individuals into a single, grammatical and readable text.

REFERENCES

1. Bannon, L., Use, design and evaluation: steps towards an integration, in *Proceedings of the International Workshop on the Design of CSCW and Groupware systems*, to appear in D. Shapiro *et al.* (eds), *The Design of Computer Supported Cooperative Work and Groupware Systems*, Elsevier Science, Amsterdam, 1994.
2. Barnard, P., Applied cognitive psychology: research for human-computer interaction, in *Engineering the Human-Computer Interface*, A. Downton (ed), McGraw-Hill, Maidenhead, 1991, pp. 28-61.
3. Bentley, R., Supporting multi-user interface development for cooperative systems, unpublished PhD thesis, January 1994, available as a technical report, Computing Department, Lancaster University.
4. Bentley, R., Hughes, J. A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D. & Sommerville, I., Ethnographically-informed system design for air traffic control, in *Proceedings of CSCW'92*, Toronto, ACM Press, Nov. 1992, pp. 123-129.
5. Bentley, R., Rodden, T., Sawyer, P. & Sommerville, I., An architecture for tailoring cooperative multi-user displays, in *Proceedings of CSCW'92*, Toronto, ACM Press, Nov. 1992, pp. 187-194.
6. Bodker, S. & Gronbaek, K., Cooperative prototyping: users and designers in mutual activity, in *Int. J. of Man-Machine Studies*, 34, 1991, pp. 452-478.
7. Cool, C., Fish, R. S., Kraut, R. E., & Lowery, C. M., Iterative design of video communication systems. in *Proceedings of CSCW'92*, Toronto, ACM Press, Nov. 1992, pp. 25-32.
8. Davenport, T., *Process Innovation*, Ernst and Young, London, 1993.
9. Downton, A., Evaluation techniques for human-computer systems design, in *Engineering the Human-Computer Interface*, A. Downton (ed), McGraw-Hill, Maidenhead, 1991, pp. 325-355.
10. Feigenbaum, A. V., *Total Quality Control* (3rd edition), McGraw-Hill, New York, 1991.
11. Draper, S. & Watley, D., Practical methods for measuring the performance of human-computer interfaces, notes accompanying talk at the *JCI/HCI Summer School*, Queen Mary College, Aug. 1991.
12. Grudin, J., Why CSCW applications fail: problems in the design and evaluation of organisational interfaces, in *Proceedings of CSCW'88*, Portland, Sept. 1988, pp 85-93.
13. Hammersley, M. & Atkinson, P., *Ethnography Principles in Practice*, Routledge, London, 1983.
14. Harper, R., Hughes, J. A. & Shapiro, D., Harmonious working and CSCW: computer technology and air traffic control, in *Studies in Computer Supported Cooperative Work: Theory, Practice and Design*, J. Bowers and S. Benford (eds), North Holland, Amsterdam, 1991, pp 225-234.
15. Heath, C. & Luff, P., Collaborative activity and technological design: task coordination in London underground control rooms, in *Proceedings of ECSCW'91*, Amsterdam, 1991, pp. 65-80.
16. Hughes, J. A., Shapiro, D. & Randall, D., CSCW: discipline or paradigm?, in *Proceedings of ECSCW'91*, Amsterdam, 1991, pp. 309-323.
17. Hughes, J. A., Randall, D. & Shapiro, D., Faltering from ethnography to design, in *Proceedings of CSCW'92*, Toronto, ACM Press, Nov. 1992, pp. 115-122.
18. Karsenty, S., Landay, J. & Weikart, C., Inferring graphical constraints with Rockit, in *Proceedings of HCI'92*, A. Monk *et al.* (eds), York, 1992, pp. 137-153.
19. Landauer, T., Relations between cognitive psychology and computer systems design, in *Interfacing Thought*, J. M. Carroll (ed), Cambridge, MIT Press, 1987.
20. Nielsen, J., Usability engineering at a discount, in *Designing and Using Human Computer Interfaces and Knowledge Based Systems*, G. Smith & M. Salvendy (eds), Amsterdam, North-Holland, 1989, pp. 394-401.

21. Randall, D., Perspectives on evaluation: putting your money where your mouth is, paper to the *UK CSCW SIG*, DTI, London, September, 1993 (available from the author).
22. Randall, D., Hughes, J. A., & Shapiro, D., Systems development - the fourth dimension: perspectives on the social organisation of work, in *The Social Dimensions of Software Engineering*, P. Quintas (ed), Ellis Horwood, London and New York, 1993.
23. RSRE, Prototype Electronic Flight Strips Video, Malvern, 1991.
24. Self, J., Special Issue on Evaluation, *Journal of Artificial Intelligence and Education*, 4 (2/3), 1993.
25. Shapiro, D., Hughes, J. A., Randall, D. & Harper, R., Visual re-representation of database information: the flight strip in air traffic control, in M. J. Tauber *et al.* (eds), *Cognitive Aspects of Visual Languages and Visual Interfaces*, Elsevier Science, Amsterdam, 1994, pp 349-376.
26. Suchman, L., *Plans and Situated Actions*, Cambridge University Press, Cambridge, 1987.
27. Tatar, D. G., Foster, G. & Bobrow, D. G., Design for conversation: Lessons from Cognoter, in *Int. J. of Man Machine Studies*, 34(2), 1991, pp. 185-210.
28. Twidale, M. B., Redressing the balance: the advantages of informal evaluation techniques for Intelligent Learning Environments, in *J. of AI and Ed.*, 4 (2/3), 1993, pp. 155-178.
29. Twidale, M. B., Rodden, T. & Sommerville, I., The Designers' Notepad: supporting and understanding cooperative design, in *Proceedings of ECSCW'93*, Milan, 1993, pp. 93-108.
30. Twidale, M. B., Rodden, T. & Sommerville, I., Developing a tool to support collaborative dialogues and graphical representation of ideas, in *Collaborative Dialogue Technologies in Distance Learning*, F. Verdejo (ed), in press.
31. Vincent, D., *The Information Based Corporation*, Dow Jones Irwin, 1990.