Towards computer supported cooperative design

Thesis

UNRESTRICTED

Towards Computer Supported Cooperative Design

Susan Turner

BSc (Hons) Occupational Psychology, University of Wales

MSc Information Technology, Loughborough University

A thesis submitted for the degree of Doctor of Philosophy of the Open University

Department of Design and Innovation and Department of Computing

June 1999

13.04.02.

Acknowledgements

My thanks go to a number of people and institutions who have contributed to the completion of this thesis. First of all, to Phil, who has gracefully and uncomplainingly combined the roles of supportive spouse, stimulating co-worker and critical (but tactful) friend. To my supervisors, Prof. Nigel Cross of the Open University and Prof. David Benyon of Napier University, without whose continuing guidance, encouragement and unstinting support this work could not have proceeded. To those individuals at 'Metre' who were involved with the DUCK project for their time and cooperation, and to the 'Mallard' design team, who bravely agreed to have their design process subjected to video analysis. To the sponsors of the DUCK project, EPSRC and the DTI. And finally to MARI Computer Systems and the Department of Computing and Mathematics at the University of Northumbria, who supported my registration as a research student.

Abstract

This thesis first reviews the literature about the nature of cooperative design activity and its technological support. It is noted that knowledge of how designers work together in real-world settings is less than complete. Moreover, after over a decade of developments in Computer Supported Cooperative Work, the state-of-the-art in collaborative technology does not fully support such activities. Two substantial case studies are presented. The first draws upon fieldwork with designers at a large, distributed engineering design company, where a pilot study of collaborative technologies was carried out, focusing on the organisational context for such interventions and the reasons behind the qualified success of the experimental technology. In the light of the lack of use of synchronous tools in particular, a second case study was carried out. This was a complementary analysis of face-to-face co-working in a series of meetings held by a small design group. The results of both pieces of fieldwork are analysed in the context of existing studies of designers in both real-world and laboratory settings. This leads to the identification of a number of important characteristics of cooperative design, some newly identified, others confirming or extending the results of existing work. They include the identification of tension between traditional engineering design culture and the underlying assumptions of new technology; the intrinsic difficulties in sharing some types of design artefacts; and the way in which design entails an interweaving of individual and group activity, with consequences for resource exploitation, distributed cognition and workspace navigation in group sessions. The findings are integrated into an organising framework for cooperative design, with emphasis on the support of co-working designers distributed across multiple sites. Current technologies are reviewed against scenarios based on the framework and recommendations are made for further work.

Contents

Table of Figures

1 Introduction

This chapter

- introduces the work in this thesis, outlines the structure of the thesis and sets it in context;
- provides an overview of the methods used and some limitations.

What are the characteristics of cooperative design work and how can technology best support such activity?

The work now described originated in challenges faced by a major engineering design company. The company, 'Metre', was organised in a highly distributed fashion and senior personnel hoped that a move to computer supported cooperative work (hereafter CSCW) might overcome some current operating difficulties. Accordingly, Metre became a prime mover in a collaborative research project[1] intended to investigate the potential for CSCW. This thesis draws on part of this work, which is complemented by a subsequent study of small group design activity.

The structure of this thesis

This chapter sets the scene, outlines the scope for an investigation of cooperating designers at work and introduces methods for the study. Next, the thesis reviews the state of knowledge from reported studies of design groups and current technology to support them (Chapters 2 and 3). Fieldwork with designers at the engineering design company (Chapters 4 and 5) and a small software design house (Chapter 6) is then reported and discussed.

Chapter 7 integrates these finding with existing work in a descriptive framework for cooperative design, with particular emphasis on the support of co-working designers

[1] This was the EPRSC/DTI funded DUCK project (Designers as Users of Cooperative Knowledge). DUCK was part of the CSCW programme and was a collaborative venture between the engineering company ('Metre' in this report), the University of Paisley and MARI Computer Systems, where I was employed at the time the fieldwork was carried out.

who are distributed across multiple sites. Current technologies are reviewed in the light of this framework and finally, recommendations are made for further work.

1.1 The overall context for the study

As Robertson (1997), observes

> *...to date, longitudinal studies of non-experimental design projects are rare and the coverage of the range of design activities remains sparse.*

This was certainly so when the fieldwork to be discussed here was conceived and carried out, in 1994 and 1995. The point about 'non-experimental' projects is important because real life contexts differ from experimental ones in important ways: design projects are integrated into the stuff of everyday organisational life (as some of the literature summarised in Chapter 2 demonstrates); team members may be subject to the demands of other work; and project lifespans are weeks or months, which means that the creation and consultation of design records are vital. Evidence for all these points may be found in the results to be reported here.

It was also less than clear how cooperative technologies could best be exploited in the design context, and what the requirements were for a basic but effective toolset. The original aims of the current study, therefore, were:

1. To study the practice of cooperative design at organisational and small group levels in a real world engineering context and to derive an initial set of requirements on collaborative technology in this domain;
2. To refine this information with the results of a pilot study introducing collaborative technology to a group of designers;
3. To review the information gained from fieldwork in the context of the reported literature and existing theoretical and quasi-theoretical models;
4. To derive a framework organising characteristics relevant to the design and evaluation of technologies to support cooperative work in design;
5. To review current technologies using the framework.

The results of fieldwork towards the first two aims raised issues about the nature of design meetings, thus adding a new aim:

1a. To build on existing knowledge about activity in design meetings in their everyday context, and to further refine the cooperative design framework.

Security constraints and access to staff time meant that small group design work of this type could not be studied in depth in the original organisation: this part of the work used instead a small group of software designers at a research and development company. Figure 1.1. shows how the different elements of the study relate to one another.

Since the time the fieldwork was undertaken, studies such as those of Carstensen and Sørensen (1996); Olson *et al.* (1996); Olson and Teasley (1996); Pycock and Bowers (1996); Perry and Sanderson (1998), and Robertson's work (Robertson, 1996; Robertson, 1997) with software designers have added to knowledge about cooperative design in its natural setting. It remains the case, however that there is little solid evidence about the practical scope for technological support for collaborative engineering design – the subject of the first part of this study, and the complex and subtle interplay between design activity and design workspace in meetings – the main subject of the second part of the work.

This study therefore contributes to the relatively small number of field-based case studies of cooperative design in an engineering context. The analysis identifies and discusses the characteristics of such activity as practised in two organisational settings. These are chiefly the tension between technologically supported distributed working and cultural resistance to computer supported design, the way in which cooperative design activities are situated in both wider organisational practice and periods of individual work, the integration of the interests and perspectives of multiple stakeholders in the design process, the range and nature of workspaces used during design meetings and the features of the design tasks for which they are exploited. The framework provided at the end of this thesis draws together these findings with existing knowledge. This is used as the basis for a scenario-based review of existing technologies which illustrates the practical potential (and limitations) of the current state-of-the art.

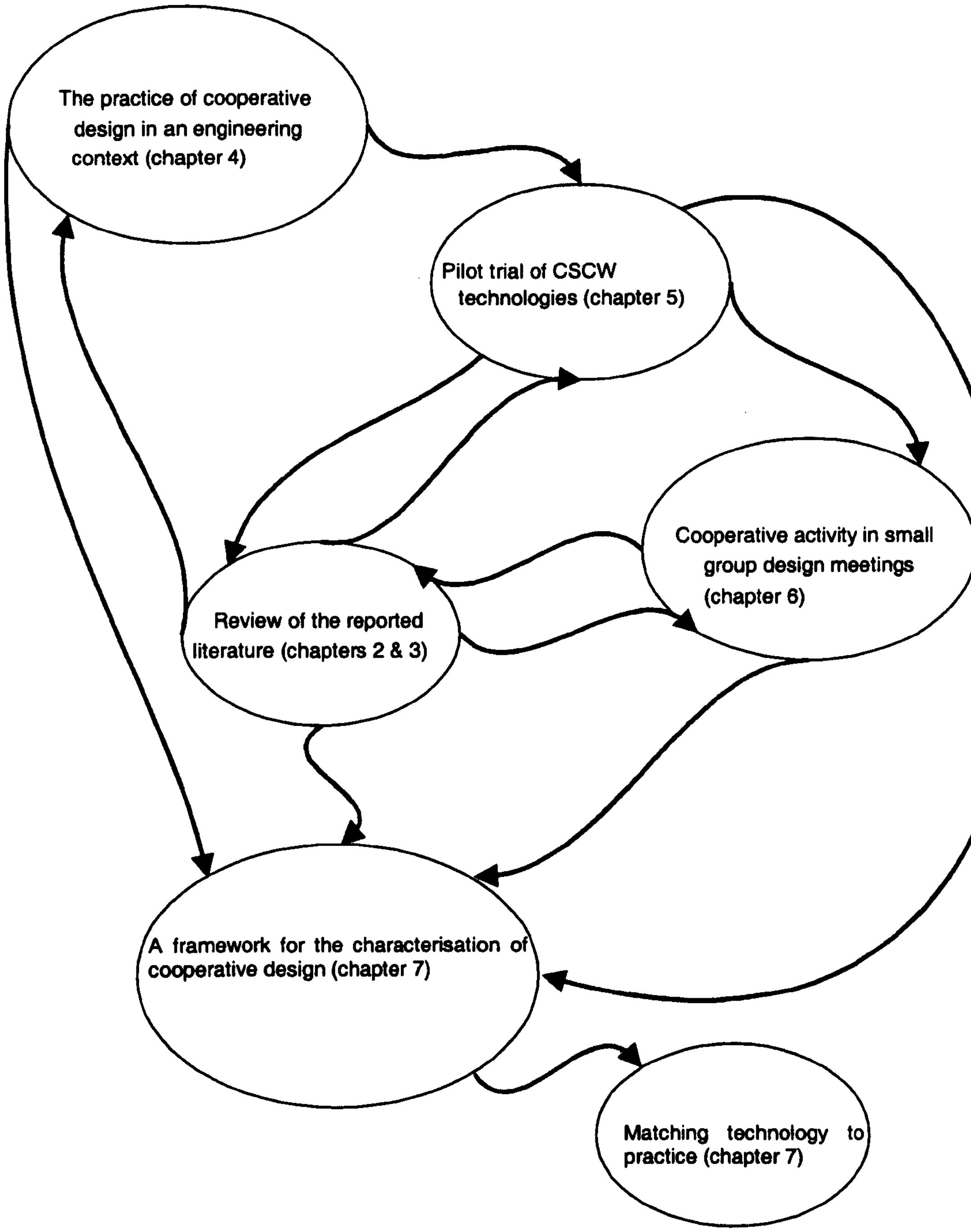


Figure 1.1 The elements of the current work

1.2 Methods overview

In the first part of the work (at the engineering organisation Metre) the approach was first to elicit requirements, through conventional interview and questionnaire methods supplemented by early discussion of potential socio-technical solutions. Potential organisational consequences were then modelled. The techniques used were adapted from the socio-technical approach (Olphert and Harker, 1994) of the ORDIT project (Esprit 2301). Once a pilot site for a trial of CSCW tools had been identified, interviews and limited observational data gathering methods were used to capture baseline data and monitor the use of the technology and users' perceptions of it. Security restrictions at Metre precluded the use of more situated techniques such as those deriving from ethnography, as described, for example, in Hughes *et al.* (1994) and the review in Plowman, Rogers and Ramage (1995).

For the more intensive data gathering from small group design meetings, a quasi-ethnographic approach was adopted. A specific analysis scheme for group design process is adapted from Olson *et al.* (1992a) to allow cross-study comparisons to be made, while analyses for gesture and workspace use derive from Tang (1991) and Radcliffe (1996).

Individual methods are discussed more fully in sections 4.1.1, 4.4.1, 5.4.3, 6.1.2, 6.1.4 and 6.6.3.

1.2.1 The choice of a case study approach

Both fieldwork elements of this thesis centre on the analysis of a case study in different design contexts. The use of case studies in CSCW and design literature is commonplace, as the many such studies cited in Chapter 2 attest, but it is worth a few words to justify the approach in this particular instance. Robson (1993), defines a case study as

> *"...a strategy for doing research which involves an empirical investigation of a particular contemporary phenomenon within its real life context using multiple sources of evidence."*

Case studies are particularly appropriate when the body of pre-existing knowledge is insufficient to support the formulation and testing of formal theories. In the design domain, while competing descriptive and prescriptive accounts of individual design exist, there are no established theories of design activity as practised by groups of designers. As for CSCW, the predominant research paradigm (aside from the development and testing of collaborative technologies) has been a case study approach. From this base grounded theories are gradually starting to emerge, *cf.* Orlikowski and Gash (1994); Blackler (1995); Henderson (1995); Nardi (1996); Lewis (1997). In both domains, there remains a basic need to expand our knowledge of just how people work together in different circumstances so that triangulations may be made between finding and solid foundations laid for theory generation.

However, the situated nature of a field case study brings its own limitations. (See Robson, 1993 for a full discussion of these issues.) Interpretation of results must take into account the local circumstances of the case study in order to be of more general application. In the Metre fieldwork, two caveats must be made. Firstly, the work reported here was undertaken under the aegis of a government-funded research project, and may therefore have had rather different outcomes to a venture run under fully commercial conditions. However, while Metre's effort was indeed externally supported, the extent of this support amounted only to around 30% of the company's costs. Thus, there was a substantial investment in the project, including most of a senior technical manager's time for many months. Further, Metre had intended to investigate the potential of CSCW even had the project not been funded. All in all, it can be considered that the existence of the project facilitated rather than stimulated the initiative and that the work was treated much as any normal technological venture. Secondly, when this work has been discussed with academic colleagues, surprise has sometimes been expressed as to the conservatism of engineering culture at Metre – discussed in Chapters 4 and 5 – and the related issue of very limited computing infrastructure. It is difficult to gauge the prevalence of similar cultures, but informal contacts suggest that such constellations of attitudes are not unknown elsewhere. Certainly many engineering organisations continue to have sparse information technology support – see section 5.4.1.

As for the Mallard software designers who form the subjects of the second field study, in general, their practice was much like that of other teams I have encountered in over 10 years in various sectors of the software industry, and as reported by others in the literature discussed in Chapter 2. More specific consequences of the methods for this part of the work are discussed in section 6.1.4.

Overall, the observations and conclusions here are necessarily grounded in the circumstances of two particular organisations at two particular points in their history, and standing alone they would be little more than a series of anecdotes. However, once placed in the context of the literature about to be introduced in Chapter 2, the results can be seen in their proper place in the web of case studies, theories and speculations which make up the current state of our knowledge of cooperative design.

2 Designers at work: the current state-of-the-art

This chapter:

- briefly considers the nature of design problems themselves and *prescriptive* methodologies for dealing with these problems;
- reviews *descriptive* reports of the problem solving strategies adopted by designers working individually and in groups;
- reviews studies of other aspects of design group working, both field studies and studies undertaken in controlled conditions;
- considers how the effects of computer aided design systems may influence the introduction of CSCW in design.

2.1 Design problem solving in theory

What special demands does design present for cooperative problem solving, and how do designers attempt to solve such problems? Design is often treated as a prototypical instance of a 'wicked problem' (Rittel and Webber, 1984): such problems are ill-structured (Simon, 1984) i.e. constraints, methods and goals are all under specified, and there is more than one 'right answer'. In such circumstances we tend to 'satisfice', to use Simon's classic term (Simon, 1957), rather than optimise a solution, making a reasonable compromise between conflicting constraints[2]. A good enough solution will do.

As Goel and Pirolli (1992) observe in their review of design problem space,

> *Design, of the sort we are interested in, is ill-structured in that the tasks involve underspecified goals and operators. The kinds of knowledge that may enter into a design solution are practically limitless... design inherently consists of the formulation of models of possible states of affairs in the world.*

Goel and Pirolli consider the differentiation of design from other problem domains, proposing a detailed list of design task environments and problem features, shown in the box overleaf. Their conclusions are validated by empirical work comparing design tasks

---

[2] Bucciarelli (1994) suggests another source of satisficing in design - reconciling the technical perspectives of different individuals.

in architecture, mechanical design and instructional design with non-design problem solving tasks. The list not only draws attention to the 'wicked' nature of design problems, but also highlights other core characteristics. Design is essentially a matter of the creation and manipulation of a series of models, resulting in a "complete description of the artefact that is to be made" (Cross, 1989). Such models may be represented as, and supported by, a range of intermediate artefacts in a variety of media. It is the characteristics and affordances of these artefacts which provide many of the challenges to the technological support of design.

The main domains for the current work are mechanical engineering design, treated explicitly by Goel and Pirolli, human factors design and software design, while product design provides a source of comparison data in Chapter 6. Human factors design – the design of artefacts such that they are readily usable by humans – fits the list provided with the exception of the 'feedback loop' identified as Goel and Pirolli's feature 'H'. Design activity in this sphere frequently takes advantage of prototyping, thus obtaining some feedback during the course of design itself. The activities of software design also exhibit a good correspondence, also diverging mainly in the occasional use of prototyping. In some cases also the use of advanced software engineering tools blurs the distinction between design, specification and the code itself. Product design shows a similarly close fit to the features listed. All-in-all, the current domains meet Goel and Pirolli's criteria closely enough to be considered as squarely design problems.

Features of design task environments (from Goel and Pirolli, 1992)

A. Distribution of information - *under specification of start state, goal state and transformations*

B. Nature of constraints - *natural laws do not determine design solutions*

C. Size and complexity of problems - *large and complex, spanning days to years*

D. Component parts - *structure does not determine decomposition, rather the practice and experience of the designer*

E. Inter-connectivity of parts - *not logically interconnected, but many contingent interconnections.*

F. Right and wrong answers - *do not exist, only better or worse ones*

G. Input/output - *input is requirements information, output is the artefact specification*

H. Feedback loop - *no feedback during problem solving, only after artefact has been built*

I. Costs of errors - *penalty for being wrong can be high*

J. Independent functioning of artifact [product] - *functions independently of the designer*

K. Distinction between specification and delivery - *distinction between specification and delivery and construction*

L. Temporal separation between specification and delivery - *temporal separation between specification and delivery or construction*

Features of design problem spaces

1. Problem structuring - *extensive problem structuring required before problem solving can start.*
2. Distinct problem solving phases - *preliminary design, refinement and detail design.*
3. Reversing direction of transformation function - *designer can transform the problem into one which already has a solution.*
4. Modularity/ decomposability - *decomposition into sparsely connected modules: some connections may be ignored.*
5. Incremental development of artifact - *interim ideas are incrementally developed & rarely discarded.*
6. Control structure - *limited commitment mode so as to allow development of components in multiple contexts.*
7. Making and propagating commitments - *designers have to make, record and propagate commitments.*
8. Personalized stopping rules and evaluation functions - *derived from experience.*
9. Predominance of memory retrieval and non-deductive inference - *deductive inference has only a minimal role.*
10. Constructing and manipulating models - *rather than operating on the real world.*
11. Abstraction hierarchies - *orthogonal abstraction hierarchies are constructed.*
12. Use of artificial symbol systems.

In response to these potentially chaotic problem characteristics a range of prescriptive structured methods have been proposed by design methodologists. A review of design methods is outwith the scope of this thesis, but some of the more influential structured methods include Pugh (1981) and Pahl and Beitz (1984) in engineering; SSADM (NCC, 1995), Yourdon Structured Method (Yourdon, 1989) and the object-oriented methods, for example Booch (1991), in software design. In brief, structured methods in whatever domain prescribe the analysis of problems into their component parts, the systematic consideration of alternative solutions to these sub-problems, the balanced development of detailed sub-solutions, synthesis into a coherent whole and finally evaluation. The overall approach is breadth-first, rather than the depth-first development of one or more components of the design. As we will see in the next section, actual practice does not always follow the prescribed structural approach.

2.2 Design problem solving in practice

2.2.1 Designers working alone

Descriptive studies of design problem solving have been reported over several decades, the domains studied ranging through town planning and architecture, engineering, product and graphic design to software design. (Note: the terminology used by authors cited in this section differs: structured approaches are also described as top-down, problem focused, systematic, or breadth-first, formally ill-structured strategies as bottom-up, solution focused[3], opportunistic[4] or depth-first. This is not to say that these terms are direct synonyms or antonyms, but their meaning is sufficiently close to permit comparisons to be made.) Many studies, for example Eastman (1970); Darke (1979) (both architecture); the studies in sundry domains reported in Lera (1983); Thomas and Carroll, (1979), Whitefield (1986, 1989) (all studies of design engineers) and Guindon's (1990) analysis of software engineers suggest that opportunistic, iterative and solution focused strategies are heavily exploited in early design problem solving. Indeed, Guindon argues that opportunistic decomposition is a highly appropriate strategy for the essential ill-structuredness of design, while top-down decomposition appears to be a special case for well-structured, familiar problems. Opportunism was also observed in mechanical engineers by Ullman, Dietterich and Stauffer (1988), albeit in the later stages of design, and by Davies (1991) as a subsidiary strategy in program design.

The notion of solution focus is further developed by Singley and Carroll (1996), who propose a taxonomy of design reasoning based on an introspective study of the development of an intelligent tutoring system. The reasoning methods proposed are:

---

[3] In the sense of generating or retrieving solutions and testing them against requirements early in the design process.
[4] A term originating with Hayes-Roth and Hayes-Roth's (1979) study of planning behaviour.

assessing the genre of an artefact; hill-climbing from a predecessor artefact; process modelling (i.e. modelling how a process works in the real world); scenario envisionment and formative evaluation. Singley and Carroll argue that the approaches are essentially complementary and in practice interleaved. However, scenario envisionment provides the essential bridge between analysis and synthesis in the generate-and-test process of design problem solving.

Other descriptive work suggests, at least for experts, a predominantly structured, systematic, top-down, breadth-first approach more akin to prescriptive models. The studies by Jeffries *et al.* (1981) and Adelson and Soloway (1985, 1986) of software engineers are in this group. However Jeffries *et al.* do note deviations from a breadth-first strategy in cases where a sub-problem appeared particularly important, had a known solution, or was very difficult.

Ball, Evans and Dennis (1994) and Ball and Ormerod (1995), report an observational study of engineers engaged on real world design projects over several months. Their results showed a highly systematic approach to design, but one adopting a top-down depth-, rather than breadth-, first approach. There was substantial evidence of satisficing, but only a small amount of opportunism was observed. Ball and Ormerod argue that that in some accounts opportunism has been under-defined and thus confused with a structured, but depth-first strategy. They put forward an integrated account of design: expert designers adopt a largely structured, breadth-first approach, supplemented by a small amount of depth-first and genuinely opportunistic behaviour; novices adopt a structured, but predominantly depth-first approach. In a related argument, Dorst (1996) proposes that while many design problems might appear ill-structured at first glance, designers do not treat them as such. In practical design tasks, problem space is constrained by past experience and in many cases by the relatively small number of realistic solutions. Finally, Davies and Castell (1992) suggest that the differing results obtained from observational studies of design may be explained by the research methodology adopted. Those studies which require the generation of verbal protocols may be subject to confounding effects. The narrative conventions of discourse may impose a superficial structure on less ordered activities, while conversely verbalising may make the cognitive load of following an ordered plan impossible, leaving designers to fall back on more data driven tactics. They also suggest that the small number of cases involved in most studies of this type render generalisation difficult.

Overall, the observations made by Ball and Ormerod (1995) and Davies and Castell (1992) about definitions and methods respectively are highly apposite. Combined with the diversity of problems and domains reported, it is extremely difficult to form a coherent model of practical design problem solving. It is only when researchers draw on identical source material (as in Cross, Christiaans and Dorst, 1996a), report the problem solving context in sufficient detail to support replication, or repeat analysis methods in different contexts that soundly based generalisations can be made. For these reasons, the small group analysis in Chapter 6 draws on heavily on existing methods.

The findings just discussed in this section are summarised in the figure 2.1, to be found in section 2.2 together with the findings for groups. As far as it is possible to draw any general conclusions, the current balance of evidence suggests that while design problems may be wicked, in practice the problem space may be smaller than it appears. Further, experienced designers may exploit opportunistic and solution-focused tactics within a generally structured framework. Several of the prescriptive models acknowledge this, for example March (1984) and Cross' strategic approach, which emphasises the need to suit design techniques to the nature of the problem and the skills of the designer (Cross, 1989).

The influence of computer aided design

Over the last two decades computer-aided design (CAD) systems have become commonplace in the design professions. From their early incarnations as special purpose analysis tools and two-dimensional line drawing packages to the current state-of-the-art plastic modellers and representations of virtual space, they have been considered both a hindrance and a help to designers. A brief consideration of their role is necessary because of their impact on both the design process itself and the attitudes of designers to other new technologies. Both topics merit a thesis in themselves and the topic is only sparingly considered here.

Somewhat surprisingly, while early studies exist (for example Whitefield, 1986), the area appears to have attracted relatively little attention in the more recent literature. There are however a number of individual case studies such as that reported by Yetton, Johnson and Craig (1994) – this latter study is considered in detail in section 5.4.5 – and Joshi and Lauer (1998). In general, early work such as Whitefield's suggests that the use of CAD systems imposes an additional cognitive load on the designer, although whether this is still the case now that such systems are employed from the earliest stages of design training may be doubtful. The case study evidence as to impact on designers is equivocal. Joshi and Lauer, for example, find a favourable impact for automotive parts designers from the introduction of CAD systems, while Yetton, Johnson and Craig report a degree of resistance to CAD in their study of an architectural design practice. This was rooted in concerns about constraints on creativity, and anecdotal evidence[5] to the effect that designers are able to identify the CAD system employed from the general style of a finished design tends to support this. It is very likely that much depends on the context of the technological intervention, and it is for this reason that detailed case studies such as the current work remain necessary. As to the design process itself, more than one older design engineer at Metre (the organisation featured in Chapters 4 and 5) remarked that younger colleagues trained on computer aided design and analysis systems would pursue the first promising idea and rely on the computer to find the problems. Of course, how far this is part of a general tendency among novices to adopt a depth-first strategy as contrasted to the influence of CAD itself remains a matter of speculation.

2.2.2 Problem solving in design groups

So far, we have discussed the approach to design problem solving in the context of the single designer. What happens when designers work together?

Stults (1988), referenced by Lu and Mantei (1991), notes that groups of architects formed an initial design concept, then summarised the issues or sub-tasks underlying the project. More detail is provided in Bucciarelli's (1994) descriptive account of engineering designers attempting to use a structured method in a design meeting. The approach chosen is Pugh's (1981) concept selection method, which prescribes steps for the collective evaluation of candidate design options against criteria established by the design team. The team attempt to adopt the method, championed by the team leader, but

---

[5] David Benyon, Napier University, personal communication.

the attempt fails and no conclusions are reached. Bucciarelli rejects the simple explanation that performance specifications and criteria were insufficiently defined to propose that their interpretation varied among group members. This interpretation is a strand in the evidence for Bucciarelli's larger thesis, that design is a socially co-constructed process.

Olson *et al.* (1992a) and Olson *et al.* (1996) report a field study of groups of software designers[6] in design meetings. The Olsons identified evidence of structure at the micro-level – design issues were generally followed by discussion and evaluation of alternative solutions. They do not, however comment on how far this was part of a larger scale structured approach. A less well-regulated software design process is related by Potts and Catledge (1996), drawing on an analysis of over 40 hours of meeting time. They observe a conflict between the desire to make progress (and consequently to avoid abandoning work already done) and the pressure to take into account new information which would lead to a higher quality result. Dwarakanath and Blessing (1996), among other analysts in the collection presented by Cross, Christiaans and Dorst (1996a), analyse product designers at work on a controlled 'laboratory' task. They note that the group follows a more structured, breadth-first, problem-oriented approach, whereas an individual engaged with the same task adopts more solution-focused tactics. Dwarakanath and Blessing surmise that in part this may be because the group is able to step back from the pursuit of individual solutions. The conclusion is supported by Radcliffe (1996) in his discussion of the same protocols, which notes explicit attention to a structured design process. Finally, Clarke *et al.* (1996) found evidence of both solution- and problem- focus in one hour design tasks undertaken by pairs of designers supported by a video link and shared drawing surface. Designers moved freely between the two strategies until the problem in hand had been resolved.

Again, it can be seen that in practice, groups of designers adopt a variety of problem solving strategies – the findings are summarised in figure 2.1 below. As in the case of single designers, however, the diversity of research approaches precludes more

[6] To be discussed at greater length in Chapter 6.

conclusive statements. The issue of group problem solving structure is considered further in Chapter 6, in the context of the process adopted by the designers in this work.

	Authors	*Domain*	*Main process findings*
individuals	Eastman (1970)	architecture	opportunistic, iterative and solution focused
individuals	Darke (1979)	architecture	opportunistic, iterative and solution focused
individuals	Whitefield (1986, 1989)	engineering	opportunistic, iterative and solution focused
individuals	Ball, Evans & Dennis (1994), Ball & Ormerod (1995)	engineering	top-down depth-first, substantial evidence of satisficing, only a small amount of opportunism (more likely to be found in novices than experts)
individuals	Ullman, Dietterich & Stauffer (1988)	engineering	opportunistic in later stages
individuals	Dwarakanath & Blessing (1996)	product design	predominantly solution focused
individuals	Guindon (1990)	software engineering	opportunistic, iterative and solution focused
individuals	Davies (1991)	software engineering	opportunism as a subsidiary strategy
individuals	Singley & Carroll (1996)	software engineering	interleaving of problem and solution focus coupled with scenario envisionment
individuals	Jeffries *et al.* (1981)	software engineering	predominantly structured, systematic, top-down, breadth-first except where sub-problems particularly important, had known solution, or very difficult.
individuals	Adelson & Soloway (1985, 1986)	software engineering	predominantly structured, systematic, top-down, breadth-first
individuals	Lera (1983)	various	opportunistic, iterative and solution focused
individuals	Thomas & Carroll (1978)	various	opportunistic, iterative and solution focused
groups or pairs	Stults (1988)	architecture	formation of initial design concept, then summary of issues/sub-tasks
groups or pairs	Bucciarelli (1994)	engineering	failed attempt to use Pugh's concept selection method
groups or pairs	Dwarakanath & Blessing (1996)	product design	predominantly structured, breadth-first, problem-focused
groups or pairs	Clarke *et al.* (1996)	product design	interleaved problem and solution focus
groups or pairs	Olson *et al.* (1992a), Olson *et al.* (1996)	software engineering	evidence of structure in discussion, no comment on larger scale strategy
groups or pairs	Potts & Catledge (1996)	software engineering	conflict between making progress and absorbing new information

Figure 2.1 Main observations on individual and group design process, organised by domain

A note about design rationale

The nature of group problem solving matters when the support of design groups is considered, because of potential constraints on the usefulness of design rationale tools. Design rationale – at its simplest the reasoning behind design decisions - has been the subject of much attention. A useful collection of perspectives on design rationale is to be found in Moran and Carroll (1996); there is also a full discussion of the issues in Buckingham-Shum and Hammond (1994), while the report by Karsenty (1996) describes the consultation of design rationale documents. It is usually argued that a focus on structured rationale is an important process aid to the making of well-founded decisions and to retrospective understanding of those decisions. Accordingly, methods have been developed to support designers in exploration and documentation and provide taxonomies of design reasoning, some of which are supported by software tools[7]. Among the most prominent are IBIS and its variants rIBIS and gIBIS (Yakemovic and Conklin, 1990; Conklin and Burgess-Yakemovic, 1991) and QOC (MacLean *et al.*, 1991). Terminology and scope vary from author to author, but all include some variant of the following main categories:

Terminology & common variants	Definition
Issue (question)	The part of the design space requiring a decision.
Alternative (option)	A way of solving the problem presented by an issue.
Criteria (argument)	Reasons for or against an alternative.

The first aim of most design rationale methods is the enhancement and documentation of design decision making *per se*. Evidence is equivocal as to how far they succeed in this, and to what degree they help or hinder the design process itself. If design discussions, because ill-structured, neglect useful alternatives or thorough evaluation, then design rationale tools could foster a more systematic, breadth first approach. However, the structure imposed by the design rationale tools could equally constrain potentially useful opportunistic discussion. The Olsons' (1992, 1996) evidence of structure suggests some scope for such support, in that a certain amount of intrinsic structure is observed in the design meetings studied. A comment from Conklin and Yakemovic about why the capture of rationale is problematic is of interest here, given evidence from this and other studies as to the embedded nature of design tasks.

> *"...in general, design deliberation is not a distinct activity separated from the rest of the process; it is part of formal and informal meetings, phone calls, chance encounters, lunch discussions and so on, and its progress is intimately bound to the progress of these other aspects of work."*

Design rationale taxonomies have also been adopted by researchers in the domain of design process and one variant is employed in the analysis reported in chapter 6.

7 See also discussion of coordination technologies in section 3.4. below,

2.3 Beyond design process – wider studies of design groups

This section reviews studies which have considered aspects of design group working beyond the issues of structure and process just discussed. The use of CSCW technologies in design is discussed in Chapter 3.

2.3.1 Field studies

Cooperation and communication in the design team and beyond

In an early example of the genre, Curtis, Krasner and Iscoe (1988) undertook a field investigation of software design practice, using retrospective interview data from 17 large projects. They identified a number of issues relating to group working. Expert designers tended to dominate consensus forming; teams composed of members from different technical areas more often utilised a breadth-first, rather than depth-first, approach. As for communications in large projects, meetings and common representation techniques were seen to be essential. This finding is echoed in the work of Reeves and Shipman (1992), who collected videotaped sessions of network designers, demonstrating how the network diagrams themselves acted as a communications medium. Again in the software engineering domain, Flor and Hutchins (1991) describe a case study of software engineers designing a small enhancement to a program. Activities are analysed from the perspective of distributed cognition i.e. the treatment of the external structures exchanged by the (human) participating agents as the 'mental state' of the distributed cognitive system. It is concluded that the cognitive properties of the distributed system are different from those of the individual programmer. (For an extended treatment of distributed cognition, see Hutchins (1995). As will be demonstrated in section 6.3.1, the software engineers analysed in the current study exhibit just such features of distributed cognition.)

Sharrock and Anderson (1996) provide an ethnographically informed account of the work of a team designing a new photocopier component. The paper notes how the process, decisions and tactics adopted by the designers were heavily influenced by organisational context and constraints. Issues capable of resolution purely in the context of the narrow design problem were relatively rare. Their conclusions add specific detail for the design domain to the work of Ancona and Caldwell (1990) who analyse the internal and external communication patterns of groups developing new products. They observe that these change over the life of the project: external information, for example, being vital in the early concept development stages but a source of distraction later in product development. The embedded nature of software design is further emphasised in the work of Potts and Catledge (1996). The authors carried out a detailed field study of a large industrial software design project at the conceptual design stage. Wider organisational processes had a considerable impact on the work of the project team delaying convergence on a common vision of the task and it was also noted that the team was 'inherently forgetful', raising implications for collaborative support tools or at the least improvements in meeting process.

Other studies have focused on the complex, interdisciplinary nature of design activity and its place within the larger context of the enterprise. The variety of tasks that comprise everyday design activity and their embedding in the larger business process is noted in the work of Pycock and Bowers (1996) in the fashion industry. Similarly, Carstensen and Sørensen (1996) observe the complex nature of large design projects in a field study of manufacturing design. Medway (1996) provides a close observation of architectural process, in the context of the exploitation of drawings in communication. The description of everyday tasks and the interactions and documentation entailed is worth quoting at length.

> *"One of these 'strands' might involve work at the drafting table in the office, including drawing, writing, telephoning, sending and receiving faxes, consulting documents and talking to colleagues, superiors, suppliers, etc. who come to visit; also included might be meetings elsewhere in the office, and site meetings with contractors, consultants, clients and public officials such as fire officers. Documents generated may range from a lengthy set of specifications, to Post-it notes stuck on the drafting table, to scribbles made by the architect in a notebook while standing on a roof inspecting the chimney."*

Perry and Sanderson (1998) report an ethnographically informed investigation of team design practice, studying mechanical and building engineers. Again, (see discussion of Sharrock and Anderson and Ancona and Caldwell above) it is concluded that interactions with others outside the immediate team were an integral part of design development. It is further observed that many interactions were informal in nature and that design development was intimately bound up with a variety of artefacts. The range of artefacts identified by Perry and Sanderson goes beyond the prototypical "public representation of the design at a given moment in time" - drawings, sketches, and prototypes. Also crucial for design development is the knowledge embodied in artefacts such as specifications of construction techniques, meeting minutes and Gantt charts. (The role of artefacts in design development is a central theme of the second part of the current study.) A further analysis of the same mechanical engineering design data is provided by Sanderson (1998), who proposes an organising framework of seven interconnected dimensions of joint technology design: management discourse and strategies; group dynamics; politics of design; social organisation of the design work and organisational context; role of artefacts; actor networks; and ambiguity and uncertainty. Sanderson argues that while the first six dimensions have been noted in the literature, the seventh - ambiguity and uncertainty - is novel.

Quantifying the extent of cooperation

As with other aspects of cooperative design, where evidence does exist about the extent of cooperation it is somewhat equivocal. Murray (1993), using data from an observational study, analyses patterns of individual and collaborative work in a graphic design practice. It is noted that although the designers in question share an office and a culture, and are aware of each other's activities through continual co-presence, true collaborative work is rare. Murray concludes that there is little potential for collaborative technology in such a group[8], and suggests that other apparent examples of team working may share similar characteristics. In work carried out elsewhere with teams of software design engineers (Turner and Turner, 1997), we found that only 30% of working time was actually spent in cooperative tasks. What appeared initially to be a highly collaborative activity was on closer inspection revealed as a set of highly co-ordinated but independent tasks. Even more strikingly, Harper and Carter's (1994) investigation of two apparently collaborating groups of engineers and architects in a design project concludes that close contact between the two disciplines in this instance was not only unhelpful but at times counterproductive. An indirect piece of supporting evidence is provided by Sonnentag

---

[8]However, there are reportedly successful instances of supporting co-presence over video links, for example Pagani and Mackay (1993).

(1996), who reports that only 31.4% of a sample of professional software designers recommended "cooperation with colleagues" as a useful design strategy.

Findings supporting close cooperation include Powrie[9] and Siemieniuch's (1990) study of the mechanisms of design in the automotive industry. They note close collaboration between different groups of specialists, e.g. design, production and quality engineers. The nature of the industry meant that these groups were often located at different sites, and a strong need was identified for the support of cooperative working through collaborative technology. Likewise, in the field of industrial product design, Leiva-Lobos *et al.* (1997) observe substantial and productive collaborative activity within creative design teams and between designers and customers.

Design meetings

More intensive fieldwork has examined small group design meetings in detail. The most exhaustive of these is that conducted by Olson and colleagues (Olson *et al.*, 1992, 1996). The Olsons make an in-depth analysis of 10 software design meetings from four projects in two organisations, examining the content of discussions from the perspectives of structure, topic transitions and design rationale (see below). Striking cross-project and cross-organisation similarities in both meeting activities and the sequential organisation of these activities are observed. Overall, most time was spent in discussion of design or design related issues, with many rapid transitions between the generation of design alternatives and their evaluation. Meeting and project management accounted for the remainder of the time, and an analysis of transitions between activities showed clusters of design and management discussions. Finally, as already noted in section 2.1, a degree of structure was observed. The Olsons' work is a major source of comparison data for the work described in Chapter 6 of this thesis, where it is discussed in more detail. Potts and Catledge's (1996) study of software designers also provides data on meeting content, though at a less detailed level of analysis. It is observed that around half the issues discussed were concerned with process rather than the design itself.

---

[9] The author under a previous name.

Robertson (1997) makes an in-depth analysis of a group of software designers in a series of meetings from a rather different perspective. The study concentrates on "embodied actions" (gestures and other, larger grain actions relating to artefacts in the design workspace) and proposes a taxonomy of such actions. It is argued that the public availability of embodied actions is core to successful communication in a shared physical workspace.

The main findings from studies of wider design issues reviewed in this section are brought together in the table in section 2.3.3.

2.3.2 Studies under controlled conditions

A number of researchers have taken advantage of the control and potential for comparison afforded by laboratory conditions and well-defined, limited tasks to study design process and design communication in detail. The approach taken is common to the studies described below. Small groups of designers, generally 3 or 4, are assigned group tasks which are intended to be completed in a session of an hour or two. Designers have a normal meeting room at their disposal, typically including the shared workspaces of a whiteboard, flipchart or large sheets of drawing paper. The sessions are videotaped and a transcript made of the naturally occurring verbal protocol of the group. Videotape transcript and often the artefacts produced are then analysed to elucidate particular aspects of the design process.

Tang (1989, 1991) presents a descriptive study of the shared workspace activity, including listing, drawing and gesturing, of groups working on conceptual design tasks for a user interface. In particular:

> gesture was found to be a valuable aid to storing information (e.g. by imitating an original explanatory gesture, expressing ideas (e.g. by explaining how a control would work) and mediating interaction;
>
> the *process* of creating artefacts in the workspace held meaning which was important to and understanding of the *product*;
>
> designers occasionally moved concepts from private to public availability, first working on sketches in proximal space, then introducing them to the group.

A set of design implications for tools to support shared workspace activity is derived. Bekker (1995) studied two sets of small groups of designers, product design students working on a user interface task and MBA students designing an Automated Post Office. The results echo Tang's in that physical actions are shown to play a part in design communication, even when they are not related to representations. They also illustrate how different types of gesture were used for different design and management activities, using the Olsons' (1992) categorisation scheme for design activities. Both Tang's and Bekker's work are used as sources of comparison data in Chapter 6.

A major collection of analyses of designers at work is provided in the volume edited by Cross, Christiaans and Dorst (1996a). The studies discuss video and verbal protocol data from groups and individuals at work on a 'laboratory' task to design an industrial product – a device to allow backpacks to be carried on mountain bikes. The data were analysed through a number of approaches, and thus provide a rare opportunity to compare different perspectives on identical source material. The areas of focus include design strategies and process (e.g. Popovic, 1996, who notes the interaction of team and individual work); group versus individual behaviour (e.g. Dwarakanath and Blessing, 1996, who observe, *inter alia*, that the group adopts a stronger problem focus and Günther, Frankenberger and Auer, 1996, who reinforce this finding and comment on individual differences within the team) and research methodology (e.g. Dorst, 1996, who speculates about the influence of the type of design problem posed). Of particular relevance for this work are Harrison and Minneman (1996), who discuss the role of 3D artefacts both as a rich information source in themselves and as elements of interactions and Mazijoglou, Scrivener and Clark (1996), who note the use of multiple design workspaces, defined as both the available interpersonal communication channels and the physical media such as whiteboards and flipcharts. Data from the industrial design exercise and the authors' other work with technologically mediated distributed design are contrasted. Other studies in the collection stress the importance of the social and interpersonal processes which underpin collaborative design including Cross and Clayburn-Cross (1996), who note social interactions eliciting support for an individual's preferred solution and managing potential conflict and Brereton *et al.* (1996) who also observe a substantial degree of negotiation. In a later analysis of the data, Cross (1997) observes how the flexibility facilitated by the group's interpersonal process permits a "relatively successful" conclusion, concluding that any collaborative technology must support just such flexibility.

Attention to the social aspects of design teamwork is beginning to emerge elsewhere in the literature. Cahour and Pemberton (1998) distinguish task-oriented interaction from those interpersonal interactions which support the social relationship in their study of student designers completing a small design exercise over the telephone. Some trend towards "safeguarding the interpersonal at the possible expense of the task" is identified – and this is also found in respect of criticism in the study by Falzon, Darses-de-Montmolin and Béguin (1996) of novice and expert network designers – but the authors note a range of cooperative styles, suggesting also that differences may be found between cultures and between novice designers and professionals. Finally, several studies examine the nature of cooperation in small group working. The review in Kvan (1997) suggests that what appears to be closely-coupled mutual activity can more accurately be treated as micro-episodes of individual work punctuated by bursts of negotiation and evaluation. Similarly, Maher, Cicognani and Simoff (1997) observe three modes of working[10]: mutual collaboration, or close co-working; exclusive collaboration, where participants work separately, occasionally negotiating or asking advice; and dictator collaboration, where one person leads the process. In the cases studied exclusive collaboration proved the most effective strategy.

In summary

From field studies of designers, existing work suggests that:

- the evidence about the degree of structure in group design problem solving is equivocal;
- design communication in meetings may be treated as an instance of distributed cognition; this is in part mediated through "embodied actions" which exploit a wide range of workplace artefacts;

---

[10] These designers were supported by collaborative technology, and thus the study properly belongs in the next chapter, but is treated here for convenience.

- designers spend much of their time working with others, but some of this collaboration may be more apparent than real;
- in many projects, both formal and informal communication extends beyond the design team;
- decisions are influenced by organisational factors beyond the design problem itself.

From studies of designers working under controlled conditions, it may be further observed that:

- gesture appears to be salient in conveying meaning and in mediating interpersonal interaction;
- the act of drawing itself may convey meaning;
- representations may move from private to public space;
- existing artefacts are a rich source of information and an integral part of communication of new concepts;
- even in peer groups, designers adopt different task-related and social roles; interpersonal relations may be safeguarded at the expense of design quality;
- the nature of collaboration itself is variable.

2.3.3 Beyond design process – the state-of-the-art in studies of design groups

As will be abundantly clear from the above accounts and the summary table below, knowledge of how designers work together is as yet very diffuse. Just as in studies of design process, it is impossible to draw general conclusions because of the fragmentary and varied nature of the information so far available. We are still in the stage of mapping out the terrain. The present pair of case studies aims to define a little more of the territory, while the framework proposed in Chapter 7 may serve as a guide to further exploration.

	Authors	Domain	Main findings
Field studies	Medway (1996)	architecture	Interaction of different types of tasks and artefacts.
	Harper & Carter (1994)	architecture & building engineering	Facilitation of close contact counter-productive.
	Powrie & Siemieniuch (1990)	automotive engineering	Close collaboration between different specialisms.
	Pycock & Bowers (1996)	fashion design	Wide variety of tasks embedded in larger business process.
	Murray (1993)	graphic design	Little true collaborative work.
	Carstensen & Sørensen (1996)	manufacturing engineering	Complex nature of large projects.
	Perry & Sanderson (1998)	mechanical and building engineering	Interactions with others outside the team, informal nature of interactions, integration of development and multiple artefact use.
	Reeves & Shipman (1992)	network engineering	Importance of network diagrams as a communication medium.
	Ancona & Caldwell (1990)	product design	External information vital in early stages but distracting later.
	Leiva-Lobos *et al.* (1997)	product design	Substantial collaborative activity within teams and with customers.
	Curtis, Krasner & Iscoe (1988)	s/ware engineering	Expert designers dominated consensus forming, multi-disciplinary teams more often utilised breadth-first approach, meetings and common representations essential.
	Flor & Hutchins (1991)	s/ware engineering	Design as distributed cognition.
	Olson *et al.*, 1992, 1996	s/ware engineering	Substantial cross-organisation similarities in activities and structure. Most time spent in design rather than process issues.
	Potts & Catledge (1996)	s/ware engineering	About equal time spent on design and process issues.
	Potts & Catledge (1996).	s/ware engineering	Influence of wider organisational issues, forgetfulness in the team.
	Robertson (1997)	s/ware engineering	Meaning conveyed in "embodied actions".
	Turner & Turner (1997)	s/ware engineering	Little true collaborative work.
	Sharrock & Anderson (1996)	s/ware & hardware engineering	Influence of wider organisational issues.
Controlled studies	Maher, Cicognani & Simoff (1997)	architecture	Identification of three distinct styles of collaboration.
	Falzon, Darses-de-Montmolin & Béguin (1996)	network engineering	Tendency to safeguard personal relationships, differences in cooperative style between novices and professionals.
	Cahour & Pemberton (1998)	product design	Distinction between task-oriented and interpersonal interactions, tendency to safeguard personal relationships.
	Cross, Christiaans & Dorst (1996a).	product design	Collection of analyses including design strategies, design process, group versus individual behaviour, information management, use of artefacts and workspace, social processes and research methodology.
	Bekker (1995)	user interface engineering	Importance of gesture in design communication, different types of gesture for design and management activities.
	Tang (1989, 1991)	user interface engineering	Gesture valuable in storing information, process of creating artefacts helps to convey meaning, use of public and private space.

Figure 2.2. Main findings from studies of group working (excluding design process issues), organised by domain

3 CSCW technologies for cooperative design

This chapter:

- outlines the design, functionality and behaviour of technologies relevant to the practice of cooperative design;
- supplements this account by reports of the use of such technologies in this domain.

Design groups have been supported by information technology for as long as the telephone, the fax machine and shared file systems have existed. This chapter is concerned with technologies specifically designed to support the process of cooperative working, often termed groupware. It sets the scene for the pilot technology implementation described in Chapter 5, and for the discussion in Chapter 7.

The cooperative work to be supported may be synchronous, as in the case of a design meeting, or asynchronous, as in the passing of a design document amongst team members for review. The technologies themselves may be designed for generic cooperative activity, or be specifically intended for the design domain. The classification used in this chapter is slightly adapted from Grudin and Poltrock (1997): technologies are organised according to whether they *primarily* support communication, shared information space, shared virtual space or coordination tasks, or some combination of these. The following sections briefly describe the technology in question and discuss reports of use in practice. It will be seen that the technologies reviewed span the decade from the early 1990s until the present. Thus, some examples have a historical cast. This is for several reasons. In some cases the tools are particularly representative of the functionality concerned, others display features not found in later developments, and more generally the review provides the technological context for the fieldwork undertaken in 1994/5 and reported in Chapters 4 and 5.

Note. This section does not contain explicit coverage of two technologies that might have been expected: multimedia tools and gestural devices. Much collaborative technology has multimedia aspects and these are covered in the discussion of the different types of tools which follow. And while gesture plays a major role in the analysis of small group design work in chapter 6, gestural technologies support input in human-machine interaction, rather than the human-human communication which is the subject of that part of the current work. Of course well-designed gestural input, in common with any other improvement in ease of interaction may facilitate the acceptance of multi-user as well as single user tools.

3.1 Technologies that support communication

Audio conferences are well established, but are increasingly carried over the Internet and PCs rather than the telephone system. Video conferences have been used by large organisations for more than 20 years, at first limited to special purpose suites used for formal meetings, and now available through the video and audio capacities of the desktop PC. Trials of collaborative tasks supported by video have generally shown that although task performance is not enhanced, a greater sense of teamwork and group identity is reported than with audio alone. Among other studies, examples include Egido (1990); Gale (1991); Tang and Isaacs (1993) and Newlands, Anderson and Mullin (1996). Despite this, and now ready availability – desktop videoconferencing has yet to be widely adopted as an everyday means of communication. Reasons for this have been ascribed to the poor image quality of systems running over limited bandwidth and the lack of availability of many visual cues for communication. Fussell and Benimoff (1995) provide a detailed account of the importance of visual information in 'perspective taking', the assessment of others' perspectives, in forming new working relationships. Accurate perspective taking depends on "...communicators' abilities to assess each others' work and social category memberships and to draw correct inferences from these memberships". Fussell and Benimoff observe that video images in desktop systems may be too small to convey the necessary information about salient features such as clothing, gesture and physical context. Similar issues are discussed in Heath and Luff (1992).

The basic desktop technology has been the subject of a number of attempts to bring communication closer to face-to-face co-working. Sellen's Hydra system (Sellen and Buxton, 1992), provides gaze awareness by representing remote participants as audio-visual units arranged around a table. A variant of this approach has been developed for the MAJIC system (Ichikawa *et al.*, 1995), where eye contact can be made with life size video images of other participants and gestures and movements clearly seen. Other researchers have explored the potential of informal video 'glances' into other offices, as in Montage (Tang, Isaacs and Rua, 1994): this has met with mixed success. Still other research laboratories have attempted virtual co-location by adding continuous video links between offices, or continuous availability of views of selected offices and common spaces. In a classic early study, whose findings are repeated in many later reports. Abel (1990) discusses a video wall used to link two geographically-distributed research labs. The system is described as just about adequate for creating a joint sense of place and culture. It worked well for sustaining relationships, but less well for making new contacts and for negotiations of sensitive points. Portholes (Dourish and Bly, 1992) provides another typical example of such technology. Overall, such systems have failed fully to replicate the ease and flexibility of co-working in real space, although efforts continue to be made to align their features with the social protocols of everyday working life. A recent example is reported by Obata and Sasaki (1998), whose system allows virtual visitors to approach gradually through public space.

Heath, Luff and Sellen (1995) review the evidence about this relative lack of success. It is suggested there may have been an over-emphasis on the support of informal, social communication as contrasted with more focused, task-oriented work. The review draws attention to the need to manage the transition between individual and collaborative work and the use of shared objects and artefacts, aspects which are further explored in the current work. Video-based systems *per se* still largely fail to address these issues, but the challenge has been taken up by the designers of collaborative virtual environments, discussed in section 3.3.

Nonetheless, video links have been shown by some authors to add value in some design and allied contexts. This is particularly so where physical objects are involved, as in a link between design engineers and the factory floor reported by Pagani and Mackay (1993), and similarly, in describing problems in the assembly of aeroplane parts to suppliers (Boeing, 1992). Harrison and Minneman (1990) advocate the use of video as a medium for design teams in their report of a lab-based trial with architects using live and recorded video. Olson, Olson and Meader (1995) describe a series of laboratory studies of groups working to define design requirements for an automated post office. Groups who worked remotely but were supported by video performed as well as groups working face-to-face, but those using audio communication alone showed a small but significant performance decrement. These conclusions are closely echoed by Harvey and Koubek (1998) in a mainstream engineering context. A less encouraging report about the benefits of video for design is provided by Harper and Carter (1994). In this case, a video link between the offices of engineers and architects engaged in a design project served only to emphasise a lack of commonality between the two professional groups. Taken together, the evidence suggests that video is helpful for limited, specific purposes in design, but less so as a means of fostering long term co-working.

3.2 Technologies that provide shared information spaces

Much collaboration among knowledge workers concerns the development of information artefacts, and thus the exploitation of some sort of information space - at its simplest, a real or virtual location where information artefacts are developed or stored. An information space may be used in real time, for example a whiteboard, or asynchronously, as a resource for the storage and retrieval of shared materials, as in the case of a shared document folder. In design, most information artefacts require information spaces that allow the use of graphics as well as text, and frequently also support for reference to, or manipulation of, concrete objects.

The use of information spaces is often facilitated by the tools that support co-ordination between group members, for example group diaries - these are discussed in section 3.4.

Real-time shared spaces are most typically presented as an electronic variant of the whiteboard or flipchart[11], available on all participants' screens. All participants may point, draw or write on the space provided, typically using mouse or stylus, although in many systems keyboard input is supported and images from other systems may be imported. Similarly, contributions are visible to all. Commune, an early example of a shared drawing space, is described in Minneman and Bly (1991) and further analysed by Lu and Mantei (1991). The account is particularly interesting because an instance of

[11] The flipchart metaphor is the more accurate, as most systems support the saving of 'pages', but the whiteboard concept is the more widely used. The closest analogy is to physical whiteboard systems which permit photocopies to be made of their surfaces.

ownership violation occurs, when one of the participants erases another's drawing without first asking permission. (Ownership arises as an issue in the small group analysis for the current work.) CaveDraw (Lu and Mantei, 1991) provides some limited means of dealing with these issues through the provision of 'personal' and 'public' sets of drawing tools. Sketches made with personal tools cannot be edited by others, but can be seen. Owners can convert sketches from one status to another. Numerous commercial instances of shared spaces exist, now usually integrated with shared applications (see below) and often videoconferencing. Increasingly, these tools are bundled with Internet and office software. Most shared whiteboards run in windows on individual desktop PCs, but variants of the technology provide separate displays of the size of conventional whiteboards, e.g. LiveBoard (Elrod *et al.*, 1992) and Tivoli (Moran *et al.*, 1995). The series of versions of DOLPHIN (Streitz, Rexroth and Holmer, 1997), based on observational studies of newspaper editorial boards are worth remark here. Dolphin is a generic meeting support system whose latest version incorporates both public and private workspaces (although the latter are not sharable), results showing that work is most productive when a combination of the two are exploited. (The use of public and private space is an interesting aspect of the small group work discussed in Chapter 6.)

Application-neutral shared spaces have also been tailored for specific purposes. Instances systems include numerous real-time shared editing systems, e.g. ShrEdit (Olson *et al.*, 1992b); the "Electronic Cocktail Napkin" described by Gross *et al.* (1998) which facilitates shared free-hand sketching for architectural design using handheld computers, and the page layout design application described by Gutwin, Roseman and Greenberg (1996). This last is one of a continuing series of studies of awareness issues in integrated group and individual workspaces, the most recent of which is reported in Gutwin and Greenberg (1998).

Olson *et al.* (1992b) describe the use of a shared text-editor by groups designing requirements for an automatic post office. The groups produced designs of higher quality than others using a conventional whiteboard and paper documents. The authors suggest that the groups using the shared editor may have been more focused on core issues. A comprehensive evaluation of the efficacy of shared workspaces is also provided by Whittaker, Geelhoed and Robinson (1993), who investigated the relative merits of audio alone and audio coupled with a shared workspace for an easy text editing task, a difficult text editing task and a graphical design task. The results suggest that for difficult text tasks or graphical design, the addition of the workspace brings overall benefits. Further adaptations of shared workspaces supporting, for example, design reviews are also beginning to emerge. Bochenek and Ragusa (1998) discuss the trial of four such systems, concluding that a sense of presence is vital to the review process, thus echoing the consensus about the role of video in technologically supported collaboration. Finally, there are some relatively uncommon variants on the usual two-dimensional workspace. Shu and Flowers (1994) describe the manipulation and combination of 3D blocks, implementing shared and individual views for different task types and a 3D pointer for conveying viewpoint and indicating objects. Shared views were preferred for highly collaborative tasks, independent views for parallel activity. More recently, Brave, Ishii and Dahley (1998) report on a system which also supports the illusion of shared physical objects, this time placing a greater emphasis on touch and haptic feedback. Such systems will have clear potential for design of physical objects without the added complications of immersive virtual reality, once they emerge from the laboratory stage.

Video-augmented spaces combine a shared information space with a video image of other participants. Simple implementations provide a separate video monitor, others display video in a window of the main screen. It has generally been shown although that task performance is not enhanced, the availability of visual cues from simple video improves co-ordination and creates a greater sense of teamwork. Among other examples are Egido (1990); Gale (1991); Tang and Isaacs (1993) and Newlands, Anderson and Mullin (1996). Of particular interest for design is ROCOCO (Clarke *et al*, 1996), where a shared 'sketchpad', implemented using a stylus and tablet, was combined with a separate video screen (The process aspects of using ROCOCO are discussed in section 2.2.) A number of researchers have developed more integrated combinations of shared space and video, such that other participants' gestures and/or faces may be seen in the same visual space as the shared work area. This is generally a drawing space, and the applications have been targeted at design tasks, with the aim of supporting the interplay of drawing and gesture observed by Tang (1989, 1991) in his influential study of design groups, and other authors. Different approaches have been taken to this challenge. VideoDraw is an early example of such a shared drawing tool which uses video to allow collaborators to see each other's hand gestures as well as sketches on a virtually shared screen. Tang and Minneman (1990) provide an account of trials employing design-type tasks. These demonstrated that the tool worked as intended within certain constraints, but there is no report of its utility or of the behaviour of designers using the tool. Scrivener's LookingGlass system (Scrivener, Clark and Keen, 1994) provides each of a pair of designers with a video image of the other person's face under the shared drawing surface, while ClearBoard and TeamWorkStation, developed by Ishii and colleagues (Ishii and Ohkubo, 1990; Ishii, Kobayashi and Grudin, 1992), integrate images of co-workers with views of the desktop or drawing surface. All these systems are reported as enhancing co-design, but are restricted to pairs of designers. Again, the underlying intent of this stream of work seems to have been pursued more recently in the context of collaborative virtual environments (next section).

Finally, *shared applications* software allows normal applications running on one machine to be available to other participants for collaborative viewing and editing. Again many commercial products exist (e.g. Fujitsu's DeskTop Conferencing and NetMeeting from Microsoft), and as before, these tools are now often integrated in a suite of utilities.

3.3 Collaborative virtual environments (CVEs)

Bowers, O'Brien and Pycock (1996) define CVEs as "where multiple individuals interact with each other in a computational environment rendered by Virtual Reality technology", thus distinguishing them from multi-party videoconferencing and interaction in text-based MUDS. Their relevance in this context is that CVEs support awareness of other participants' activities in the shared space, while (usually) maintaining a common frame of reference lacking in the simpler forms of videoconferencing. CVEs may support asynchronous as well as synchronous work, as described by Benford *et al.* (1997) in an account of an virtual environment VE which mimics the affordances of documents for everyday co-ordination in an office setting – for example, indicating whether work has started through the position of a virtual document on a virtual desktop.

However, most applications concentrate on facilitating real-time collaboration. Perhaps the most prominent of recent CVEs, MASSIVE-1 and MASSIVE-2 (Bowers, Pycock and O'Brien, 1996), model spatial awareness is through the concepts of *aura* (a defined region of space around an object or person), *focus* (an observer's region of interest) and *nimbus* (the observed's region of influence or projection). Intersections of these three 'bubbles' of space allow interaction between or among objects or surrogates for human participants in the CVE. Human surrogates generally take the form of simple 3D shapes, but MASSIVE-2 also supports some video-based communication. Reynard *et al* (1998) describe further integration of video into a 3-D virtual world in which the movement of participants determines the quality of video service provided – i.e. the video image sharpens as participants move towards areas of the virtual world, for example individual 'offices'. It is claimed that this affords a degree of spatial consistency and therefore improved support for mutual reference, and thus better integration of access to shared resources. Finally, the Spin system reported by Dumas *et al.* (1998) is a recent example of particular relevance in the context of the design meeting analysis to be discussed in Chapter 6. Spin is "designed for multi-user real-time applications, to be used in, for example, meetings…". Users are represented by 3D 'clones' and the clones and applications in use are displayed as if on a panoramic screen around a conference table. Telepointers allow interaction with one or more 3D 'documents'. The system aims to support eye contact, gaze awareness, facial expressions and a small range of gestures for pointing and as a means of expressing reactions, etc., but has yet to be fully evaluated in use.

It generally appears that CVEs are still some way from serious application rather that experimental trials. There are as yet few accounts of the use of such systems for specifically design tasks, although Greenhalgh (1997) reports that MASSIVE-1 has been used for 3D problem solving. A more strongly design-focused trial is provided in Hindmarsh *et al* (1998) where participants in a CVE manipulated physical objects, specifically in the context of designing furniture layout in a virtual room. The authors note problems with identifying the subjects of verbal references, which were only resolved by intrusive additional conversation. The results thus lend retrospective support to the evidence about deictic reference from the meetings study reported in Chapter 6.

3.4 Coordination technologies

As we have observed, much cooperative work is undertaken asynchronously, and a range of technologies have been developed to co-ordinate and support individual efforts. Since most such technologies are now everyday commercial products, rather than of significant research interest, they are only briefly referenced here.

Numerous examples exist, now available on the World Wide Web, of bulletin boards and news groups. More relevant to the current domain of interest are the examples of text based computer conferencing, where individuals can participate in, or simply browse, asynchronous textual discussions composed of asynchronous individual contributions on a particular topic or 'thread'. While many applications have remained at this level of functionality, others – of which the pre-eminent commercial example is Lotus Notes™, built around a set of shared databases – have added far more sophisticated features, incorporating document management, workflow and supporting the development of special purpose groupware utilities. Structured shared workspaces and discussions are also provided by BSCW (Bentley *et al*, 1997), a web-based system seeing growing adoption, and Interlocus (Nomura *et al*, 1998), among other recent examples, while applications such as those described by Maher and Rutherford (1997) and Jeng and Eastman (1999) carry structure a stage further by integrating shared CAD applications and database management. Document management systems are in widespread business use, and organise the creation of a document as well as its subsequent storage and re-use, while workflow tools allow for the design and management of complex group tasks as they 'flow' from individual to individual. Research work on the application of workflow systems has often tended to suggest that the regularity imposed restricts human capacities for improvisation and dealing with the unexpected, for example, Abbott and Sarin (1994), but more recently, as Grinter (1997), has reported, the value of such systems in automating and supporting the more routine elements of technical work has been demonstrated. Rogers' telling study (Rogers, 1993), of how a group of civil engineering design engineers attempt to coordinate the use of CAD files without such technology illustrates clearly the situations which give rise to the need for tools of this type. As already noted, the collection of design rationale is an important but difficult element in the coordination of complex projects: Klein (1997) typifies more recent developments in its aim to integrate rationale with geometry through a web-based tool.

The most pervasive coordination technology, is of course, email, both in its simple form and integrated with many of the other technologies just reviewed. As will be seen, email proved to be the key technological element in the field trial of collaborative technologies discussed in chapters 4 and 5.

3.5 The integrated use of collaborative technologies

In reality, of course, distributed design groups exploit a range of technologies to support their work, making selective use of those features which support working practice. There are comparatively few accounts of such usage in real-world contexts, but Olson and Teasley (1996) and Robertson (1997) provide two instances. Olson and Teasley report on a distributed design group in the automotive industry, supported by a range of off-the-shelf CSCW tools (desktop video, an object camera, an electronic whiteboard, application sharing, a version control system, email, a shared diary, tailored Lotus Notes™ databases, telephone, paper memos and a shared file server) most of which saw limited usage. Robertson provides a rich account of how a small software company - in a rare reported example of physically distributed design - designed an innovative educational product. The flexible interweaving of face-to-face and technologically mediated communication (phone, fax and the exchange of files and messages via a bulletin board) is described, together with the range of supporting artefacts involved. The communication tools defined work practice and were embedded in it. It is stressed that work included both design tasks themselves and the explicit articulation activity in support of shared meaning and resources.

3.5.1 Virtual design studios

A growing body of reports discusses the use of integrated technologies to support work between groups of distributed designers in controlled settings. These experiments are generally characterised in studies of architects as 'virtual design studios', but the term is equally applicable to other domains. (The technology does not provide immersive virtual reality, but rather aims to support working as if in the same office.) The material below discusses some of the more interesting instances of these. It may be observed that all of these have taken place in academic contexts.

Gay and Lentini (1995) provide a representative example of this work. Three groups of students at geographically dispersed locations worked on an engineering design problem – a windmill – supported by videoconferencing, text chat line, shared drawing tool, shared multimedia database of engineering information and online engineering textbooks. The channels were used in different ways, for example, videoconferencing for materials used as referents in discussion; multiple channels were used for parallel communication activities and one channel substituted for another when technical difficulties occurred. The authors observe that researchers need to consider which aspects of design activity such as gaze, actions and relative position need to be communicated. Maitland, Steinfield and Jang (1997) report a similar trial of collaborative technologies, also between three sites. Unsurprisingly, it was found that videoconferencing is not useful between sites with large time differences while email proved much more effective. More stimulating conclusions are derived by Vera *et al* (1998) in their study of pairs of expert student architects collaborating on a well-defined task using a shared whiteboard and either a text 'chat' connection or videoconferencing. No difference in quality of task output was found, but participants in the 'chat' condition spent less time communicating about low level design issues and more time on planning and negotiating the process. The authors suggest that high bandwidth applications such as video may not be necessary, since the designers adapted their process to the conditions, but acknowledge that their conclusions may not generalise to other types of task or less expert designers. The negotiative aspects of design are addressed in several recent developments, of which the CAIRO system described in na-Mora and Hussein (1998) is reasonably typical. This incorporates an "agent based coordination and facilitation mechanism to support the design negotiation process" but the efficacy of such systems in practice does not yet appear to have been investigated. Agents also mediate and co-ordinate CIFE WORLD, (Khedro, 1999) an integrated set of eight design and engineering applications for use in a geographically distributed environment.

A somewhat different perspective is taken by Kolarevic *et al* (1998) in their study of a week-long collaboration between groups of building design students in three institutions, supported by CAD systems, a central database and videoconferencing integrated via a Web-browser interface. Since the three locations were in different time zones, most collaboration was serial rather than synchronous. The report considers the nature of the collaboration, rather than how the tools were used. It is observed that co-working in this way was possible and indeed successful and that spatial (and temporal) separation forced a focus on the project, rather than the individuals involved while the shared database provided a design history.

3.6 In conclusion – CSCW technologies in design and issues in their introduction

The review above provides a picture of a burgeoning catalogue of CSCW tools, increasingly carried through the medium of the World Wide Web, many of which appear attractive in the context of the support of cooperative design. It is possible not only to communicate with distant colleagues by real-time transmission of electronic text, by speech and (albeit constrained) face-to-face means, but also to share 2D and 3D workspaces, documents and artefacts and our moment-by-moment exploitation of them. Distributed projects can be coordinated through shared planning tools and artefacts, often managed by agent technology, and the automatic routing of work from person to person and task to task. Every CSCW-centric conference – there normally at least two a year – brings a plethora of new tools, as do many issues of journals in the field. Still more developments appear in publications belonging to the many domains where there is potential for CSCW, not least that of design. However, many technologies do not re-appear in the CSCW application case studies situated in real-world contexts which form a large part of the remainder of the literature, and still fewer where such application has been mostly or wholly successful. Of course a gap between invention and deployment is to be expected for any technology, but here it seems particularly wide. Some reasons are no doubt simply practical – many of the more exotic tools either require the type of networks, hardware and software support not readily available outside research laboratories, while others are insufficiently robust to survive without the care of their inventors. It is well acknowledged that the more interesting reasons lie with the fact that

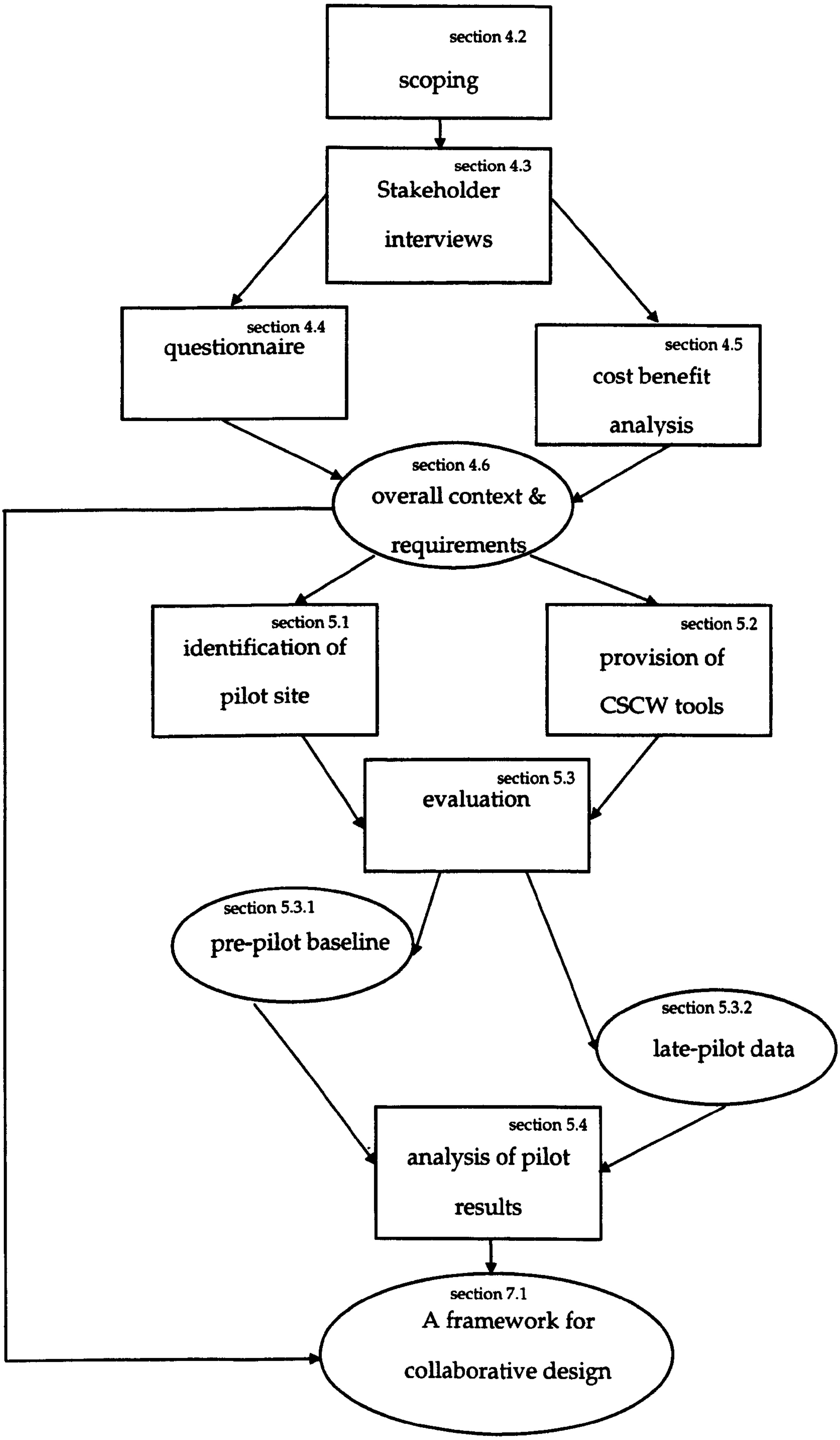
CSCW tools are not just, or even pre-eminently, aimed at supporting individuals carrying out a task, but at individuals working together. Thus the dynamic to be supported is person to person, or group to group, rather than person to artefact. With the introduction of interpersonal or intergroup dynamics, the social and organisational factors which influence the uptake of individual technology become much more complex, and provide the focus for much of the discussion in later chapters.

4 Understanding cooperative design: the organisational level

This chapter reports field work aimed at understanding the practice of collaborative design at Metre, a large, distributed engineering organisation. It then describes the derivation from this information of an initial requirements specification for cooperative design support tools, to be further explored through a pilot implementation using existing technology. The pilot itself is described and discussed in chapter 5.

The 'route map' overleaf illustrates the relationship between the activities which were undertaken in this part of the work, and also shows where each is discussed in the text.

Figure 4.1. The text linked to the flow of activities [box] and outputs [oval].



4.1 Identifying context and eliciting early requirements

At the inception of the Metre fieldwork, liaison had been established with a board-level technical manager in the organisation. The initial tasks were:

- to understand how Metre operated as an organisation involved in the practice of collaborative design and initial requirements for CSCW systems; and
- to identify potential contexts for a pilot trial of CSCW technology.

The underlying philosophy is reflected in the following statement from Tang (1991):

> *The design of collaborative technology needs to be guided by an understanding of how collaborative work is accomplished. By understanding what resources the collaborators use and what hindrances they encounter in their work, tools can be designed to augment resources while removing obstacles in collaborative activity.*

"How collaborative work is accomplished" necessarily involves understanding organisational context as well as end-user characteristics and requirements. As has been demonstrated in section 3.6, it is well established that CSCW applications must be tailored to organisational realities, whether the whole organisation, for example, Orlikowski's case study of the introduction of Lotus Notes™ (Orlikowski, 1992); or individual workgroups, their practices and cultures, exemplified by numerous reports such Bødker and Pederson (1991); Bowers (1994). In establishing context for CSCW at Metre, organisational structure, infrastructure and culture were explored together with perceived requirements for CSCW technology. There was a strong focus on existing expectations and perceptions of such applications, so as to identify costs and benefits for those potentially involved as well as appropriate technology, training and support. In this context, the approach was to *elicit* requirements, rather than to capture, define, specify or generate them, to follow the distinction made by Jirotka and Goguen (1994). The practical experience of the author and other members of the team elsewhere (Powrie and Siemieniuch, 1990 and other unreported work), strongly suggested that requirements do not exist fully-formed in the minds of stakeholders[12], ready to be acquired and

---

[12] The various groups of people who will have a *stake* or interest in the system. As well as direct system users, they include indirect users (who may use the system through an intermediary), remote users (who use the output of the system), systems staff, developers, change agents and budget holders.

tabulated, but need to be elicited through an early consideration of potential socio-technical solutions which are grounded in current organisational realities.

4.1.1 The ORDIT approach

Elements of the ORDIT approach to requirements elicitation were adopted in the initial fieldwork at Metre. The ORDIT[13] project (Olphert and Harker, 1994; Eason, 1996) sits firmly in the socio-technical tradition, exemplified in early studies by the Tavistock Institute and more recently by Soft Systems Methodology (Checkland, 1981), ETHICS (Mumford, 1983) and OSTA (Eason and Harker, 1989). As Emery and Trist (1969) note, the main tenet of the socio-technical approach is to optimise both the social system and the technical system to support the primary transformation process – the main purpose of the system. ORDIT embodies an emphasis on organisational context (the prime contact at Metre was vociferous that an understanding of their context would be vital for the success of any CSCW venture) and a strong, early solution orientation (useful in helping stakeholders to envision future technologies). It is also particularly suitable for CSCW projects, considering multiple stakeholder groups and the relative costs and benefits, in the widest sense of the phrase, accruing to them from the new system. Other ORDIT techniques would facilitate consideration of the likely effects of the proposed technology for Metre. On a more pragmatic note, both I and DUCK[14]'s project manager had both contributed to the development of ORDIT and were therefore familiar with the techniques.

Other approaches would, of course, have been possible. Requirements analysis could have been carried out following a conventional information systems method. For example, later versions of SSADM (in particular V4+) incorporate some user and organisational issues, but the version current at the time the work was undertaken did not facilitate consideration of these matters. Of the hybrid methods, Multiview (Avison and Wood-Harper, 1990) is informed by a strong socio-technical viewpoint, but lacks explicit consideration of organisational (as contrasted to end-user) concerns. Despite this, Multiview may have been a suitable approach, but its claims were overcome by a

---

[13] Funded by the European ESPRIT progamme, reference ESPRIT 2301

[14] The project under whose aegis this part of the work was conducted.

subsidiary objective of using ORDIT in an industrial-strength trial. Since at this stage the exact context for the CSCW pilot had not been identified, detailed information and data modelling techniques were not yet appropriate, but would be postponed until work on the pilot itself commenced.

Finally, much reported work in the early stages of CSCW projects (as, for example, reviewed in Plowman, Rogers and Ramage, 1995) draws on ethnographic techniques such as observation and videotaping of the continuing stream of everyday work. These techniques may have added a valuable complementary source of data, but were largely precluded by resource constraints and security considerations at Metre, except for such observations as could be made in the course of conventional interviews and meetings.

The ORDIT method as adapted for the current work had the following components:

i. Scoping the problem. At the beginning of any project it is important to scope the project and to agree with the client what will be done and what will be produced. The aims of scoping are to discover what the organisation is trying to achieve, and why it is considering change. The analyst and client:

- establish the proposals for change;
- draw up an outline analysis of the organisation;
- identify stakeholders;
- draw up and agree future requirements elicitation plans.

ii. Stakeholder interviews. Representatives of all stakeholder groups are interviewed to establish their current working practice, roles and responsibilities, presenting problems, goals and wishes for the new system. Stakeholder interviews may be supplemented by questionnaires, focus groups or other appropriate data gathering methods. ORDIT encourages the use of future scenarios in requirements work as a means of stimulating discussion of possible socio-technical solution options. The outputs of this stage feed into requirements and system specification documents.

iii. Organisational Impact Cost-Benefit Analysis. Once an outline system specification is available, cost benefit analysis is used to identify organisational and job related gains and losses for each group of stakeholders if the system is introduced.

iv. Responsibility modelling. Responsibility modelling represents in a diagrammatic form *who* is responsible to *whom* for *what* organisational tasks. The diagrams also show the information transformed or exploited in the task. In general, they are useful for both clarifying an analyst's understanding of co-working practice, and for reasoning about the effects of new systems upon roles and responsibilities. The information gathered at the requirements stage of this study was at too high a level for responsibility modelling to be fully useful, but it was intended to use the technique once a pilot project had been selected.

Each of the first three techniques and its output, is now discussed.

4.2 Scoping the problem

The scoping work produced an outline organisational analysis, an outline stakeholder analysis, established high level organisational requirements, identified the areas of Metre where CSCW might be introduced, and agreed a plan for the remainder of the requirements work.

This section reports the results of the scoping exercise. Information sources were:

- the Technical Manager (a senior post), Aquatics Division, Metre and the Metre Facilities Manager (existing technology issues only)[15];
- Metre company literature and organisation charts.

In the material that follows in this section, the headings follow the ORDIT Scoping Checklist as provided in the ORDIT handbook (ORDIT, 1993). The viewpoint is very much that of the Technical Manager.

4.2.1 Metre's organisational function, history and structure

Metre provided products and services and acted as a prime contractor in marine engineering, software engineering and command and control systems. Clients were predominantly from the defence industry. It was a joint venture company formed by two large organisations in the engineering domain. The history and ancestry of the company was however rather more complicated than this simple statement would suggest, and involved a number of other organisations which had changed names, reappeared and disappeared over the years. This had consequences for organisational structure, culture and work practices.

There were three divisions: Tactical Systems, Engineering and Aquatics. The fieldwork was to focus mainly on Aquatics, since this was the main preserve of Metre's representative on the project, the Aquatics Technical Manager. Within Aquatics were four technical subdivisions and central support functions: Personnel, Quality, Finance, Commercial and Technical. The subdivisions were further split into a number of business areas.

Geographical Distribution

The company had sites widely scattered throughout the length and breadth of the UK. This distribution partly related to the location of the antecedent companies; other sites were located close to prime customers, usually parts of the MOD. There was at the time no intention to rationalise this distribution, although it was thought possible that one or two sites might close in future. Divisions, subdivisions, business areas, project teams and technical specialisms were distributed across sites.

4.2.2 Current distributed working and communication methods

Current working practice followed a process of successive design refinement, with the need to reconcile interfaces and system-subsystem interactions almost continually. Teams varied in size from around 3 to 30 or more people; projects were of anything from 6 months to 2 years or more duration, although there were a few smaller projects. The complexity of these collaborative projects meant that design rationale was sometimes elusive if needed retrospectively. It was often decided that a team or part of a team should be co-located and members were then required in many cases to move house or to commute long distances daily or weekly. The team was then reallocated at the end of the project. This practice was found highly disruptive by most of those involved. Staff also moved to further career aims or timeshared between projects at different sites. Nonetheless, it was common for teams to be spread across several sites. For example, a

---

[15] No other stakeholders could be made available at this point.

(then) current project for the updating of a minesweeper design involved 30 people in two different sites in the south-west of England, the main Scottish location and a Scottish dockyard. In such cases, co-ordination was achieved through frequent meetings, a practice in part dating from procedures found convenient by staff distributed around central London. This meant that travel was a heavy overhead: senior staff would be away from their base at least one day a week and frequently as many as four or five days, but still found it difficult to manage distributed projects. Such travel was *not* generally regarded as a benefit. The frequency of meetings meant that although geographically isolated staff did not feel cut off, they did often miss potentially important informal contacts.

Travel was also an issue for another group: highly skilled 'remote experts' who were based at one main location and acted as specialist consultants for projects at their home base and elsewhere. Example specialisms included ergonomics and artificial intelligence. They were rarely dedicated to any one project for any length of time. For this reason, and because it was thought difficult to maintain motivation and career development without a sufficient critical mass of specialists in one place, remote experts were unlikely to be moved from their home location. The practice was to travel as required, but this made the experts temporarily inaccessible to other sites and prevented ready access to files and reference material.

4.2.3 Metre's organisational culture

Metre presented a diversity of organisational cultures, which varied with the type of work undertaken, the expectations of clients, and the traditions of particular disciplines or sites. For example, sites closely linked with MOD establishments were popularly considered to have a highly bureaucratic culture, and often a significant number of their staff were ex-MOD personnel. Such variation occurred even within individual sites, making apposite Bødker and Pederson's (1991) distinction between organisational culture and workplace culture, where workplace culture applies to "...a relatively small workplace environment, where it makes sense to talk about the workers as an entity."

Climate for technological and organisational change

Most staff were thought to be receptive to technological change, although organisational change was recognised to be a more sensitive issue and to require careful handling. There was also some suggestion (from Facilities Management) that users were often unaware of the potential of currently available applications, and therefore the introduction of CSCW would need to be backed up by substantial training effort. One issue was raised in relation to this was that of 'ownership' of data, machines and other facilities, since most information and technology belonged to particular projects and by extension to their external clients. Again, it was stressed that the climate for change would vary with the part of the organisation concerned. It was thought likely that staff would *prefer* to be involved in planning for change, but they would not necessarily *expect* such involvement. Management views on staff involvement would vary with the part of the company concerned.

Autonomy and flexibility

Individual project managers had their own purchasing budget for equipment - see section 4.2.4. At the time, there was considerable autonomy in matters of working practice, but there was growing pressure towards company-wide 'best practice'. Historically, sites, business areas and previous companies all contributed to the mix of practices and procedures. The adoption of the quality assurance standard, BS5750, had introduced some uniformity, but only on a per site basis. It was thought that collaborative technologies might result in some further voluntary harmonisation.

4.2.4 The technology baseline

The details of existing technology are not central to this study. However, the summarised points below indicate the technological context in which potential users of collaborative technology worked, and the skills they might have been expected to have acquired.

Hardware

- the choice of platforms varied with divisions, there were substantial numbers of both Macintoshes and PCs;

- all divisions also used computer aided engineering workstations for CAD, calculations etc., software engineering projects used Sun 3 and Sparc workstations;
- machines were generally shared by 2-3 people, and were not normally one-per-desk;
- senior staff commonly did not have computers, and where they did it was thought that their machines were not always exploited to the full - Macintosh Powerbooks were often chosen by this group, and tended to be regarded as a status symbol.

Software

- office tasks used Word™, Excel™, Powerpoint™ and MacDraw™ and a variety of project management tools;
- documents of any sort were almost all prepared electronically, in theory at least in conformance with standard templates and styles;
- software development projects used the TeamWork™ CASE environment - all material was held in computer compatible form, usually as text and diagrams;
- mechanical engineering projects use Autocad™ and in some cases, Intergraph™, for design and drafting, together with specialist analytic packages;
- Macintosh application sharing was available under Timbuktu™, but little used - this was thought to be because of lack of awareness, licence limitations, the need for a coherent working group to make this worthwhile, and unfamiliarity with the technology.

Networks

- private, secure, data links connected all sites;
- most, though by no means all, machines were linked to local networks;
- no serious attempt had been made to introduce email, in part for reasons of security, although it was used by Facilities Management and software engineering projects;
- classified material was not discussed over the phone but in face-to-face meetings;
- access to machines themselves was not strictly controlled by passwords or other means - this relied on physical access control to the building and the fact that all staff

were positively vetted, while temporary staff or visitors had to be escorted at all times.

Responsibility for technical facilities

There were three different approaches to the provision of technical facilities. One tradition, derived from one of the antecedent companies, was of centrally managed facilities which were then passed out to users. By contrast, a second tradition existed of project-based purchase (financed directly from external project revenues) and management, and finally Central Facilities Management purchased special purpose machines (high-specification PCs, Apollos and Alphas) for use by technical groups, e.g. for design visualisation, finite element analysis or acoustical analysis. Overall strategy for technology was guided and reviewed by the Technology Board. Membership comprised the overall Technical Director, Technical Managers from all lines of business and the Aquatics Technical Manager (the principal source of scoping information). The Facilities Manager was not a member of the Board, but had the responsibility of ensuring that information about the existence and capabilities of new applications was disseminated. The co-existence of these procedures is significant, as it subsequently constrained the extent of technology which could be implemented for the CSCW pilot. In particular, the prevailing culture was one of reluctance to invest in tools whose cost could not be recouped directly from external revenue.

Thus the technology infrastructure was extremely heterogeneous, and level of provision in some areas surprisingly low for an organisation producing state-of-the-art devices. Interconnectivity was constrained by both cost and the highly sensitive nature of much of the company's work. Given this baseline, the scoping work now turned to a consideration of how Metre personnel perceived the potential for the introduction of CSCW.

4.2.5 Perceived scope for CSCW

Driving forces for change

The initiative for change at the time came very largely from the Technical Manager, Aquatics Division, who was the main project contact. He was strongly of the view that

Metre should introduce some sort of CSCW or groupware in order to maintain efficiency and therefore competitiveness, and that to a large extent this could be achieved with commercially available products. Other managers were also said to be interested. Once new facilities were introduced, it was thought that use would percolate beyond the original group. The company Technology Board had a presence on all sites and once convinced of the potential for CSCW through a successful pilot project, would act as lobbyists for the technology to those who controlled the necessary budgets.

The DUCK project therefore provided a timely vehicle for experimentation, which if successful could be exploited as discussed above and if not written off as blue sky research. The funding supported some of the time of the Technical Manager and other staff to work on the requirements gathering and trial process, and a contribution to allied expenses and capital equipment costs. However, since project funding was at a level of around 30% of costs incurred, Metre had to find the remainder from existing resources. While reasonable staff costs could be managed, there were more constraints on speculative capital expenditure in such a project-based organisation, a limitation which was to have a significant effect on the implementation of the pilot project. (Note: this paragraph, exceptionally for this 'Scoping' section, reflects my own viewpoint rather than that of Metre.)

Application areas

Four potential domains were identified, and within these a number of current or potential projects. This list guided the choice of sites and interviewees for the next stage of the work. The generic areas were management meetings; proposal preparation (involving early design, creative technical writing, and financial authorisation procedures); software design and mechanical engineering design. All had projects requiring the co-ordination across multiple sites. Five design examples illustrate the implementation contexts envisaged at this stage – and amply demonstrate how widely teams could be distributed.

- a software design project, extending an existing oceanographic data analysis system, employing a team of 6/7 people based at two different sites some 70 miles apart in the Southwest of England;

- a development to provide interoperability between communications systems via a satellite link, involving a team of about 8 distributed across two sites three miles apart in the Southwest and London, in collaboration with an Italian partner;
- the design of non-corrodible materials, based in the main Scottish location and a site in the Southwest – a project which at the time relied on communication via a poor quality fax machine;
- the updating of a minesweeper design, involving 30 people in two Southwest sites 67 miles apart, the main Scottish site and a Scottish dockyard;
- a new multinational frigate design project, involving a large team in the main Scottish site, the London office and multinational partners.

Overall presenting problems and expected benefits

In summary, presenting problems were the cost and inconvenience of travel between sites, the disruption caused by co-locating project teams, the difficulty of co-ordinating the work of a distributed team and the lack of recorded design rationale in collaborative work.

Benefits expected were the alleviation of the above problems and a resulting increase in efficiency. More specifically, the software to be developed within DUCK was expected to support the special requirements of collaborative design and the capture of design rationale, and to provide some integration of off-the-shelf groupware tools. (This phase, which followed the initial pilot, does not form a part of the current study.)

Overall priorities and constraints

Identifying appropriate off-the-shelf facilities, e.g. email and desktop videoconferencing, and demonstrating their potential were considered by the main informant, the Aquatics Technical Manager, to be the first steps in creating a climate sympathetic to the introduction of CSCW. More generally, he was of the view that careful support of organisational change and integration of existing modes of working would be the key to the successful introduction of collaborative technology at Metre and the shift in organisational culture towards CSCW. Achieving a critical mass of users would be crucial.

As for technology, the tools should not require a new hardware platform, and should preferably run on more than one of the common Metre platforms. Current levels of security must also be maintained.

4.2.6 Stakeholder identification

The groups of stakeholders identified during the scoping work are shown overleaf. The list was used to organise the subsequent interviews. As will be apparent, there were many fingers in this particular pie, a factor that was to influence the choice and extent of the technology eventually implemented, the selection of the pilot project using that technology and the duration and evaluation of the pilot itself.

Terminological note: *Direct users* will have hands-on use of the proposed technology, *indirect users* may use the technology through intermediaries (typically a secretary) or use the outputs of the system concerned, *remote users* have some stake in the process but are not directly involved.

Direct users

- project team members
- project managers
- business managers
- remote experts
- computer facilities management
- clients and subcontractors

Indirect users

- business and other senior managers
- clients and subcontractors
- finance staff
- quality staff
- contracts staff
- administration staff

Remote users

- clients
- suppliers/subcontractors
- researchers in the DUCK project
- the CSCW programme

Systems staff

- central Computer Facilities staff

Developers

- DUCK team members

Change agents

- the Aquatics Technical Manager
- members of the Technology Boards
- The Division Executive
- Facilities Manager

Purchasers

- budget holders in business units
- Facilities Management
- DTI and EPSRC[16]

Figure 4.2. Stakeholders at Metre

4.3 Stakeholder interviews

The next major step was to interview a range of stakeholders so as to compile a comprehensive user requirements list and, more generally, to obtain a detailed, multi-viewpoint picture of the context for a CSCW project.

Some 22 semi-structured interviews were conducted by another member of the DUCK project team and I, both being present for each interview, using a structure designed by myself. At his request, the interviews were observed by the Aquatics Technical Manager. The Technical Manager, in liaison with local management, selected stakeholders from each of the four generic domains identified as candidates for CSCW technology: proposal preparation, management meetings, software engineering and mechanical engineering. Although the request was made to interview a full range of staff across the application domains, senior staff were over-represented in the interviews. Since this may have affected the information gained, the interview material were subsequently augmented by

[16]Sponsors of the DUCK project

a questionnaire distributed to a more balanced selection of staff. This is reported in section 4.4.

The main Scottish location and three sites in the south west of England were visited, providing a spread of both domains and sites. Interviews lasted from 30 minutes to approaching 2 hours, depending on the length of time which could be spared by the interviewee, the relevance of CSCW for the interviewee and the stamina of all concerned. The issues to be covered were:

- tasks and responsibilities;
- reporting structures and procedures;
- communication and travel patterns (with whom, where, how often, how...);
- experience of group working;
- problems in current working practice (emphasis on communications or group working related problems);
- software and hardware used, and access to this;
- extent of use of personal day book/design journal (of more relevance to the later stages of the project, beyond the scope of this study);
- scenario-based[17] exploration of the possibilities of collaborative technologies e.g. application sharing, electronic whiteboards....

Interviews were as far as possible conducted in or near the interviewee's normal workplace, and thus a reasonable ethnographically informed impression of the working environment was also acquired. Very detailed notes were taken of the interviews by the two interviewers and written up shortly afterwards on a per interview basis, structured around the topics listed above. Initially a small tape recorder was also used, but sound quality on playback proved unacceptably poor. Best practice at this stage would have been to have verified the content of the interview records with the interviewees, but resource constraints at Metre prevented this.

---

[17] Scenarios as a technique are discussed in section 6.6.1. They are particularly useful for illustrating the potential of future technologies.

At three of the sites, lunchtime seminars were held for anyone interested: a short presentation about the project was made and numerous questions answered. The points raised at the seminars were noted and used to supplement the interview data.

4.3.1 Detailed requirements elicited from the interviews

The interview data was consolidated and current problems and new opportunities for collaborative working identified. An initial table of requirements for CSCW systems, organised by application domain and stakeholder type was then constructed. This may be found in Appendix A. The requirements relate to the technical system (e.g. a shared electronic whiteboard is required) and the social system (e.g. substantial training and user support will be necessary). This was both inevitable given the socio-technical approach taken in requirements elicitation and desirable, since it focused attention on the organisational work which would be necessary for a successful implementation. (Note: the black art of specifying requirements for systems re-design from rich contextual information of this type is a fascinating but vexed research domain of its own, and thus beyond the scope of the current work, whose focus is the practice of cooperative design and its support.)

The following list of generic requirements was derived from the domain specific list. These requirements were prioritised by the principal stakeholder representative (the Aquatics Technical Manager) into three categories:

1. the pilot trial would fail if these requirements were not satisfied;
2. the usefulness of the pilot would be significantly enhanced if these requirements were satisfied;
3. 'wish-list' enhancements which would be desirable given unlimited resources.

Ideally, the prioritisation should have been done by a representative group of stakeholders. Constraints at Metre did not permit this.

The generic lists, starting overleaf, show the requirement in the first column and an initial mapping to technology (where appropriate) in the second. As far as possible the requirements have been expressed in technology-neutral terms, except where technology was explicitly stated by interviewees. It can be seen that almost all the priority 1 and 2 requirements could potentially be met by simple, readily available off-the-shelf collaborative technology combined with improvements in networking and server provision. (In this respect requirements at this stage are very comparable to those reported by Olson and Teasley, 1996.) The main outstanding requirement, for a shared online journal, was to be met by software development at a later stage in the DUCK project (Turner and Turner, 1997) and outwith the scope of this study.

This first set of requirements relates to communication related issues. (The grouping of requirements is somewhat arbitrary, but served to impose some structure on a very long list.)

Requirement	Technology mapping
Priority 1	
Universal email, universally used.	Company-wide email.
Better network speed.	Improved network bandwidth.
Priority 2	
Easier communication inside Metre - including the ability to leave messages when recipient not available, confirmation that messages have been read.	Company-wide email, one per desk to foster uptake.
Enable staff who timeshare between sites to stay in touch.	Company-wide email, remote access to files
Distance working for remote experts	Email, telephone conferencing, desktop video-conferencing, shared applications.
Improved communication when off-site.	Portables/lap tops & modems
Remote access to own resources.	Universal network and roving login
Access to Internet.	Access to Internet.
Running demonstrations remotely.	Shared applications.
Remote software installation & diagnostics.	Shared applications?
Priority 3	
'Please call me' messaging system.	Electronic alert
Easier communication outside Metre - including the ability to leave messages when recipient not available, confirmation that messages have been read.	External email and link to Internet
Support for occasional home working	Portables/lap tops & modems
Desktop access to Management Information System	Link to Management Information System
Co-ordination of work across stand-alone PCs.	Network & configuration control.

The perceived need to reduce the overhead of meetings and co-working with colleagues at distant sites generated a further set of requirements on tools to support synchronous working.

Requirement	Technology mapping
Priority 1	
Effective negotiation with external partners, preserving social cues, etc.	Videoconference suite or desktop videoconferencing
Improved interaction and co-ordination among team members.	Email, telephone conferencing, desktop videoconferencing. Editor with annotation. Shared whiteboard/graphics for story-boarding & diagramming.
Maintain informal contact with members of distributed teams.	email? Video links between offices?
Reduce time spent in face-to-face meetings and travel.	Desktop videoconferencing, shared applications, shared whiteboard.
Priority 2	
Use of graphics with non-native English speaking partners/clients.	Shared electronic whiteboard.
Reduce time wasting in meetings.	Desktop videoconferencing, whiteboard and shared applications[18].
Reduced travel for design reviews at other sites and 'remote experts', but avoiding the need to be 'chained to a keyboard'.	Email, desktop videoconferencing - ideally hands-off. Shared applications including drawing tools & whiteboard. Annotation tools, possibly voice.

[18] Olson *et al.* (1992) suggest that electronic meetings may spend less time on side issues.

Other requirements with a clear technology mapping – this time largely to asynchronous tools – stemmed from a wish to improve the sharing of information across the organisation.

Requirement	Technology mapping
Priority 1	
Easier access to information stored in project journals (A4 notebooks, kept as personal logbooks, but archived with other project documents.)	Online journal – must be as portable as its paper equivalent, and support sketches and graphical information.
Shared project journal for important decisions especially early in the design cycle, including diagrams as well as text.	Online journal, extension of CASE tools to early design decisions.
Preserve security of sensitive material, especially with regard to external access	Secure network, and online implementation of current practice.
Document transfer at anytime.	Central server with public access – permanently available, email.
Better technology dissemination & training.	Email, bulletin board, shared applications.
Multi-site configuration control	Extension to CASE tools for multi-site configuration control.
Priority 2	
Access to shared project material across sites.	Full networking, shared filespace, shared applications.
Easy retrieval of material for tenders, software reuse.	Shared libraries.
Document reviewing across sites.	Shared applications. Named, time & date stamped annotation to documents
Improved document management & access to post, invoices, etc.	Scanner, OCR., document management tools.
Reduction in time to produce reports	Shared applications, online reviewing.
Better 'skills' information	Skills database.
Improved access to remote reference documents.	Online reference library.
Improved access to, and organisation of project records.	Central server with public access – permanently available, document management tools
Shared diary.	Electronic diary.
Improved traceability, especially between multiple sites.	Electronic forms and authorisation, design history recording
Quick and easy way of recording decisions	Decision recording tool.
Better dissemination of information.	Email, bulletin board.
Access to online information, planning and projection tools.	Full networking; better dissemination.

Still more requirements related not so much to cooperative working in itself, but to the administrative support for such work.

Requirement	Technology mapping
Priority 1	
Better version control	Shared documents with version control.
Priority 2	
Consistency of styles, templates etc.	Application sharing?
Approval for large documents quickly, at short notice.	Email, electronic signatures, application/document sharing.
Better support for proposal planning process.	Shared planning tool.
Reduce time spent on project administration, internal authorisations, e.g. for travel..	Electronic forms and authorisation
Improved communication with management.	Email with read-receipt.
Streamline QA procedures.	Electronic forms & authorisation.

And finally, the remaining requirements which applied across all types of technology.

Requirement	Technology mapping
Priority 1	
More reliable network	More reliable network
New tools should be simple to use.	New tools should be simple to use.
Operation across heterogeneous platforms and operating systems	Tools must run across heterogeneous platforms & operating systems
Priority 2	
Cost savings.	Distributed meetings support.
Priority 3	
Minimise interruptions.	Email, voice mail.

Organising the requirements in this way does however lose the overall picture gained from the interviews. Senior staff were in the main enthusiastic about the potential of collaborative technologies. Although their potential was seized upon as a means of reducing the burden of travel, managers also saw scope for improving coordination in distributed groups and streamlining tiresome administrative and reporting tasks. It was very evident that tasks which were strongly designerly – laying out a control panel, checking the ultrasonic signature of an underwater weapon – were supported by a gamut of quasi-design jobs – checking the project schedule, submitting a report of this month's resource consumption, documenting QA procedures. But enthusiasm was not unalloyed.

Most were sceptical that videoconferencing, in whatever form, would be an effective substitute for face-to-face negotiation, and neither could they envisage how technology could support the informal awareness of everyday work-in-progress effortlessly provided by co-location. A variant of this concern was expressed by one of the 'remote experts' who relied on his occasional physical presence to maintain his visibility to his internal customers. More junior team members sought improved support for their individual tasks, citing, for example, the need for ready access to reference materials kept at other sites. (It was said more than once that while most design work was a team effort, tasks were organised as far as possible to minimise the need for interaction.) For almost everyone, concerns about security dampened enthusiasm for enhanced communications and increased information sharing – Metre's working material was in the main not merely commercial-in-confidence, but security classified because of close links to the MOD. Finally, while almost all interviewees considered themselves to be IT literate, they were doubtful about the proficiency of others, in particular (other) senior staff. One of the features of the site visits that continued to surprise us was, for a company developing state-of-the-art defence equipment, the relatively low level of IT support – several software engineers sharing one small Macintosh, for example. This was in part attributable to the reluctance of clients to finance expenditure on hardware, but also a matter of culture – more than once, the view was expressed that good engineering design is not done on a computer.

4.4 Expanding the requirements data with the questionnaire survey

The 22 stakeholders interviewed were a small proportion of the potential target population for CSCW. It was therefore decided to extend the base of the requirements gathering exercise by the cost-effective means of a questionnaire. The questionnaire was to cover the same ground as the semi-structured interviews, to gain an impression of how different CSCW technologies were viewed and to test three specific hypotheses arising from the interview data. It also allowed anonymity - important as a counterweight to any tendency in the interviews to provide 'acceptable' responses. Against these advantages, those completing the questionnaire (since it was not compulsory) would inevitably comprise a self-selected sample, with whatever bias this might introduce into the results. But since some people might be motivated to respond because the technology seemed a particularly bad idea, while others might be equally motivated by enthusiasm, any bias could be expected to be unsystematic.

The specific hypotheses to be investigated through the questionnaire were:

Hypothesis	Rationale
Senior staff will express more positive attitudes to the CSCW technology than junior staff.	This was the case in the interviews. Junior staff were more preoccupied with the problems of carrying out their own individual tasks, which in most cases were relatively self-contained.
Staff who travel more will express more positive attitudes to CSCW technology.	The technology could reduce the need for some travel.
Responses will differ according to the location of the respondent.	During the interviews references were frequently made to the differing characteristics of staff and culture at different sites, e.g. degree of computer literacy, sociability and so on. Further, staff at 'distant' sites might be expected to be more enthusiastic about enhanced communications than those at 'central' sites.

4.4.1 The administration of the questionnaire

Response was voluntary and anonymous and it was made clear in that the exercise formed part of the DUCK project rather than a management initiative. The questionnaire was distributed to the 200 staff located across 6 sites (including two client sites) of the Tactical Systems sub-division of Metre's Aquatics Division.

A copy of the questionnaire text[19] may be found at Appendix B. The questionnaire mixed forced choice questions and open-ended opportunity for comment. Items covered the following areas:

- personal details e.g. job title;
- current working practice e.g. computer use, amount of travel;
- problems relating to communications, distributed working and travel;
- perceptions of the technology e.g. perceptions of the utility of each service, expected frequency of use.

[19]The questionnaire wording, but not its content, was designed and piloted by a human factors specialist at Metre.

The tools covered by the questionnaire were:

- internal email
- external email
- remote sharing of applications
- shared electronic whiteboard
- desktop videoconferencing
- online design journal

Since some of the tools were likely to be novel, the functionality of each was described in the questionnaire text and a brief usage scenario provided. In the case of desktop conferencing a screen shot was also included.

Application sharing

This allows you to share software running on your machine, e.g. a word-processor with one or more other people anywhere on the network in real time. The others are able to see the same view of the document on their computer screens and if desired, each person can take it in turns using the mouse, keyboard, etc. to input to the application. Only one person needs to have the software running on their computer in order to share it with others.

Examples of usage: suggesting and agreeing amendments to a text documents e.g. part of a bid proposal, or design drawing, with one or two other people, asking for, or providing advice on a problem e.g. in running a particular piece of software.

(18a) Have you ever used application sharing? (Please tick).

Yes ☐	No ☐	Don't know ☐

(18b) How useful do you think application sharing would be to you if it was available? (Please circle).

Not at all useful				Very useful
1	2	3	4	5

Figure 4.3. Example showing part of the questionnaire text for application sharing.

The text was piloted with several subjects before distribution via internal mail. Instructions were included for its return by the same means.

4.4.2 The questionnaire results

One hundred and three questionnaires were completed and returned, a 51.5% response rate. The results for each of the questions and hypotheses are discussed below.

How were the different tools viewed?

Subjects were asked to score each of the tools for potential usefulness to them on a scale of 1 – 5, where 5 was the most positive score. Overall, the response was lukewarm, the mean for the most positively viewed service (internal email) being 3.41. The mean for the least positively viewed service (desktop video conferencing) was 2.65. Thus, as Figure 4.4 shows, the range in perceived usefulness was apparently small.

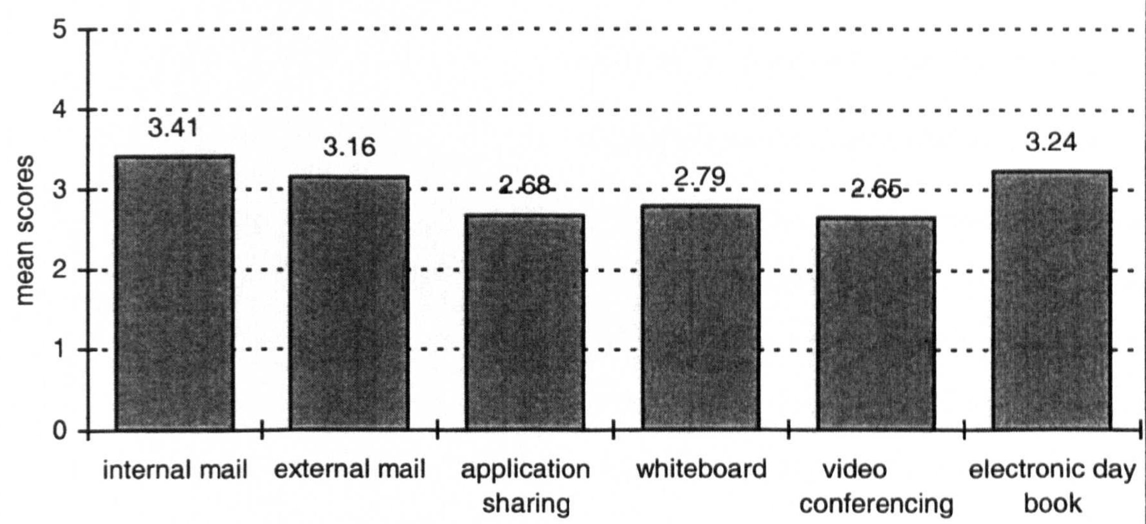


Figure 4.4. Mean scores for the tools, scale of 1 - 5.

Closer inspection of the data, however, suggested that there were differences between the perceptions of the various tools which were masked by the overall statistics. A diagnostic analysis of variance of the data, hereafter ANOVA, (9 incomplete data sets discarded) had more interesting results. Contrary to first impressions, the obtained value of F indicated that there were massively significant ($p = 0.000015$) differences between the services-as-perceived. A series of HSD tests[20] were conducted to identify where these differences lay. It was clear from the results that there was significantly more enthusiasm for:

[20] Runyon, R. P. and Haber, A. H. (1971). *Fundamentals of Behavioral Statistics.* Addison-Wesley, p223. Although the scoring scale used in the questionnaire cannot be shown to be of at a strictly interval level of measurement, the HSD tests as used here serve as a *diagnostic indicator* of sources of variation which can be investigated further.

internal mail as compared with

application sharing	(significant at the 0.01 confidence level)
desktop videoconferencing	(significant at the 0.01 confidence level)
the electronic whiteboard	(significant at the 0.05 confidence level)

and

the **online design journal** as compared with

application sharing	(significant at the 0.05 confidence level)
desktop videoconferencing	(significant at the 0.05 confidence level).

Consideration of these results suggested grouping the six tools into asynchronous and synchronous services, a distinction commonly made in taxonomies of CSCW systems. For this purpose, asynchronous services are those where users' communications are distributed over time, synchronous services those where communications take place more or less simultaneously[21]. It seemed probable that the source of the variance could be located by comparing scores for these two groups.

Asynchronous	**Synchronous**
Internal mail	Application sharing
External mail	Electronic whiteboard
Online design journal	Desktop videoconferencing.

The mean score awarded by each of the 94 individuals (discarding 9 incomplete data sets) to the services in each of the two groups was obtained. A Wilcoxon matched pairs signed ranks test was then performed. (Siegel, 1956, p. 76, considers the Wilcoxon to be appropriate where measurement is in the nature of an ordered metric scale, which lies in strength somewhere between an ordinal scale and an interval scale. The scale used here is of that nature.) The difference between the scores for perceived usefulness of synchronous and asynchronous services is significant above the 0.001 confidence level. The asynchronous services were scored much more positively, as shown graphically by the different peaks of the two distributions in Figure 4.5.

[21]This spilt is only one of convenience: a fast exchange of email could be considered synchronous, whereas passing control of the input device in shared applications is strictly speaking asynchronous.

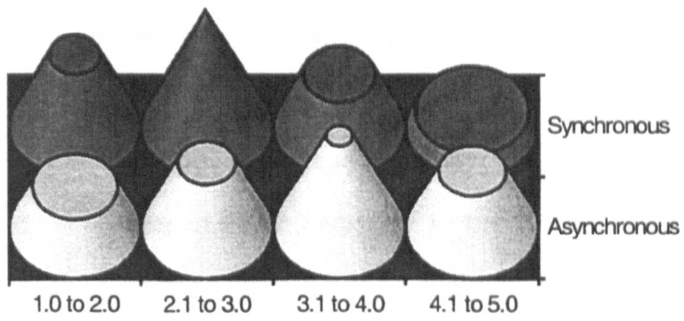


Figure 4.5. Mean scores for asynchronous and synchronous services compared

Respondents were also asked to consider which existing means of communication would decline in use if new services were introduced. Perceptions of the synchronous services are further illuminated by this data. From Figure 4.6 it may be seen that over 20% of respondents expected that the *synchronous* technologies of application sharing, electronic whiteboards and videoconferencing would reduce *asynchronous* communication by fax and memo. This suggests that for these people at least, the introduction of CSCW might not necessarily reduce travel for face-to-face meetings, but would facilitate a shift to directly interactive working where this was not possible at present.

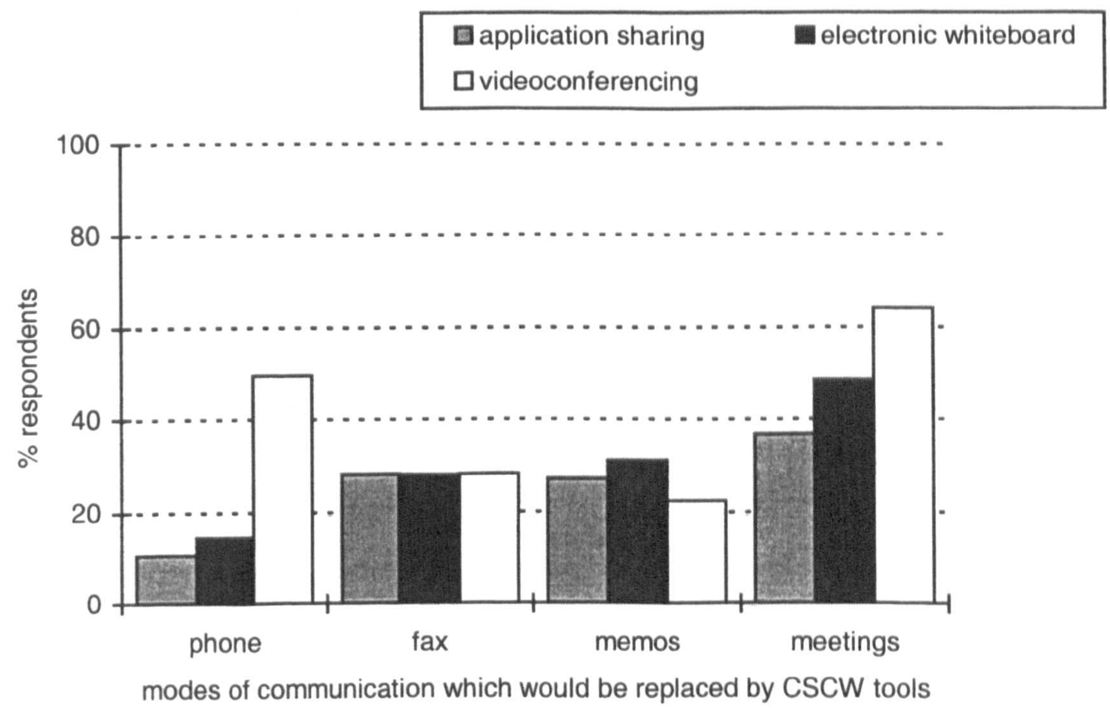


Figure 4.6. Modes of communication which would be replaced by CSCW tools

Do senior staff express more positive attitudes to the CSCW technology than junior staff?

Respondents were grouped according to their seniority using their reported job title, supplemented by background knowledge of job content at Metre. This resulted in groups of 33 senior staff and 61 junior staff (discarding 9 incomplete data sets). A Mann-Whitney U test (the most appropriate to this type of data and allowing correction for ties) was then performed on the mean score awarded to the tools. The results showed that senior staff were significantly more positive than their junior counterparts (significant at the 0.05 confidence level). The hypothesis to this effect is therefore supported.

Application of a Wilcoxon test to the separate groups of senior and junior staff showed that the more positive view of asynchronous over synchronous tools held for both levels of seniority (significant above the 0.001 confidence level).

Do staff who travel frequently express more positive attitudes to the CSCW technology?

Respondents were asked to categorise the frequency of their business travel into one of six groups. These were:

Never	Once a month
< once every 2-3 months	Every 2-3 weeks
Every 2-3 months	> once a week

If the responses are combined into two groups:

non-travellers and infrequent travellers (up to and including 'every 2/3 months';

frequent travellers ('once a month' or more);

this gives 37 non- or infrequent travellers and 57 frequent travellers. Carrying out a Mann-Whitney U test (with correction for ties) on the mean score awarded to the tools (discarding 9 incomplete data sets) supports the hypothesis that frequent travellers would be more positive about the CSCW technology (significant at the 0.05 confidence level). However, this result must be treated with a degree of caution, since if the respondents are grouped differently according to frequency of travel, allocating only those who travelled 'every 2/3 weeks' or more to the frequent travel group) the results narrowly fail to reach significance.

Do responses differ according to the location (and therefore organisational culture) of the respondent?

Respondents were grouped by home site (two Scottish sites and four in the Southwest of England). One Scottish site (only 2 respondents) and one Southwest site (1 respondent) were discarded. An ANOVA was carried out on the mean scores awarded by individuals. No significant variance between the remaining sites was found and therefore the hypothesis that response would differ according to location was rejected.

This completes the reporting of results concerning the *a priori* hypotheses. Consideration of these, in particular the modest level of overall enthusiasm, prompted examination of one further factor, access to computers[22]. Expectations may have been dampened because of current problems in access to machines.

Do responses differ according to ease of access to computers?

Respondents were asked how many other people used 'their' computer. As was already apparent from the requirements interviews, more than half (53.4%) shared their machine with at least one other person, while 27.2% of the total shared with two or more others.

---

[22] It was also useful to have information on the level of sharing for the planning of any large scale implementation of CSCW.

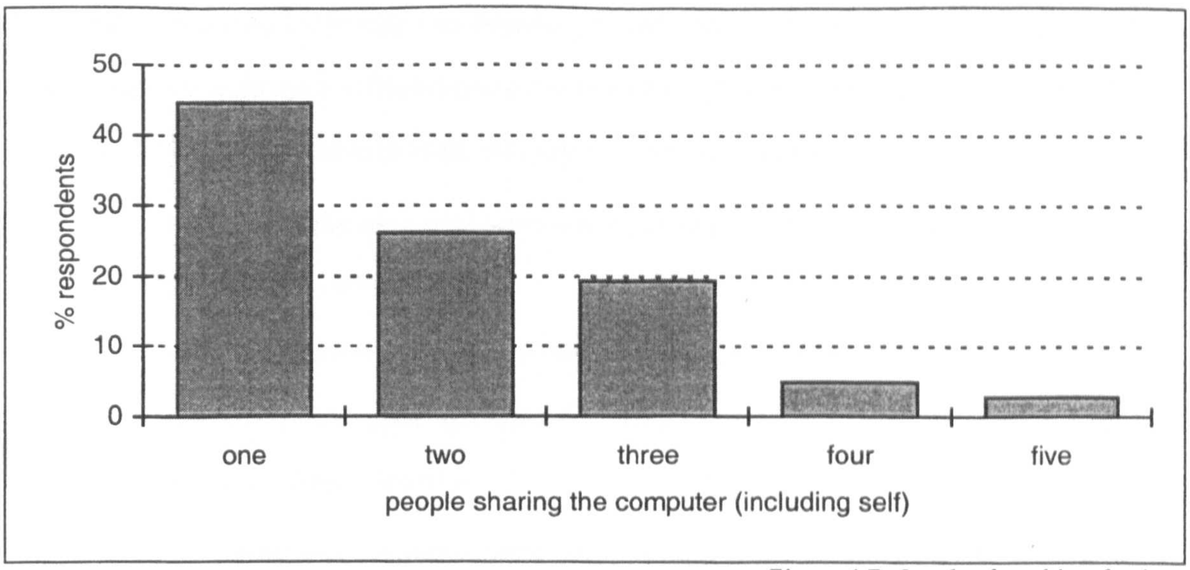


Figure 4.7. Levels of machine sharing

Respondents were grouped according to the number of co-users and an ANOVA carried out on the mean scores awarded to the services by individuals. No significant variance between the groups was found. A Mann-Whitney U test (with correction for ties) was then carried out to discover whether any difference existed between all those who shared machines and those who were sole users. Again, no significant difference was found.

Considering the questionnaire results

The questionnaire results in the main confirmed the impressions gained through requirements interviewing. Enthusiasm for CSCW was muted, but more evident in more senior staff and frequent travellers. There was also significantly more interest in asynchronous, as contrasted to synchronous, services. While managers and consultants expressed the most positive views, detailed examination of the data showed that they wished to retain face-to-face meetings for matters of negotiation, for making new relationships and for maintaining their own visibility. For more junior staff, perceived benefits again lay in improved access to information. The opinions expressed did not appear to be related to the home site of the respondents, contrary to expectations derived from reports of widely varying company culture. Nor did the ease of access to computers, as expressed by the degree of machine sharing, appear to have an effect on potential users' views.

Why was the overall level of enthusiasm somewhat tepid? Some of the free form comments from the questionnaire are collected below. Additional comments may be found at Appendix C.

People may have had difficulty in thinking constructively about novel technology when they found their current infrastructure inadequate. This position is worsened if it is felt that only management benefit from the latest technical wizardry.

Prior to any of the glossies mentioned here being procured, it would be nice to have my Mac SE upgraded. (Technical Consultant)

...at present it's difficult to get even a free Mac or a free telephone tie line. (Senior Consultant)

...expensive toys for the upper management to show off to clients... and eventually end up in a cupboard. (Senior Design Engineer)

Much of Metre's business involves information which is classified for reasons of national security. There is a good deal of concern as to how this could be transmitted over public networks and even about classified information being accidentally visible over video.

What about protection of classified information? (Systems Engineer)

Inappropriate to use in the defence environment because of the lack of control of views from cameras. (Principal Consultant)

People are reluctant to be tied to their desktops.

A variety of means to communicate - telephone, fax, meetings etc., allows me the opportunity to have a break from the Mac. (Secretary)

There is anxiety about effects on group dynamics.

Would give undue power to the participant familiar with the technology - not a natural way of interacting. (Senior consultant)

Those who express the most positive views of CSCW technology are managers and consultants, and those who travel frequently. Again, the comments help to flesh out this finding.

I have high hopes that CSCW will enable me to manage multi-site projects successfully in the future. We cannot easily overcome our geographic dispersion so this promises a means of reducing travel costs, personal disruption etc. (Project Manager).

People seem to be more and more reluctant to work away from home and I believe CSCW will make it easier and prove that it is both acceptable and entirely possible. (General Manager)

This questionnaire suggests an infatuation with 'problem-solving by technological overkill' that ignores the major problems (cost and time) involved in trying to make the things work. (Senior Design Engineer)

*Too k*******d. Too expensive. No facilities for working once meeting over. Seriously limits possibility of social life. Wife gives me s**t. (Consultant, on travel related problems)*

The finding that there is significantly more enthusiasm for the asynchronous services than synchronous ones is illustrated by these final comments:

Need [real] face-to-face to meet new people, once you know them, telephone is OK. (Principal Consultant)

Personal visits have the virtue of forcing people to prepare properly which they otherwise (in my experience) often wouldn't do for an electronic interface. (Senior Design Engineer)

The user community's views in perspective

How realistic are the views expressed by the potential user community for DUCK? Many of the issues raised are discussed in published case studies and meta-analyses of CSCW implementations. Overall, the prospective evidence from the DUCK user interviews and the questionnaire accords with much the retrospective evidence from the literature.

Little is explicitly reported about users' expectations of CSCW technology, although Bullen and Bennett (1990) observe that expectations derived from the original description of groupware continued to influence patterns of use some five years on. The concept of technological frames proposed by Orlikowski and Gash (1994) formalises such observations. Orilowski and Gash note how users' initial perceptions of the purpose and effects of technology (in this case Lotus Notes™) constrained takeup. It may be that the tepid enthusiasm was simply a more general reaction to novelty - in a recent survey of users, suppliers and 'experts' views on groupworking technologies, Lewis (1994) notes that the results of market research are often misleadingly negative for new products. As the response from the pool of potential users at Metre implied, in many groupware implementations only the simplest tools are used. Bullen and Bennett's survey (*ibid.*) of groupware use in 25 enterprises provides strong evidence of this. More recently, Bowers' (1994) account of the introduction of CSCW systems to a government organisation further illustrates the phenomenon, as do Olson and Teasley (1996), working with mechanical engineers, and the work reported by Blythin *et al.* (1997) in a banking context. (This area of the literature is considered further in section 5.4.5.)

Anxieties about undue advantage to those familiar with the technology are supported in some studies of electronic meeting rooms. Austin, Liker and McLeod (1990) studied how groups distributed control of the technology, the determinants of which members took control, and the consequences of acquiring control. Proficiency with the computer interface and the social influence were factors which predicted who would take control. Another concern about unequal benefits, that benefits would only accrue to part of the target group, in this case, management, is certainly reflected in analyses of the failure (or partial success) of groupware systems, as in Grudin (1988); Bowers (1994); Grudin (1994); Reynolds (1994). Finally, as discussed in section 3.1, there is also support in the literature for the user community's wariness of video links for getting to know people and conducting negotiations.

4.5 Organisational Impact Cost Benefit Analysis

The ORDIT Organisational Impact Cost Benefit Analysis tool provides a technique for assessing user and organisational acceptability of a proposed system by focusing on major work roles and organisational match. The proposed social and technical system is reviewed against a pre-supplied checklist of issues, with the aim of identifying benefits which must be realised, and costs which must be managed, if the new technology is to succeed. Costs and benefits are not viewed in strictly financial terms, but as potential changes for the worse or better. Costs should be viewed as pointers to areas where either the system specification needs modification, or where special care should be taken with user support, training etc., or both. The technique is used in the early stages of specification and highlights any organisational changes or modifications to the technical system which may be required. Each cost or benefit identified is scored on a three-point scale, where 1 represents some benefit/cost, 2 a clear benefit/cost and 3 a major benefit/cost. A change may be result in both benefits and costs, so may be scored in both categories. (A full account of the use of user cost benefit analysis for prospective evaluation in CSCW may be found in Eason and Olphert, 1996.)

Stages 1 and 2 of the procedure record information gathered during the requirements elicitation process about the proposed system, the organisational background and roles of key stakeholders. Stages 3 and 4 assess organisational impact for each user group and the organisation as a whole.

The process was carried out towards the end of the requirements phase with a view to clarifying the requirements and assessing the appropriateness of the technology as then envisaged (including both off-the-shelf tools and applications to be developed later by DUCK e.g. the online design journal). Ideally Stage 3, User Cost Benefit Analysis, should be carried out with a group of representative users. Resource constraints at Metre precluded this: the assessment is based on information gained from the requirements interviews and with the participation of the Aquatics Technical Manager of Metre as user representative. It was hoped that once the pilot project had been identified it would be possible to carry out Stages 3 and 4 again for those particular users, but once again this proved impossible because of time constraints at Metre. The more interesting of the cost benefit tables are included overleaf.

This first table shows the (then) current allocation of tasks to work roles: this contains summary information about existing practice not shown elsewhere.

Task	*Existing Work Role*	*Existing technology*
management reporting	project leaders and all levels of management	paper reports produced on word-processor & spreadsheets, MIS system, shared folders over network, meetings.
proposal authoring & co-ordination	full-time proposal team & domain experts as part-time contributors	word-processor, shared folders over network, exchange of disks, Mac broadcast facility, phone, meetings
proposal review & approval	proposal team & senior managers (some as members of co-ordinating committee)	comments on paper, meetings
mechanical engineering design	team members, team leaders, project leaders, tech. specialists	CAD software (PCs & workstations), calculation software, word-processor for reports (PCs & Macs), journals, paper drawings, reference manuals, standards etc.
software design	team members, team leaders, project leaders, tech. specialists	CASE tools & usual software engineering tools (PCs, Macs & workstations), (PCs & Macs), journals, reference manuals, etc.
design team co-ordination and management	team members, team leaders, project leaders	informal communication arising from being in the same office, meetings, Work Instruction Forms, phone to other sites, weekly paper reports, project repository of drawings, code etc. (on-line, with access control, version control etc.)
project management	senior managers	meetings, often involving travel to other sites
liaison with clients, partners	usually project leaders and above	email where available, phone, fax, meetings, paper and electronic documents.
provision of specialist advice	technical specialists acting as remote experts	usually travel to other sites, phone, fax
document production	secretarial staff if not done by authors themselves	word-processor (Macs & PCs)
technical support	Central Facilities staff	site visits or located on site, phone, memos
travel requests, expenses etc.	admin. staff	manually

Figure 4.8. Overall allocation of relevant tasks to work roles (current practice)

The second table holds the results of the overall organisational assessment. Other tables showing the results of the analysis may be found in Appendix D.

Issues	Change	Benefit	Cost
1. Planned benefits			
a. Resource reduction	Reduce time/money spent on travelling.	2	
b. Resource optimisation	Make better use of staff time by avoiding travel. Facilitate effective use of existing tools e.g. email.	2	3
c. Individual enhancement	Reduce travelling and relocation, improve communication, access to information, design history.	2	1
d. Organisational enhancement	Cost savings, improved quality of team work, better quality through design traceability.	2	2
2. System Operation			
a. Reliability	The integrity of data may be compromised unless proper access & version controls are implemented along with shared access. The system itself must be very reliable.		3
b. Security	Security may be compromised unless very careful access controls are introduced, & security considerations may limit exploitation.		2
c. Compatibility	New tools must be compatible with existing system.		1
d. Vulnerability to stoppages	If communication relies on the system, organisation is highly vulnerable to system downtime.		3
3. Organisational match			
a. Control mechanisms	Facilitates the introduction of company/division wide procedures.	2	2
b. Flexibility	Increased capability to work effectively wherever staff are located, and to change to meet client needs.	3	2
c. Adaptability	see b.	3	2
d. Culture and values	Matches the perception of the organisation as a hi-tech company, but may conflict with traditional culture derived from engineering and close links with MOD.	2	
4. User group assessments			
a. Team members	Minor gain	2	1
b. Team / project leaders	Minor gain	2	1
c. Managers	Significant gain	3	1
d. Specialists /remote experts	Significant gain	3	1
	Totals	28	25

1 = some benefit/cost; 2 = clear benefit/cost; 3 = major benefit/cost

Figure 4.9. Organisational Impact Assessment

Consideration of the organisational match assessment suggested that the major benefits of CSCW for Metre would be in terms of flexibility and adaptability, but that these benefits would only be realised provided that the system provided absolutely dependable reliability and security. It was therefore recommended that the project development team should pay particular attention to these two aspects. Even at the pilot stage, any suggestion that the technology was less than secure and reliable would be likely to prejudice extension to the company as a whole.

A successful implementation is equally dependent on the goodwill and enthusiasm of pilot users. The cost benefit analysis[23] for the various user groups (see tables in Appendix D) suggested that the main beneficiaries were likely to be managers and technical specialists acting as remote expert consultants to projects – exactly those groups which had produced the most positive response in the interviews and questionnaires.

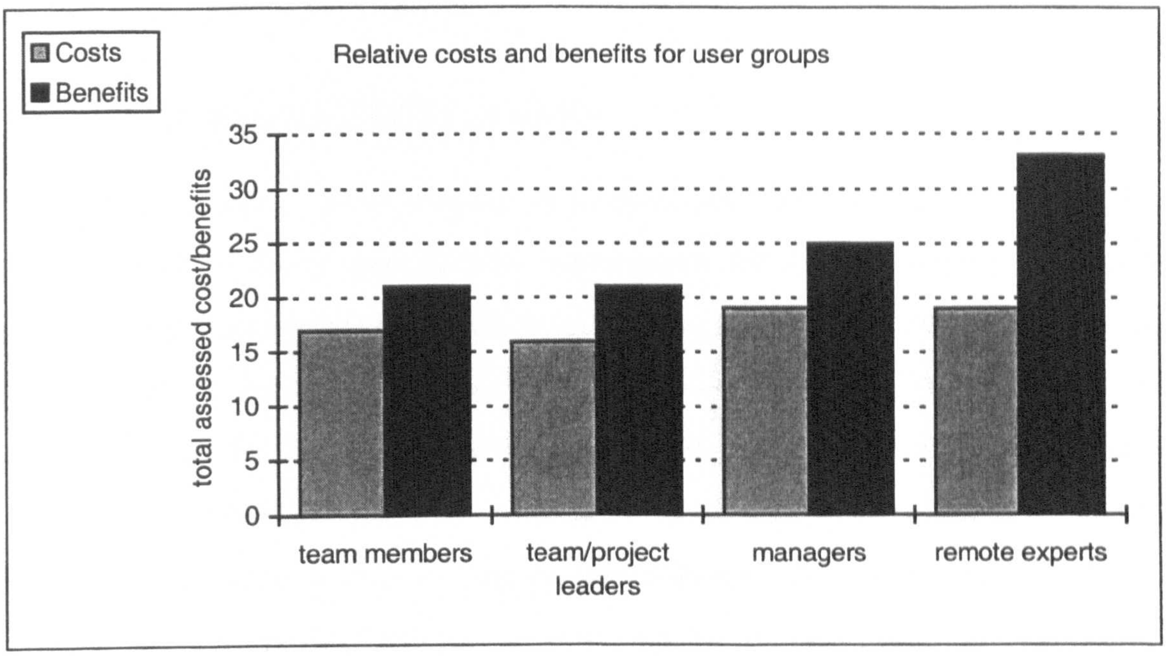


Figure 4.10. Costs and benefits for user groups

Since the success of CSCW technology requires a critical mass of users, something clearly had to be done to provide improvements for junior staff. The chief reward would be that if the technology were to be proven and accepted, project teams would no longer need to be located together. This would eliminate a good deal of the domestic disruption associated with moving project, but would be a very long-term pay-off. Moreover, such disruption seemed to be accepted as a normal fact of corporate life. The source of shorter-term benefit for this group was therefore unlikely to be the communications technology, but improved access to shared resources and eventually the proposed Online Design Journal. It was therefore recommended that access to shared references and reusable materials should play an important part in the initial pilot trial.

Finally, for all groups access to others to an individual's work-in-progress and knowledge resources was likely to be a sensitive issue. For example, Orlikowski (1992), in her study

---

[23]Note that the scores throughout the cost benefit tables can only be considered as ordinal data. Summing of costs and benefits for each group, and the histogram of these scores, must therefore be treated with a degree of caution.

of the introduction of groupware into a large management consultancy, notes that the norm that expertise was an individual's private capital adversely affected the take-up of the technology. While the organisational climate at Metre is far more conducive to cooperative work than the highly-competitive environment of management consultancy, design of the technology in this area clearly needed to take careful account of current norms about privacy and ownership.

4.6 The context for CSCW at Metre

The contextual work – initial scoping, stakeholder interviews and questionnaire – had now identified a set of quite modest requirements for tools to support cooperative working, and in particular the practice of cooperative design across distributed sites. For managers, these requirements were in the main concerned with the problem of co-ordinating distributed teams, for more junior staff, with access to distant resources necessary for their individual tasks. Most of the needs identified by staff and classified by the key stakeholder as high priority (see list at section 4.3.1), could apparently be met through improved network support coupled with off-the-shelf groupware tools. Such tools would comprise full email, supplemented by libraries of resources, workflow applications for routine adminstration, and where there was a need to work synchronously over distance, simple application sharing, an electronic whiteboard and possibly desktop videoconferencing. The requirement to track distributed design history would require a purpose built application, which would be the work of the second phase of the DUCK project (Turner and Turner, 1997) but is not considered here.

Finally, a review of the reported literature (as of Spring 1994) suggested a number of extra considerations to be taken into account for the initial pilot. These are listed below, sorted into those which appeared to have a realistic chance of immediate implementation, and those which appeared promising, but were either unacceptable in this context or required substantial development work.

Considerations for the initial pilot

Consider extension of existing single-user tools where these are well-accepted, rather than the creation of novel groupware. (Grudin, 1994)

Practical now, but possibly organisationally unacceptable

Use of video to foster a sense of team working when members are in different locations, although it may not make any appreciable difference to the end product in itself (Gale, 1991) This may account for the finding that the addition of video increases the usage of other desktop conferencing tools. (Tang and Isaacs, 1993)

Video records of meetings, to be used to enhance communication about design decisions to team members or clients not present at the time. (Harrison and Minneman, 1990)

Transmission of video images of the desktop, facilitating discussion of paper documents, and allowing communication by paper-and-pencil sketching. (Ishii and Ohkubo, 1990)

Use of video to convey gestures, as an aid to understanding design ideas (e.g. by showing how a control sketched on a whiteboard would work) (Tang 1991, Ishii, Kobayashi and Grudin, 1992)

(The issue for all these applications of video was one of security: one of the interviewees was concerned that sensitive items in the background of the video shot, for example engineering drawings mounted on the wall, would be easily overlooked by the those captured on video but readily visible to unauthorised parties at the other end of the link.)

Considerations for future development

Both informal annotation and more formal recording of the rationale for design decisions, linked to the parts of the design (e.g. a CAD drawing) to which they relate. (Reeves and Shipman, 1992)

Pen-based interfaces for the search and retrieval of information from records of small group meetings. (Wolf, Rhyne and Briggs, 1992)

4.6.1 Mapping current practice to CSCW technology

Current cooperative work practice at Metre could be classified according to the conventional CSCW matrix below, derived from Ellis, Gibbs and Rein (1991). This is of course a simplification of the complex sub-classes of synchronous and asynchronous interaction as discussed exhaustively by Antillanca and Fuller (1999), but serves the purpose of organising the Metre information well enough.

Cooperative Work Space / Time Matrix

	Same Time	**Different Times**
Same Place	face-to-face interaction	asynchronous interaction
Different Places	synchronous distributed interaction	asynchronous distributed interaction

The table below shows how practice at Metre and the target technologies identified map to this matrix. The general aims for collaborative technologies were to enable the migration of some cross-site activities from face-to-face interaction to distributed working, and to provide better support for those co-working practices which was already distributed.

	Same time, same place	Same place, different times
Examples of current Metre practice and technology where relevant	Meetings with clients and within teams Design reviews Informal co-working on parallel tasks by individuals	Design sign off (paper copy and manual signature)
Potential technology	Meeting support systems Design rationale tools	design rationale tools workflow electronic authorisation
	Same time, different places	Different places, different times
Examples of current Metre practice and technology where relevant	Liaison within teams (telephone/teleconference)	Iterative proposal writing & review (paper based, shared folders, physical transfer of files) expense claim processing (paper based)
Potential technology	Shared applications Electronic whiteboards Videoconferencing	document annotation and review tools email bulletin boards online discussions workflow

However, this apparently simple technological implementation had to be carried out within a number of constraints. Staff were sceptical of the utility of the more novel technologies suggested. Moreover, this was particularly so for junior staff, who had also been shown by the cost-benefit analysis to be less likely to gain from the introduction of collaborative technologies. Much engineering design work was not done on computer, and, partly for this reason and partly because of resource constraints, computers were commonly group resources rather than one-per-desk individual tools. And while improved networks and email would apparently solve many problems, such services could only be used if they supported current levels of security for material which was commonly of a highly classified nature.

Some of these points reflect the fact that requirements work of this type is inevitably grounded in current organisational reality and familiar technology. It should be stressed here that this was only the first part of the requirements process - the findings from the first pilot implementation were expected to be vital in revising the requirements set in the light of practical experience. Once some technology was in place, it was expected that users would both adapt the tools to meet their needs and adapt working practices to meet the potential of the tools - a process which has been termed 'adaptive structuration' in the literature (Galegher and Kraut, 1992).

Moving from the general to the particular

Up until this point the requirements work had been generic, and its results were potentially applicable to any CSCW initiative within Metre. The next step was to explore these requirements further by experimenting with technology to support group working in a particular pilot project. This would also have the demanding goal of demonstrating the efficacy of CSCW tools under the constraints identified above, thereby enhancing the receptiveness of the organisation for further moves towards technologically supported distributed working.

5 The CSCW pilot implementation and requirements revisited

The task was now to validate and expand the requirements information, while demonstrating the effectiveness of CSCW. This was to be done by identifying a pilot site which was representative of Metre's engineering design work, selecting, customising, installing and supporting technology so as to meet as much of the requirements list as possible, and evaluating how real life users exploited the new tools. This chapter:

- reports the implementation of the pilot (sections 5.1 and 5.2);
- discusses its subsequent progress and evaluation (sections 5.3);
- reviews initial requirements and expectations in the context of the pilot results (section 5.5.1 – 5.5.4);
- discusses the findings in relation to the reported literature and reconsiders existing frameworks for CSCW uptake at the organisational level (section 5.4.5).

5.1 The project and its requirements

The choice of project for the CSCW pilot was made by Metre, informed by the results of the requirements work. The decision was constrained by the small range of design projects operating in a distributed fashion over the period scheduled for the exercise and by security considerations. The project selected, referenced hereafter as SUB2, was engaged in the preparation of a bid proposal for the design of a new submarine – the bid itself included a substantial amount of detailed design work. The entire team, at its largest, comprised some 240 people who were distributed between three sites at Barrow (where the main project office was located) Glasgow, and Portsmouth. The duration of the project was scheduled to be 14 months, with the pilot running for eight months, starting three months into the project.

Two sub-teams were to be supported by the pilot technology

(i) Engineers working on the main propulsion machinery. The design problem involved fitting a substantial bulk of machinery within fixed hull dimensions. About twenty staff worked together in this team, some in Barrow, some in Glasgow. The two sites are just over 180 miles apart, the only practical transport between the two being a drive of about 3½ hours.

(ii) A team of human factors engineers (about 10 in total) who were concerned with the ergonomics of the whole submarine. These staff were almost all based in Glasgow but spent several days per week at the other sites. Significantly, the project manager for this part of the work was primary contact at Metre, the Aquatics Technical Manager. He was located at Barrow.

At this stage, best practice would have been to refine the requirements gained from the general exercise described above by working closely with those people who would actually be using the technology. Technology could then have been selected and tailored to provide a close match with working practice, and user confidence fostered. However, the extreme pressure of work on the SUB2 team – including its manager - and the secure conditions under which that work was carried out made this impossible. The immediate consequence of this was that, despite earlier intentions, insufficient information was available to use analysis tools such as dataflow diagrams and entity relationship diagrams in any meaningful way to model the technical system, nor could the ORDIT socio-technical tools be used to surface wider issues around the introduction of new technology.

5.2 The technology and its support

Despite these limitations, the requirements work discussed in Chapter 4 had provided a sound knowledge of the company's overall practice in proposal preparation and engineering design, the potential requirements and constraints on CSCW tools, the technical infrastructure and the perspectives of different stakeholder groups. Moreover, one of the stakeholder interviews had provided detailed data about a closely equivalent design proposal process. Work in the wider DUCK project drew on all this information to select off-the-shelf technology for the pilot, systematically reviewing a large range of products against the requirements identified. The choice was also heavily constrained by cost and the need to operate in a PC environment.

The selection of technology, the design and build of applications, the design and production of support materials and the implementation of the overall CSCW system were all undertaken by the wider DUCK project. Thus, although I participated in most of these activities, they are not directly part of this study.

5.2.1 The CSCW tools

The tools selected, both running in a Microsoft Windows™ environment, were:

Fujitsu DeskTop Conferencing™ (DTC) which supported synchronous working through remote application sharing, a shared electronic whiteboard or flipchart and also included integrated file transfer;

Lotus Notes™ (version 4), which supported asynchronous working by providing a structured, shared information space, email and file transfer. A number of purpose built applications were provided, described below. It should be noted that the functionality of the requirements and reviews database was specified by the SUB2 project manager.

> **A discussion database** supported asynchronous discussions of project issues.
>
> **A requirements tracking database** managed the definition and allocation of requirements for the design. A requirement definition would be entered by the project manager and posted to one or more people who would supply the necessary information. Their contribution would be linked to the original requirements document and progress monitored.
>
> **A review database** managed the document review process from submission of a document, through posting of the document to the reviewers, receiving comments and keeping track of progress.
>
> **A project archive** would store all final versions of reviewed material. Documents from the review database were automatically transferred to the archive.
>
> **A library** was to be used as a repository for common documents such as contract and standards information.
>
> **A project diary** was to be used to display deadlines and allow all group members to keep track of the project timetable.

5.2.2 Technical and user support infrastructure

The sites were linked by an ISDN line which enabled the collaborative use of DTC and Notes, which were installed on PCs at Glasgow and Barrow. Resource constraints at Metre limited the implementation of the pilot technology to two general access PCs, one at each of the two sites, a limitation which was to prove significant in the running of the pilot. However, it should be remembered that one-per-desk was not a normal feature of Metre working practice.

The Barrow PC acted as a general project resource as well as running the CSCW technology. For reasons relating to network configuration, it was impossible to access the network on which most of the project work was carried out and the CSCW network in parallel: the machine had to be rebooted in order to switch between the two. The machine was located in an open office, next to one of the potential users (the project manager again) but not on his own desk. The site was a large one, and the DUCK machine was 4 or 5 minutes walk away from some users, some of whom were located in a different building. In Glasgow, the pilot PC was installed in an office which was usually empty apart from the occasional meeting. The office had a conference phone, and was near to the project team's working area.

Simple, step-by-step manuals for both the Lotus Notes™ applications and DTC were provided, together with a 'procedures' guide which illustrated how the tools could support the distributed tasks undertaken by the group. This manual was drawn up in liaison with the manager of the team. A half-day's demonstration and training was provided for as many of the pilot users as could be gathered together. Throughout the lifetime of the pilot, a member of the DUCK team (not myself) was available three days per week at the Glasgow site and could be easily reached by telephone from Barrow. She provided support for the day to day use of the tools. Technical support was handled by another team member, again not myself, located outside Metre, but readily available by phone or email.

5.3 Evaluation in the pilot

Evaluation techniques for CSCW extant at the time ranged from tightly controlled experiments in (typically) university laboratories, for example Losada, Sanchez and

Noble (1990) and Olson *et al.* (1992b), to anecdotal accounts of systems in real life use, for example Abel (1990) and Reynolds (1994). Appendix E holds brief details of evaluation techniques employed in 34 studies across this range obtained from a review of reported CSCW case studies available at the time the evaluation was being planned, in 1994. The use of the methods chosen was as much an evaluation of the techniques themselves as of the results of the pilot, since there were, and still are, no commonly-agreed techniques for evaluation in CSCW, although Ramage (1999) provides an interesting approach to a methodology. There are a number of reasons for this.

At the heart of the problem is the fact that collaborative technology is necessarily multi-user technology. The effectiveness of the technology for individual users is important, but the usefulness for the group is even more so. As Grudin notes in his now classic paper, (Grudin, 1988) if individuals do not accrue some benefit, the system is unlikely to be used for long, at least if use is voluntary. But enthusiastic *individual* uptake is not the same as exploiting technology successfully to support *cooperative* working. Orlikowski's (1992) study of Lotus Notes™ in a large consultancy organisation provides an account of this phenomenon: where the product was used effectively it was as a personal information repository rather than a shared resource. There is also evidence of a similar effect in the introduction of Notes in a large manufacturer of consumer products (Turner and Turner, 1995). Thus effectiveness cannot be gauged by simply summing individuals' views, but nor can these be ignored. Blythin *et al.* (1997) provide a more recent commentary on this particular complication in their report of the results of one of DUCK's sister projects, introducing CSCW technology to a major bank, observing that perceptions of success depend on the varying objectives and motivations of stakeholder groups. Moreover, collaborative technologies are intimately linked to work practice. Introducing new technology inevitably changes such practice, and thus it is difficult to disentangle the effects. Finally, many evaluation techniques fail to address organisational and business goals - a system may be easy to learn, well-accepted and heavily used, but fail to have any impact on organisational effectiveness.

The main constraint for evaluation in the pilot project was that while the DUCK team themselves had ample manpower to devote to the data capture, the level of input which could be expected from the SUB2 staff was severely limited. Furthermore, the SUB2 work was both security classified and carried out in secure buildings, which meant that, as in the requirements phase, ethnographic approaches, *cf.* Suchman and Trigg (1991); Bentley *et al.* (1992); Hughes *et al.* (1994) and more recently, Benford *et al.* (1997) and Blythin *et al.* (1997)with their emphasis on observation of everyday work activities situated in their normal environment, were impractical. The table below shows the planned evaluation techniques and what actually happened in practice. Many of the techniques derive from Tang and Isaacs (1993) - this field study of CSCW technologies by a group of software engineers was the closest available match at the time to the pilot context.

Pre-pilot baseline evaluation		
Measure	Planned technique	*Implementation*
User profiling	Self-administered checklist or part of structured interview.	*Structured interviews carried out with all potential DUCK users*
User expectations of the technology.	Self-administered short questionnaire or part of structured interview.	*Structured interviews carried out with all potential DUCK users*
Logging of communications (inc. meetings).	Users to record using simple forms.	*Pressure of work on the SUB2 project precluded any significant logging.*
Projected time & effort schedule.	Prior to briefing about technology, each group leader to document a project schedule (including effort) based on past experience.	*Pressure of work on the SUB2 project prevented this. However, after the end of the trial period, estimates were made of the cost savings of using DUCK.*
Evaluation during the pilot		
Measure	Planned procedure	*Implementation*
Logging of use of tools (except email)	Partially automatic but also required users to record using simple forms.	*It proved to be impossible to log DTC use automatically. Notes use was logged automatically. Users were provided with manual log forms but these were not used.*
Logging of use of email.	Partially automatic but also required users to record using simple forms.	*Use of email logged automatically by Notes.*
Logging of other communications (inc. meetings).	Users to record using simple forms..	*Because of pressure of work, only some users completed forms - discontinued.*
User views of the technology.	Self-administered short questionnaire or part of structured interview.	*Structured interviews carried out with all potential DUCK users*
Collection of help requests, comments and suggestions.	Software help-desk & monitoring of Notes discussion database.	*Only formal help requests concerned difficulties with the Notes server. Discussion database not used, so no data. Informal requests, etc. collected by DUCK support person.*
Project profiling	Weekly (short) questionnaire to identify peaks & troughs of activities & fluctuations in tasks.	*Not possible during the pilot because of the pressure of work - some key milestones were identified in retrospect.*
Observation of the technology in use.	'Hanging about the workplace'; possibly supplemented by use of videotape.	*Videotape impossible for security reasons, some informal observation of the use of one of the DUCK terminals was carried out.*

Figure 5.1. Evaluation methods for the pilot project

Observation and *ad-hoc* questioning of users by the support person at Metre provided excellent background information. The structured interviews were relatively time-consuming for in terms of hunting down the target users, who were dispersed over the two major sites and sometimes working away at third sites, and in conducting the interviews themselves. However the interviews proved an acceptable way of gathering useful information - for example, the questions could be explained where necessary - and also provided an opportunity to address any queries and encourage the use of the tools. The results of the interviews are in section 5.3.1.

It must be stressed that users of the pilot technology were under heavy time pressure on the SUB2 project. Thus they were understandably reluctant to undertake a logging activity which provided no direct benefit for them, even when the initial simple forms were re-designed to make logging even more straightforward. Although a few users maintained a log of some of their communication activities, insufficient material was obtained to be a useful record of activity. Automatic usage logging is invisible to the users, and therefore has none of the problems associated with manual logging. However, as implemented by Notes, usage logging was not ideal for logging email activity, since, for example, the writing of an email message is only recorded if the message is saved as well as sent.

All-in-all, the experience suggests that pilot users engaged on real commercial work where there are time pressures (and few projects do not have such pressures) are very reluctant to undertake 'extra' evaluation activities themselves. The exceptions to this are where the activities are short, and undertaken in conjunction with a persuasive member of the experimental team, or users have some sort of 'stake' in the experimental project, or the activities are enforced by management - undesirable if the acceptability of the pilot project is to be promoted.

5.3.1 Pre-pilot baseline evaluation results

14 potential users of the DUCK technology were interviewed just prior to the start of the pilot. The interview structure was devised, and the resulting data analysed, by the author. The interviews themselves were conducted by the DUCK support person at Metre.

User characteristics

From the interview data, it was clear that the pilot users had varying background experience and domain expertise, but had already formed an established working group. The group had the following specific characteristics:

location: 2 described themselves as based at Barrow, 10 at Glasgow (the normal 'home' site for most users) and the remaining 2 spent their time about equally between the two sites;

discipline: 9 were engineers and 5 human factors staff.

allocation to the project: 11 were full-time on SUB2 and 3 part-time.

task experience: 10 had at least some experience of putting together similar proposals and the remaining 4 did not.

co-working experience: everyone had worked with at least some of the other people before.

project experience: everyone had been working on the project for some weeks at least before the pilot and some people had around nine months history.

technology experience: 6 had at least a reasonable degree of familiarity with PCs, a further 6 had some familiarity and the remaining 2 had none.

Patterns of collaborative working

The pilot group was also asked about their patterns of collaborative working, using an adaptation of an index of task interdependence developed by Van de Ven and Delbecq (1976). They were asked to estimate how much of their work on SUB2 fell into each of the following categories:

Independent working	Work is performed by you independently and does not involve anyone else in the team.
Sequential working	Work flows between you and one or more other members of the team, but only in one direction.
Reciprocal working	Work flows between you and one or more other members of the team in a reciprocal 'back and forth' manner.
Team working	You and one or more other members of the team problem solve and collaborate as a group at the same time to deal with the work.

Taking the means of percentages of work allocated to the four categories produced the following results[24]:

independent	sequential	reciprocal	team
19%	10%	40%	31%

On this evidence, then, the pilot users on this cross-site project spent much of their time working with others, and in particular in ways which might be thought to be suitable for support by the technology to be provided - email, asynchronous conferencing and information sharing and synchronous electronic conferencing.

Expectations of the CSCW technology

Having seen a comprehensive demonstration of the technology, users were asked to assess the tools for potential usefulness on a scale of 1 to 5, 5 being the most positive score. For individual tools, the mean scores were as shown in the histogram below. Although the variation in mean scores is not large, it can be seen that file transfer was viewed most positively and the discussion database least. (Note: at this pre-pilot stage, the file transfer tool is that provided through DTC.)

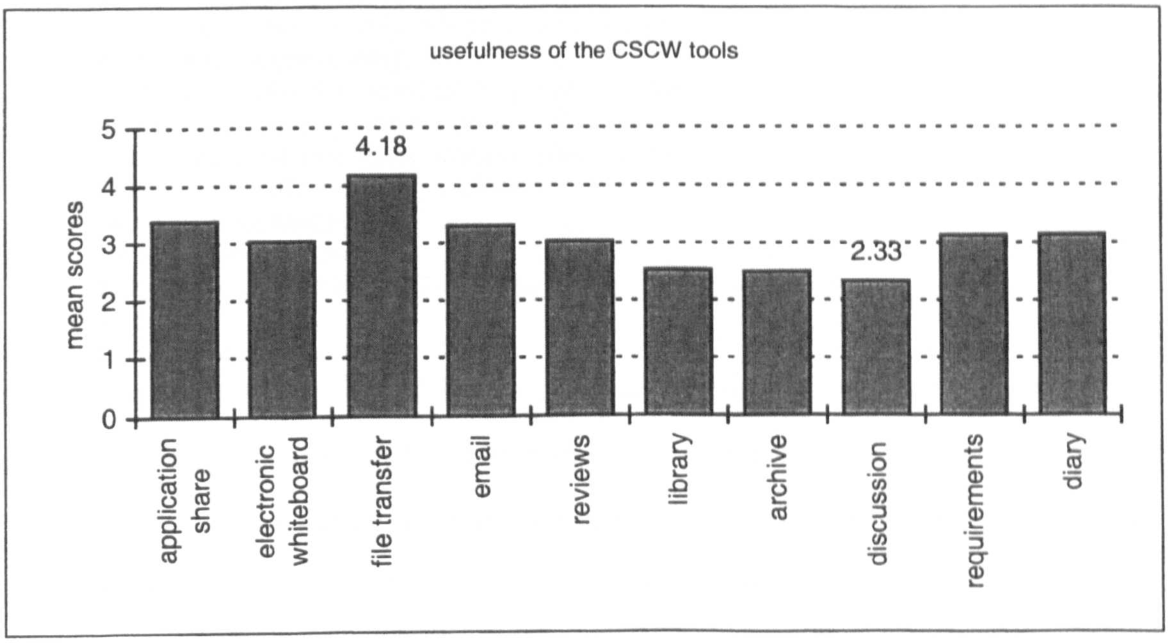


Figure 5.2. Perceived usefulness of the CSCW tools

The overall scores are shown below, together with the nearest comparable services canvassed in the questionnaire distributed to the wider Metre population and discussed in section 4.4. Enthusiasm in the pilot group was a little higher than that of the more

---

[24]This analysis ignores one user who invented a new work style 'director', which accounted for all their work.

general population[25]. One might speculate that the raised enthusiasm was the result of experiencing the demonstration, but this remains a speculation.

Mean for pilot technology as a whole	3.50
Questionnaire mean for all tools	2.99
Mean for DTC tools	3.43
Questionnaire mean for application sharing and shared whiteboard	2.74
Notes tools	3.21
No questionnaire equivalent	

The user group at the start of the pilot

In summary, the pilot users were working together across two sites, and were moderately convinced of the usefulness of the technology, in particular the file transfer function. However, comments invited at the end of the structured interviews, elicited a number of doubts. A summarised list of these follows, together with the number of times each issue was raised - note that some of the 14 interviewees made comments on more than one issue, and others made none.

1.	Machine not on own desk/inconveniently located	4
2.	Should have been introduced earlier in the project	2
3.	Little need for co-working	2
4.	No time available for experimenting with the tools	2
5.	The applications (e.g. library) need to be populated	2
6.	The applications (e.g. diary) already exist elsewhere	2
7.	Does not fit SUB2 working procedures	1
8.	More training needed	1
9.	Concerns about security	1
10.	Potentially unreliable/SUB2 too important to risk using unproven technology	1
11.	Cannot use from home	1

Comments 1 and 11 reflect the fact that, because of resource constraints, access to CSCW technology was limited to one PC at each site. The background to comments 2, 3, 5, 6 and 7 was that the group selected for the pilot were part of a very much larger project, with tools, working methods and procedures which had already been in place for some months and did not envisage electronic conferencing, etc. Comment 10 is a common reaction to any new technology, particularly when this has been introduced from outside the parent organisation. Time pressures on SUB2 accounted for the issues raised in

[25] Because response to the general questionnaire had been anonymous, it is impossible to know how many of the pilot group were represented in the questionnaire data.

comments 4 and 8, while 9 relates to the limitation that only unclassified material could be transmitted over the network used for the pilot project.

5.3.2 Late pilot evaluation results

Evaluation activities took place over a period of almost 5 months. Thirteen of the original 14 interviewees were re-interviewed after the pilot had been running for 4 - 5 months and a further 5 users (the Aquatics Division financial controller, a systems designer, an accounts assistant and two secretaries) who became involved after the pilot start-up were added. In contrast to the pre-pilot interviews, the interviewees were considerably more positive about the technology.

Use and perceptions of the technology

The increased enthusiasm masked the fact that the only tool to have been used seriously by this point in the pilot's life was Notes email. This had been exploited almost exclusively as a means of transferring files between Barrow and Glasgow.

Users were again asked to assess the tools for usefulness on a scale of 1 to 5, 5 being the most positive score. This time the technology received an overall mean score of 4.5 (3.5 in the pre-pilot evaluation). Email was again rated more positively (4.41) than before the pilot (3.31). This positive view was reflected in the users' comments, several of which observed that email had been essential to the team's work, and that its use for file transfer had avoided many car journeys between Barrow and Glasgow. Why were the other services not used? Drawing on the data from the first interviews, the 18 users were asked which of the original list of doubts applied. The results are overleaf, showing the number of instances with equivalent pre-pilot figures in brackets. Taking into account the increased number of interviewees, there are no changes of note here.

Machine not on own desk/inconveniently located	7 (4)
No time available for experimenting with the tools	5 (2)
More training needed	4 (1)
Should have been introduced earlier in the project	4 (2)
Poor usability of tools or support materials	4 (0)
Concerns about security	3 (1)
Potentially unreliable/SUB2 too important to risk using unproven technology	2 (1)
Little need for co-working	2 (2)
Does not fit SUB2 working procedures	1 (1)
The applications (e.g. library) need to be populated	0 (2)
The applications (e.g. diary) already exist elsewhere	0 (2)
Cannot use from home	0 (1)

The issue of fit with working procedures requires some amplification. The comment related firstly to the fact that the wider project had been running for some time, and had naturally established process and procedures – these were necessarily specific and detailed in order to coordinate input from so many sub-projects and individuals. Secondly, some of the artefacts used simply did not lend themselves to co-working over simple electronic media. The best example of this was provided by a human factors designer, who described how he and his colleagues would literally walk around a one-to-one plot of the submarine bridge to ensure that all controls were reachable and that someone would actually fit into the space available. Finally, though this did not surface as a specific comment in the evaluation interviews, the pilot support engineer observed that most users were involved in unrelated activities which necessitated physical presence at the other site, and usually took the opportunity to see to SUB2 business at the same time.

Four of the 19 users voiced criticisms of the ease of use of the tools or the supporting manuals. This view of the technology was not entirely surprising - the Lotus Notes™ user interface mixed end-user and developer features and was thus over-complicated for simple end-user tasks. As for the Fujitsu DeskTop Conferencing™ software, some features of this were found opaque even by members of the development team who used the software frequently. However, aside from the characteristics of the technology itself, there was another contributory factor: four out of these five people (and nine out of the total of 19 users) had little or no prior experience with PCs and therefore with the Windows environment. It should also be remarked that these issues are ones of *relative* difficulty. When the benefit was sufficiently convincing - avoiding a long drive just to transfer a file - a whole series of obstacles would be overcome. Here is the typical sequence of actions required to email a file, demonstrating just how persistent users were capable of being.

1. Write the file to a floppy disk
2. Walk several hundred yards to another building
3. Dispossess the habitual user of the CSCW machine (a senior manager)
4. Reboot the machine (CSCW tools did not work when connected to the normal network)
5. Make the ISDN connection
6. Run Windows
7. Run Lotus Notes
8. Change the Notes user ID if not the last user (a single ID per site was implemented in the later stages of the pilot)
9. Login to Notes
10. Compose a Notes message, attaching the file
11. Send the Notes message
12. Close down Notes
13. Reboot the machine and relinquish it to its *de facto* owner.

The suggestions elicited as to how the tools might have been improved complement the perceptions data. Suggested improvements are shown below - the numbers in parentheses indicate the frequency of each suggestion.

Improvements in user support		*Improvements in technology*		*Improvements in match to context*	
Simpler manuals	(4)	Terminal in own office/more terminals/network access	(15)	Match tools, training, manuals & procedures more closely to practice.	(4)
More training/explicit encouragement of culture change	(3)	Sort out operating system level problems before introduction	(4)	Earlier introduction in project lifetime	(1)
Introductory training/manual available throughout for new team members	(3)	Improve Notes user interface	(3)		
Advertise the tools throughout the pilot's life	(1)	Combine DUCK tools with conference calls	(1)		
		Include videotelephony	(1)		
		Encrypted line	(1)		

Figure 5.3. Suggested improvements to the pilot implementation

In conclusion, then, the interview results show that the technology had very narrow use. However, the one service that was used, email, was considered indispensable. The underlying reasons as stated by the interviewees very much reflect problems with access to the technology and its fit with the SUB2 project's normal work practice. The importance of these matters is further supported by the suggested improvements.

Monitoring usage

Representative data was not obtained from manual logging of user activity. Such manual logs as were available, the automatic Notes logs of user activity, observation of the terminal in use and the informal questioning of its users, confirmed in more formal user interviews, underlined that only the Notes mail service had been used in earnest, very largely for the transfer of files. However, one DTC session *was* recorded on a log form - the application sharing tool was used to demonstrate the use of a word processing template.

Analysing user activity from the automatic activity logs for Notes mail is not entirely straightforward. As noted previously, the writing of an email message is only recorded if the message is saved as well as sent. Furthermore, confident users tended to send messages on behalf of their more trepid colleagues. In view of these circumstances, the most useful index of email activity in the SUB2 group is the aggregate of the receipts[26] logged for each member's mailbox, and for the two common mailboxes set up for Barrow and Glasgow. Notes email activity within the SUB2 group was almost entirely confined to the members of the group themselves.

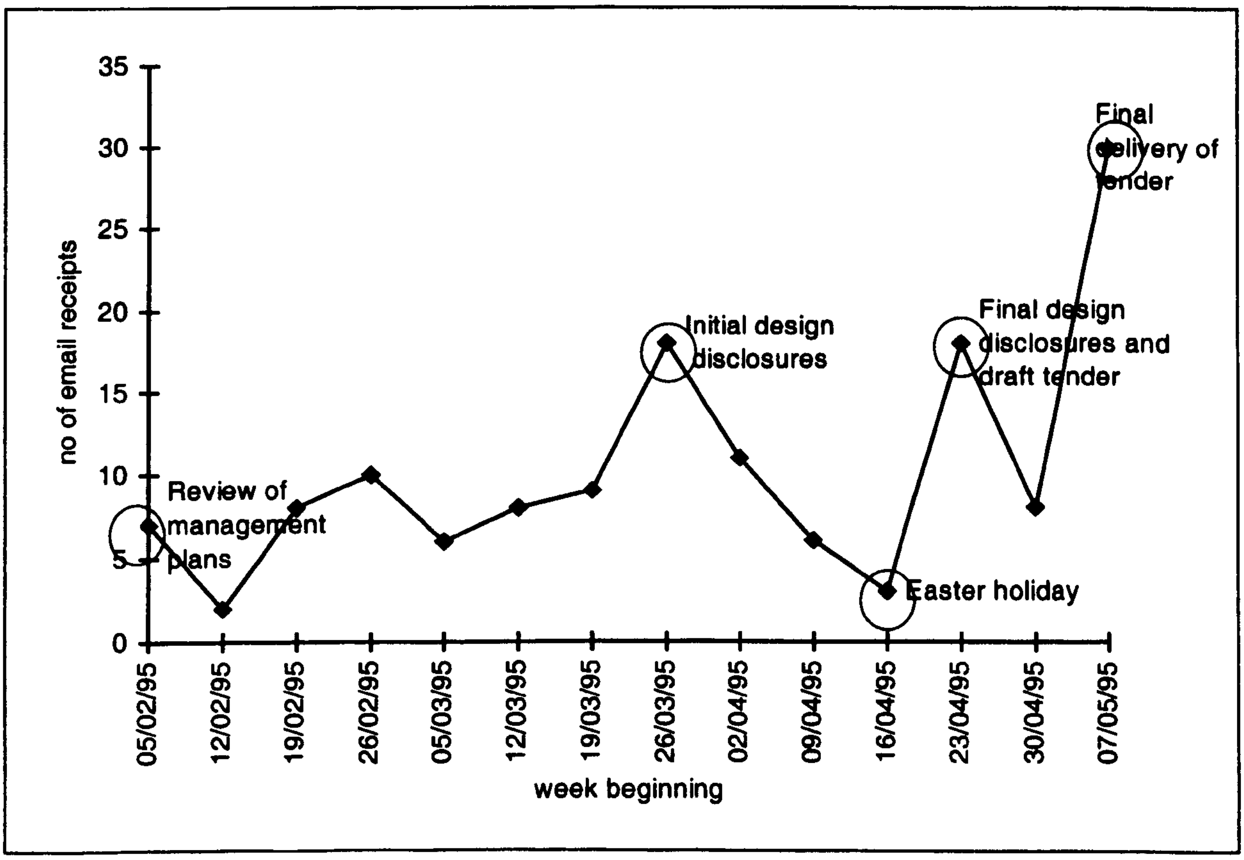


Figure 5.4 Email usage and project milestones

As can be seen from the graph covering the period from when the pilot started in earnest to the end of the evaluation period, the level of email usage was subject to quite marked fluctuations: unsurprisingly the peaks tend to correspond to project deadlines, the troughs to holidays or periods of less urgent work. The graph also shows the main milestones for SUB2 during the period: this information was obtained from members of the project team.

5.3.3 Cost savings achieved

An internal Metre exercise estimated cost savings achieved through the technology. Senior members of the SUB2 team were of the view that, without the support provided, more staff would have been moved to the Barrow site for longer periods – or even for the entire proposal preparation period – in order to complete the bid by the final milestone shown in the graph at Figure 5.4. Comparing the absolute costs of doing this with the running costs (ISDN rental and call charges) of the tools, the savings estimated amounted to a substantial £11,000 over a 6-month period. Further opportunity costs in terms of seconded staff being unavailable for other work would also have been incurred.

5.4 Discussion of the pilot results

As observed earlier, evaluating success in CSCW is not a straightforward issue, and is inextricably related to the perspectives of different stakeholder groups. The evaluation data obtained was both encouraging and somewhat disconcerting for the DUCK team. The user group was extremely pleased with the technology and many felt that their current tasks could not have been achieved without it. Indeed, a new verb had been coined - 'to duck' - meaning to send a file by email. However they had seriously exploited only one of the range of facilities provided - the Notes email service. Notwithstanding this the pilot enabled a worthwhile saving in estimated project costs, and attained one of its main goals of stimulating interest in the potential for CSCW at Metre: the Metre personnel involved in the DUCK project received a significant number of enquiries about the potential for CSCW elsewhere, and these continued long after the pilot had terminated. The remainder of section 5.4 goes on to discuss the results in the context of original requirements, expectations and related CSCW and IS literature.

[26] Receipts are logged by Notes as server writes to the mailbox.

5.4.1 The requirements viewed with hindsight

Since most of the tools were not exploited, firm conclusions cannot be drawn about revised requirements. It is nonetheless clear in retrospect that the choice of technology for the pilot was over sophisticated, particularly bearing in mind the lack of familiarity with PC platforms. The user group was not unusual for Metre. Indeed recent contacts with engineering firms in the north-east indicates that even in 1998, individual PCs or other workstations are not commonplace for technical staff[27]. This suggests that there is a requirement in this domain for a small set of simple applications. Indeed all that is essential may be a common interface which provides:

- send/read messages
- asynchronous transfer/sharing of files or other artefacts
- live sharing of an application with a minimal set of functionality
- a simple, pre-structured forum for asynchronous group discussions and sharing of generally useful material.

The mimimalism of such a toolset would afford flexibility, and in itself require little process redesign, being capable of supporting many different uses. A further issue is the explicit support of work groups which share machines. While designers require sufficient machines to enable convenient *ad hoc* use, some of which allow co-working under private or secure conditions, they do not necessarily want one machine per desk. Collaborative technologies under these circumstances are best supported by network environments such as NT v.4, which support mobile IDs and profiles, together with access from any machine to personal files.

Finally, the pilot at Metre was undertaken some four years before the final draft of this report in spring 1999. One might speculate what the results would have been had 1999 technologies been available. Internet and intranet technologies, had they been adopted by Metre, would certainly have made the implementation process easier. However, the security constraints which influenced so much of the pilot process mean that the widespread adoption of such technologies is unsuitable for this context. As for the tools

27 I am indebted to Phil Turner of the University of Northumbria for this information.

themselves, application sharing and sundry permutations of computer and video conferencing software are widely and cheaply available on standard software and hardware platforms, and Lotus Notes™ now exists in a simple, cutdown version designed specifically for end user work. All this would have further aided implementation, the more so if more PCs could have been acquired. However, while cheaper and in some cases easier to use, the underlying functionality is still very similar: it is doubtful whether the results would have been significantly different. There is also a larger issue here about conducting research drawing on field trials of technology in rapidly changing technological environment. Our knowledge of how best to fit technology to organisations cannot advance without practical trials, yet the technology used will often be overtaken by the time the results are collected, still less analysed and reported. Research under such conditions must abstract away from the surface features of specific tools and interfaces to a consideration of the essential functionality afforded and its relationship to user and organisational tasks.

5.4.2 The match with projected user benefits and costs

The initial requirements work had included a prospective analysis of relative costs and benefits (see section 4.5). Here are the organisational impact cost benefit tables again, this time with retrospective notes on whether the projected benefit or cost was actually realised in the pilot implementation. It will be seen that most of the projections made were in fact realised, despite the partial uptake.

Issues	Projected change	Actual change
1. Planned benefits		
a. Resource reduction	Reduce time/money spent on travelling.	Achieved
b. Resource optimisation	Make better use of staff time by avoiding travel. Facilitate effective use of existing tools e.g. email.	Achieved (Email was not available to these users before the pilot.)
c. Individual enhancement	Reduce travelling and relocation, improve communication, access to information, design history.	Partially achieved - sharing and transfer of current information was improved, but information repositories not exploited, and no mechanism for design history.
d. Organisational enhancement	Cost savings, improved quality of team work, better quality through design traceability.	Partially achieved - cost savings made, no impact on design traceability.
2. System Operation		
a. Reliability	The integrity of data may be compromised unless proper access & version controls are implemented along with shared access. The system itself must be very reliable.	Very largely achieved - system reliable once fully operational. Access & version control not an issue because of limited exploitation.
b. Security	Security may be compromised unless very careful access controls are introduced, & security considerations may limit exploitation.	Avoided: system only used for non-sensitive material
c. Compatibility	New tools must be compatible with existing system.	Not achieved
d. Vulnerability to stoppages	If communication relies on the system, organisation is highly vulnerable to system downtime.	Only an issue once, when disk space exceeded because of heavy use of the system just before a deadline.
3. Organisational match		
a. Control mechanisms	Facilitates the introduction of company/division wide procedures.	Not relevant because of limited scope of pilot.
b. Flexibility	Increased capability to work effectively wherever staff are located, and to change to meet client needs.	Partially achieved - more effective work without travelling.
c. Adaptability	see b.	see b.
d. Culture and values	Matches the perception of the organisation as a hi-tech company, but may conflict with traditional culture derived from engineering and close links with MOD.	Some initial reluctance may have been culture-related; relative unfamiliarity of PC based office applications may have influenced perceptions of ease of use.
4. User group assessments		The number of users too small to draw reliable conclusions about differential gains between groups. However, all pilot users considered the tools to have been useful.

Figure 5.5. Organisational Impact Assessment reviewed

5.4.3 Super-user centred methods?

User-centred methods are intended to ensure that new systems are grounded in users' needs, tasks, characteristics and context. Most such methods, including ORDIT, stress the importance of involving representatives of all stakeholder groups in the design process. However, as we have seen, constraints in client organisations do not always permit this. In such circumstances, working with the narrow range of stakeholders who are available can be counter-productive. The system design may still be user-centred, but it is centred on one particular user group. The design becomes unbalanced, a serious problem in any CSCW initiative where the cooperation of different stakeholder groups is entailed. In this case, the design of technology and proposed uses were strongly influenced by one individual 'super-user'. While this design perspective did not deliberately exclude the interests of other stakeholders, it may have had an effect on takeup, and certainly precluded any fostering of ownership through participatory design. Under these conditions, a nominally user-centred approach would seem to be effective only if combined with experience of similar initiatives, probably from outside the organisation. An analogous point, this time about technology versus organisational strategy led developments, is made in the Yetton, Johnson and Craig (1994) case study discussed in section 5.4.5.

5.4.4 The process of conducting a CSCW pilot

The lessons learnt about the process of conducting a CSCW pilot are an important output from this part of the work. Although the limited scope of this particular exercise was determined by the structure of the DUCK research project, there are a number of points of more general application.

Boundary management

Users have much less incentive to take up groupware tools and new processes if many routine interactions are outside the boundary of the technologically supported work group. Pilot projects should therefore ensure as far as possible that the group's tasks are contained within the boundary supported by the pilot. Potential users are often members of multiple work groups, so even if an individual's interactions for one particular set of tasks are wholly within a 'supported' group, she may spend much of her time working with others in a more traditional manner.

Starting afresh

The temporal boundaries of a pilot project also require management. A good time to start a technology trial may be when a distributed work group has just formed – although both the evidence of the Metre interviews and the published literature suggest that collaborative tools are not ideal for forming new relationships - or when an existing group starts a new project. In each of these cases there is a reasonable chance of integrating the tools into working practice right from the start, and of involving target users in process and tool design.

Thought must also be given to the termination of the pilot. In the current case, the structure of the wider DUCK project entailed removing the technology after some 5 months. The close-down coincided with a change to less intensive distributed work, but users' comments and informal observation indicate that some team members were just beginning to have sufficient slack time and confidence to explore the full scope of the technology. Certainly careful judgement is required as to when to discontinue an experiment which has been initially perceived as relatively unsuccessful.

User support and liaison

Naturally the plans for any well-thought out pilot will include provision for the training and support of the user group at the beginning of the trial period. Less immediately apparent is the inconvenient reality that the population of work groups changes, and so training and initial support must be available throughout the lifetime of the pilot. This implies that those managing the pilot need to keep themselves informed of personnel changes - another reason for this is that outgoing staff should be contacted for evaluation purposes. And as well as newcomers who are full members of the work group, there will be others with a sporadic role in the group's tasks. In the Metre case such people included an accounts assistant and a financial controller. One way of dealing with these issues, not explored here, is to train local experts in the support of the system, but it should be remembered that such support duties can be a significant overhead.

5.4.5 Uptake issues in CSCW - the context of other case studies and related literature

How do the results fit with the findings reported in the CSCW literature? The following extract is from Bowers (1994), describing a CSCW initiative in a UK government department.

> *The CSCW Network has settled into a fairly stable pattern of usage over the last nine months or so. Person-to-person electronic mail is extensively used on a day-to-day basis by a 'hard core' of six Network users... Recent interviews with users... reveal a continued reluctance to use or experiment with the other applications on the Network, except in a casual fashion. Certainly, for many users, it has only been the electronic mail facilities which have even come close to being part of their daily working lives. The applications more closely related to or inspired by CSCW research have remained, as one user put it, "exotica".*

The comparisons with the results of the pilot are striking, the more so since the applications discussed by Bowers included Lotus Notes™ and an application sharing tool. However, there is one important difference between the two user populations – Bowers' study was located in an organisation specifically responsible for exploring new computer developments, many of whose staff were specifically interested in CSCW. For a variety of organisational and technical reasons, even these people - who might be expected to be pre-disposed to experiment with these sort of tools - used only the most mundane of applications.

More generally, a survey of 200 UK companies by the PA consulting group (PA Consulting, 1995) suggests that only 12% of companies had achieved significant benefits from groupware. Similar evidence is reported in the academic literature. Bullen and Bennett's (1991) review of groupware take-up and use across 25 US organisations includes a wealth of observations and user comments about partial exploitation which could apply, without any rewording, to the Metre experience. Similar points are made by a number of other authors, including Grudin (1988); Kraemer and Pinsonneault (1990); Brunet, Morrissey and Gorry (1991); Cool *et al.* (1992); Orlikowski (1992); Sanderson (1992); Grudin (1994) and Sanderson (1994). Olson and Teasley (1996) report on a year's experience with CSCW applications in a distributed automotive design environment, again finds only partial use. The team was originally supported by desktop video, supplemented by an object camera, an electronic whiteboard, (video and whiteboard were part of the same product, and rarely used because of problems with set-up and speed), shared window software (of limited use, again because of speed problems) a version control system, email, a company-wide shared diary, telephone, paper memos and a shared file server. These tools were supplemented during the period of the investigation by tailored Lotus Notes™ databases. Notes subtly changed some aspects of the work, particularly with respect to commitments and responsibilities, and gradually fell into disuse. Reasons for this were ascribed to the extra work entailed, the unwelcome potential for monitoring by management, and to some extent, the lack of incentive to contribute to the databases. However that this selective use of system functions is by no means confined to CSCW tools, but has been established for some time as a general characteristic of IT implementations: Eason (1989) quotes the example of a system for bank staff where of 36 available functions, five accounted for 75% of the usage.

Overall, while the lack of machines was one major influence on the low usage rate, the results of the pilot confirm the reported findings of other studies: perceived (non-)ease of use was a significant factor in discouraging take-up; the tool which most closely paralleled a non-electronic task (sending a file by email) was the only one to be used seriously; users recognised that process re-engineering (which they did not have time to undertake) would be necessary to exploit the tools to the full and finally, take up suffered because the technology was isolated from existing technical and organisational systems.

Uptake – CSCW models

Two overall models of uptake in CSCW have had a pervasive effect on work in this domain. Firstly, the notion of 'technological frames', Orlikowski and Gash (1994), conceptualises differential uptake between stakeholder groups in terms of the understanding of purpose and functionality (or 'frame') which users have of technology. In the exploitation of Lotus Notes™ in a management consultancy, for example, some users viewed the databases as personal productivity tools whereas others saw them as a repository for shared information. In our case, there is some evidence to suggest that users conceptualised the tools as a means for improving the transmission of materials for co-working rather than for cooperative design itself. For example, users would telephone someone at the other site to warn them that a file was about to be sent, email the file, then telephone again to discuss the contents of the file rather than viewing and editing the material together using the application sharing tool. However, incidents such as these may be just as easily explained by the perceived usability problems.

Grudin (1994), further developed in Grudin and Poltrock (1997), proposes eight non-technical "challenges to designing, developing and deploying groupware". These are summarised below, accompanied by a consideration of their application to the Metre case. It can be seen that while all the 'challenges' were evident in the experience of the pilot, the list does not extend to the problems arising from the way in user tasks were interwoven with other collaborative work which was not supported by the new tools.

Disparity in work and benefit. The new system may require additional work by some individuals and groups who do not obviously benefit from the changes.	Additional work certainly required to learn the system, but there should have been direct benefits from reducing inter-site travel and relocation.
Critical mass. Groupware requires a critical mass of users to be successful. Additionally, overuse may reduce efficacy.	Lack of a critical mass very probably hindered uptake.
Disruption of social processes. Social or political norms may be unintentionally violated, or prove impossible to support in groupware.	Groupware supported the need to share information, but probably not the perceived need to maintain interpersonal relations, especially with people not part of the pilot team.
Exception handling. Groupware may not handle exceptions to normal process and improvisation.	No evidence directly supporting this, in fact the way files were transferred was an improvised exploitation of email.
Unobtrusive accessibility. Because groupware features may be used relatively rarely, they should be unobtrusive, and/or be integrated with normal functionality.	Almost certainly the lack of integration with usual tools and files systems was an obstacle.
Difficulty of evaluation. Because groupware evaluation is difficult, we are prevented from learning from experience.	The pilot implementation took into account the results of other studies but could not always implement emerging 'best practice' even where earlier evaluation results were clear enough to be useful.
Failure of intuition. Intuition is often unreliable in the complex context of groupware: hence bad design decisions are made.	This was most evident in assumptions made about the PC skills of the users.
The adoption process. Introduction requires more care than developers have recognised; important issues also arise around group-by-group implementations.	Here Grudin and Poltrock acknowledge difficulties concerned with the purchase and location of technology for groups which cross organisational boundaries, but do not directly address the important issue of multiple group membership.

Uptake – Information Systems perspectives

The Information Systems (IS) community has much to say about technology acceptance and organisational fit. The literature in this domain is extensive. This section does not attempt a comprehensive survey, but briefly introduces several influential models (selected because of their extensive citation by other authors), together with other recent work of particular relevance and discusses the Metre case in the light of these.

Many of the observations made about the Metre pilot users have striking parallels with the conclusions of innovation characteristics research. Tornatzky and Klein (1982) provide a meta-analysis of 75 papers in this area. Three innovation characteristics (compatibility, relative advantage and complexity) had the most consistent relationships to innovation adoption, where the innovation is, in the broadest sense, any new technology or practice. A similar metanalysis by Rogers (1983) adds triability and observability. IS research has sought to apply these general models and variations of them to the particular context of new technology, leading to the identification of a number of extra factors which are held to influence uptake. (Note: one of the problems in synthesising and applying research in this field is the inconsistent terminology. Terms found include acquisition, adoption, acceptance, implementation, assimilation, routinization, and, of course, use. While it is clear that some can relate only to specific stage – acquisition, for example, is simply purchasing the technology – others are used much more loosely and inconsistently. Adoption, for example, can mean the decision to purchase, or the routine use of technology by end-users. In this account we are concerned with the *use* of technology.)

Much work (of which Hebert and Benbasat, 1994; Miller, Rainer and Harper, 1997; Straub, Keil and Brenner, 1997; Loh and Ong, 1998 are among many recent examples) extends now classic technology acceptance models. These are primarily the work of Davis (1989), who argues that user perceptions of usefulness and ease of use have a critical influence on acceptance and usage and Moore and Benbasat (1991), which include perceptions of relative advantage, compatibility, ease of use, demonstrability of results, effect on image (of the user), visibility, trialability and voluntariness. The work of Loh and Ong on the Davis model has particular relevance for this study. In their empirical study of an Internet based stock trading system, it was found that perceived usefulness had a positive relationship not only with perceived ease of use, but also duration of use and user satisfaction. The Metre results are apparently consistent with both the Davis model and Moore and Benbasat's dimensions of relative advantage, and compatibility, but since there is no hard data available on the *extent* of use it is difficult to be more precise.

Agarwal and Prasad (1998) take a slightly different theoretical base, drawing on the three innovation characteristics identified by Tornatsky and Klein. Their study relates the decision to adopt (in this sense, meaning use by an individual) new technology to perceptions of relative advantage, ease of use and compatibility with existing values and practice. However, two additional moderating dimensions are proposed, the innovativeness of individual users and the media through which potential users become aware of the technology and its features. A field study conducted at a large manufacturing company showed that innovativeness interacted with perceptions of compatibility, but not the other dimensions. Agarwal and Prasad suggest that this may indicate that the cognitive costs of perceived incompatibility are so high that only more innovative individuals will decide to adopt the technology in question. As for media channels, both relatively impersonal channels such as presentations, videotapes, etc. and interpersonal channels had an effect on perceptions, interpersonal contact having the greater direct effect. Since many of the comments made by users in the Metre pilot related to incompatibility with existing practice or values, Agarwal and Prasad's observations may partially explain the low take-up, but this must remain a supposition in the absence of innovativeness data for the users. Similarly the manner in which users were introduced to the technology (some attended an initial group presentation and briefing, others directly inducted by other users or the Metre support person) may well have affected perceptions and consequently use, but again data is lacking.

Other work, notably Swanson's 'tri-core' model of information systems innovation (Swanson, 1994) models factors affecting IT use according to the organisational locus of impact of the technology. The relevant aspects of the model here are those concerning an organisation's administrative and core business functions. For these areas, it appears from Grover's work on the Swanson model (Grover, Fiedler and Teng, 1997), that positive relationships exist between new technology use and (i) the diversity of technology already deployed and (ii) the professionalism of the supporting IS unit. There was also some evidence of a relationship between use and slack resources in the organisational unit in question. Again, the Metre evidence is congruent with these findings. While the characteristics of the IS unit are probably irrelevant in our case, since they were not directly involved in the pilot project, there was little baseline technology exploited by the pilot users, and certainly no slack resources at all. Additional locus of impact related characteristics arise from the meta-review provided by Prescott and Conger (1995), which identifies management support, the presence of strong innovation champions, user training, organisational characteristics and organisational fit as influencing the use of new technologies at the intra-organisational level.

Turning to the *process* of achieving technological and organisational fit, one of the more influential models in the last decade has been the MIT 'Management in the 1990s' framework (Scott-Morton, 1991). The model, shown as (a) in Figure 5.6, emphasises the interdependency of overall strategy, organisational structure, management processes and individual skills and roles in realising the potential benefits of IT. In commenting on the model, Yetton, Johnson and Craig (1994) note the absence of an explicit path through the transformation process, observing that the conventional argument is that an organisation "...devises the strategy, chooses the structure and management processes that fit, aligns IT and ensures that individuals are appropriately trained and that roles are well designed."

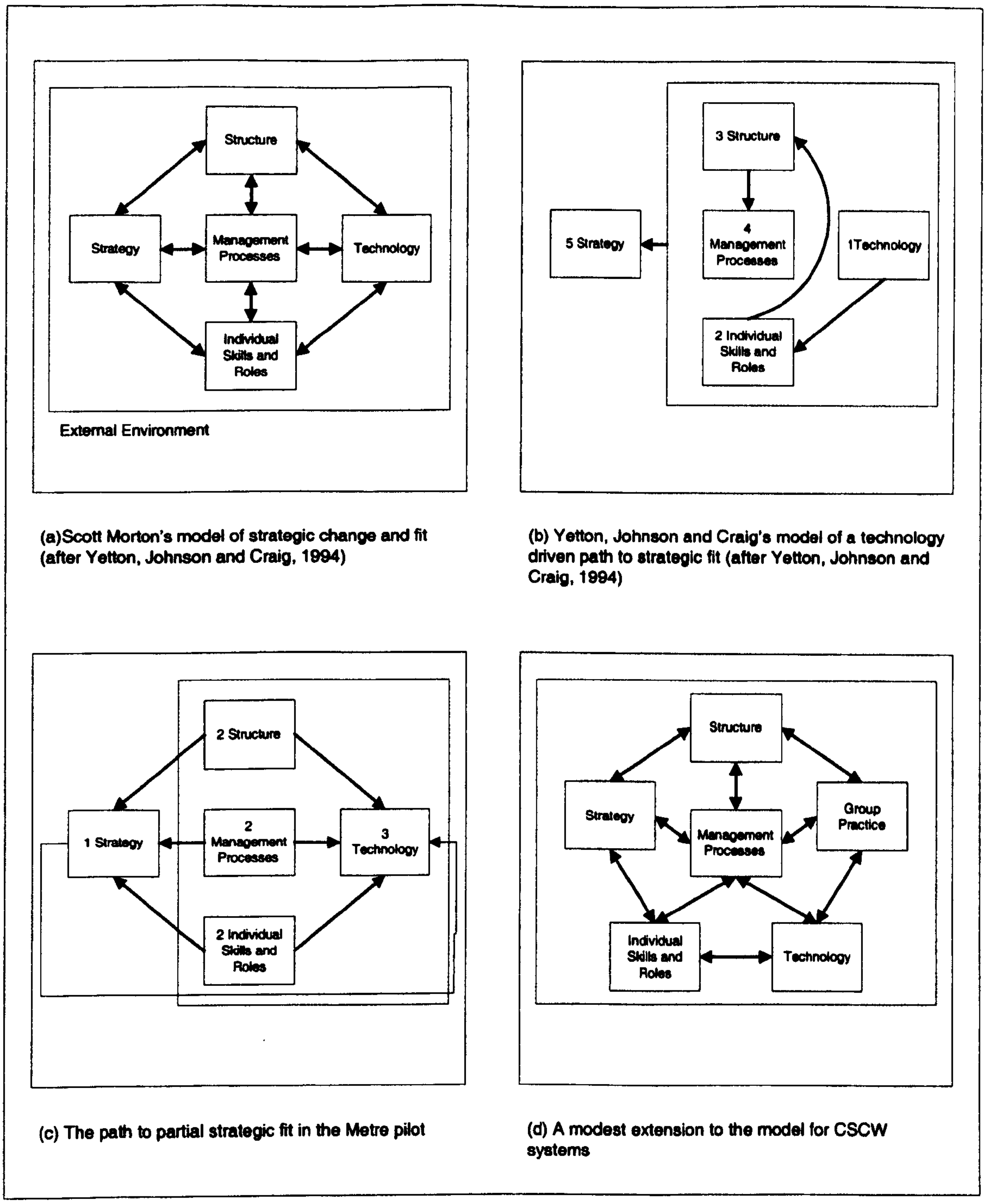
Yetton, Johnson and Craig go on to present a case study that is a stimulating contrast to both the general thrust of IT/IS management literature and the Metre results. The authors discuss the introduction of CAD tools into Flower and Samios, a small architectural design practice of 14 individuals. Since the domain is design, and in the broadest sense, the technology had some CSCW aspects, it is worth considering in some detail here. The firm's partners, initially goaded by the success of a competitor using CAD as a presentation medium, bought two Macintoshes and architectural CAD software as an experiment. Impressed by the potential productivity gains, the senior partners mastered the system and mandated its use throughout the firm on a project-by-project basis. Once a worthwhile number of workstations were in place, a network, a file server, project management software and supporting applications such as costing applications were added.

Some five years after the initial experiment, the practice project portfolio had dramatically increased, and all design and presentation work was done on computer, allowing close co-working with customers and other professionals involved in the design process. There had been neither an overall strategic decision to change the practice of design by introducing technology, nor a careful alignment with current working methods, but rather an incremental, tactical re-organisation of structure, processes and individual roles once the potential of the CAD software became apparent. However, while the change process may have been incremental at an organisational level, it was far from incremental for individuals. Once an architect had been equipped with the new tools, the use of pencil, paper and drawing board were forbidden, to the extent that a newly recruited (and experienced) architect who preferred to use old technology for conceptual work left the firm after a few months. The authors also comment on an overall resistance to computer support in the wider architectural world, stemming from concerns about constraints on creativity – echoing the view expressed by some design engineers at Metre that 'good design isn't done on computer'. In analysing the process of this particular venture, Yetton, Johnson and Craig stress that the initial intervention was through technology, followed by the transformation of individual skills and roles, structural changes and finally re-alignment of management processes. This is illustrated in model (b) of Figure 5.6. The dynamics of this path through the Scott Morton model of strategic change, it is claimed, are very different from the conventional process.

What, then, can be said about the Metre pilot in the context of the generic strategic change model and the experience of the Flower and Samios case study? Considering the generic model first – (a) of Figure 5.6 – it is apparent that the process at Metre addressed some elements only in a unidirectional manner. Despite the recognition in the socio-technical approach that human and organisational aspects of a system are integral to the redesign process, any significant changes to roles, structure and management processes were out of the question. Generic knowledge from the requirements work, and the rather sketchy information about the pilot teams themselves explicitly influenced the design of technology, but there was no locus for the any complementary re-alignment of organisational aspects. The process model in the Metre pilot is shown as (c) of Figure 5.6. Hence the eventual 'fit' was approximate at the very best.

But what of the strong version of technologically mediated change described by Yetton, Johnson and Craig above? The cultural stance towards IT at Metre was ambivalent, and

CSCW implementations are acknowledged to be complex and unpredictable (*c.f.* Grudin's points 6 and 7 above), even with the fullest information about current group practice, so at first sight a technologically driven approach seems attractive. But the tactics at Flower and Samios evolved around a highly incremental process. Not only were the CAD tools introduced on a project by project basis, but also to each architect in turn within projects. In this way, adjustments to practice could be made on a basis of solid experience, and the technology itself tailored to reach its full potential. This worked well, but the CSCW aspects were not core to the main purpose of the system. In the Metre case, of course, the whole point was to support collaborative working over distance and it is difficult to conceptualise an individual-by-individual implementation of groupware tools. There may well be advantage to leading with technology in some circumstances, but the experience at Metre tends to suggest that effectiveness of this strategy in an all-at-once implementation is limited. Organisational change for CSCW seems to be modelled more appropriately by the original model (a) with a modest extension to include attention to group practice, shown at (d) of Figure 5.6.



(a)Scott Morton's model of strategic change and fit (after Yetton, Johnson and Craig, 1994)

(b) Yetton, Johnson and Craig's model of a technology driven path to strategic fit (after Yetton, Johnson and Craig, 1994)

(c) The path to partial strategic fit in the Metre pilot

(d) A modest extension to the model for CSCW systems

Figure 5.6 Models of strategic fit for CSCW, extending Scott Morton (1991)

So is CSCW fundamentally different in kind from other IS initiatives, or can it be considered as a special, rather more complicated case of IS implementation? Considering the Metre pilot and other CSCW case studies in the context of the IS literature tends to support the latter view. For CSCW, the issues of user perceptions, acceptance and organisational change are more complex because different stakeholder groups must be considered, and because incremental introduction is not a realistic option. In the case of cooperative design, the change process is particularly intricate, because designers are frequently members of multiple groups, work on multiple projects and may be part of a culture which holds that the nature of design work itself is fundamentally unsuited to computer media. However, even in this domain there appear to be no adoption features which have not been observed elsewhere. In creating an overall model of uptake for CSCW, the next step for researchers is to identify *which* of the many dimensions and process characteristics noted above are most salient for CSCW in general, and individual domains and contexts in particular. The body of CSCW case studies is now sufficiently rich for this to be a realistic venture.

5.5 In summary

The work reported in this chapter and Chapter 4 produced a snapshot of requirements and expectations for CSCW at Metre and then used this information in the implementation of a pilot trial of collaborative technology by a distributed design team. A number of findings were made about the features of cooperative design work in this context.

- There is evidence of a conservative attitude to computer based tools for design which spills over into a general scepticism about collaborative technology, especially the support of synchronous work. More specific doubts concern the ability of simple collaborative tools to cope with design artefacts such as one-to-one plots which afford quasi-direct interaction with the representation. Such perceptions and expectations appear to be related to subsequent uptake of technology, confirming the findings of earlier studies.

- A related feature is that computers are often shared resources used for specialist analyses (for example finite element analysis) rather as than the environment for most of an individual's work.

- Within this overall pattern, senior staff are more positive than junior staff about the potential for CSCW, seeking ways of reducing travel for meetings. Junior designers perceive their work as a series of individual tasks, and are therefore more concerned about better access to remote information resources.

- Design tasks on any one project are embedded in the wider organisation in several ways: design tasks are subject to general project control procedures; designers may work on more than one project at a time, and mix design and non-design work.
- Uptake of collaborative design technologies appears to be a special, somewhat complex case of IS acceptance in general.

6 Understanding cooperative design: the small group level

This chapter:

- places this study of small group design meetings in the context of the field work at Metre and of other studies (sections 6.1.1 and 6.1.2);
- describes the design team studied, and the data collection methods used (sections 6.1.3, 6.1.4 and 6.2);
- analyses issues of representation, workspace, gesture and process (sections 6.3, 6.4 and 6.5);
- reviews methodological issues in the analysis of small group design work (section 6.6.3;
- summarises issues for the design of technologies to support distributed design meetings, grounding these issues in a scenario (sections 6.6.1 and 6.6.2)

6.1 Introduction

The requirements and contextual study reported in Chapter 4 elicited information about how design is practised, supported and structured in its organisational context at Metre. It also discussed the initial expectations and requirements of designers and their colleagues for technology to support distributed design groups. Next, as reported in Chapter 5, a pilot study was carried out to discover how a suite of existing tools tailored to meet these generic requirements would be exploited by a particular design group. The original requirements were revised in the light of the pilot study and conclusions drawn about the fit of existing technology with design practice and the process of introducing such technology.

The initial requirements work at Metre had shown that many staff were involved in cross-site projects, that frequent travel was necessary for meetings on such projects and that this was perceived as an unwelcome overhead (sections 4.2.2 and 4.3.1). From the questionnaire came the information that, despite this, CSCW tools for synchronous co-working were not viewed with any great enthusiasm (section 4.4.2). And in the pilot, conducted with a cross-site project, the tools supplied were not used for any form of distributed meeting (section 5.3.2). All this raised important questions about the nature of design meetings themselves and how they might be better supported when designers could not be physically co-present. As will be demonstrated in section 6.3.1, small group design work is an instance of a distributed cognitive system (Hutchins, 1995). Such systems depend on the particular properties, or affordances, of the representational and communications media used, and therefore changes such as an increased use of new technology inevitably change the properties of the system itself. A deep understanding of how such work is currently carried out is therefore essential.

Because of the pressing demands of the Metre designers' everyday work, and the security classification of their subject material, it was impossible to carry out a close study of their work in small group situations. The core data for the analysis was therefore obtained from a series of meetings held by a real-world team of software designers outside Metre[28]. Some aspects of this have been extended by data from a group of engineering product designers, reported in the studies contained in Cross, Christiaans and Dorst (1996a) and further analysed here. Further comparison data has been extracted from the reported literature, in particular Tang (1989,1981), Olson *et al.* (1992a), and Bekker (1995).

6.1.1 The context of previous work

Chapter 2 reviews studies of design meetings and related activities, observing that:

> the evidence about structure in group design problem solving is equivocal: Curtis, Krasner and Iscoe (1988); Olson *et al.* (1992a); Olson *et al.* (1996);
>
> design communication in meetings may be treated as an instance of distributed cognition (Flor and Hutchins, 1991), exploiting representations in both 2D and 3D artefacts (Harrison and Minneman, 1996) and instantiated in the available interpersonal communication channels as part of the cognitive system (Mazijoglou, Scrivener and Clark, 1996);

[28]At MARI Computer Systems Ltd, an organisation which carries out commericial and grant-funded software R & D.

in particular, design representation is carried in gestures (Tang, 1989, 1991; Bekker, 1995) and "embodied actions" which exploit a wide range of workplace artefacts (Robertson, 1997);

design concepts may move between public and private space (Tang, 1989, 1991; Minneman and Bly, 1991);

in real-world projects, both formal and informal communication extends beyond the design team: Ancona and Caldwell (1990); Powrie and Siemieniuch (1990); Pycock and Bowers (1996); Perry and Sanderson (1998).

design decisions are influenced by organisational factors beyond the design problem itself: Carstensen and Sørensen (1996); Potts and Catledge (1996).

However, it remains the case, as Olson *et al.* observed in 1992, that there are few studies extant of designers at work in face-to-face meetings in their everyday contexts. The literature cited above, the results of the Metre pilot and experience in working with designers suggest that that the everyday environments of design teams introduce elements that are not usually present in controlled conditions. These include an evolving design history; participation in larger tasks beyond the responsibilities of the design team under observation; the planning and coordination of continuing activities; the recording of information for future use by oneself or others. As will be seen, these features introduce a wider range of representations into the distributed cognitive system than are normally present in more constrained tasks.

There is a further issue in the review of design meeting studies: it is clear that not only context but also the tasks undertaken differ considerably. This is not just a matter of domain, but related to the way in which meetings are often situated in other, continuing work. In Metre, for example, designers engaged on SUB2 estimated, using the Van den Ven index described in section 5.3.1., that only 30.51% of their work took place in meetings. Subsequent work with a team of software designers at Metre (Turner and Turner, 1997, not included in the current study) similarly established that 17% of their work was meeting based, while 70% was entirely independent. Potts and Catledge (1996) characterise the overall group design process as an 'Inquiry Cycle' of three activities "the expression of design ideas (e.g. in written documentation), the discussion of these designs (e.g. in informal review meetings) and the refinement of the expressed designs as a result of decisions reached during the discussions." Experience and consideration of the literature suggest that the tasks of many design meetings, including those analysed here, are a mixture of informal review and related design development. Other types of meetings brainstorm possible solutions *ab initio*, while still others take the design from initial problem to design specification in a single session. Thus in comparing studies, it is important to consider the underlying tasks of the meetings studied.

6.1.2 The focus and approach of this study

In complementing existing studies, the work reported here concerns meetings which were part of a real design project and were largely concerned with informal review and associated design development. The focus is primarily on the role and use of the workspaces which form part of the distributed cognitive system. Other, subsidiary aspects are also investigated to enable a more complete analysis of meeting features and comparison with earlier studies. The techniques adopted are in the spirit of the contextual analysis approach to human-computer interaction design (Holtzblatt and Ramey, 1993, 1996; Suchman and Trigg, 1996) and the ethnographic tradition in CSCW studies (for example Hughes *et al*, 1994; the review in Plowman *et al.*, 1995 and Robertson (1997b). In particular, the work follows Olson *et al.* (1992a) in their approach to the observation of software design meetings and uses their analysis scheme for design process (of which more later). However, instead of sampling meetings across organisations, this study takes a series of meetings held by one team over the early design phase of a project. The focus of the current work is also different in that issues beyond design process are considered.

Issues for analysis

In the light of the Metre study reported in Chapters 4 and 5, the underlying premise was that existing technology would only partially fit requirements. In this case, this was because of the rich opportunities for design representation and communication afforded by face-to-face meetings. Specific issues for analysis were identified *a priori* from the literature, and are listed below, together with a brief indication of how the current work either extends existing material or validates it in a new context:

the role of distributed representations and multiple workspaces in design meetings *(this complements Tang (1989, 1991) and the studies reported in Cross, Christiaans and Dorst (1996a) by studying real-world meetings over a period of time and introduces a more abstract design domain, also applies the Hutchins (1995) concept of distributed cognition to this context;*

the role of gesture in navigation amongst representations and workspaces *(this complements Tang (ibid.) and Bekker (1995) by introducing a focus on this particular role of gesture and is also complementary to Robertson's (1997) concept of embodied actions);*

the relationship of design process to exploitation of design workspace *(not studied in depth elsewhere);*

changes in meeting design process with design development *(this is primarily a consistency check for behaviour between meetings and with other studies, also complements Olson et al. (1992a,b) over a more prolonged series of meetings).*

A further issue was emergent in nature, arising from repeated viewing and consideration of the material in the video recordings that comprise the case study data:

the fluidity of boundaries between public and private workspace.

In general, the data captured and analysed in this part of the work affords an opportunity to specify for the design context the general observations about the electronic support of interpersonal communication made by Fussell and Benimoff (1995). The table overleaf summarises the salient dimensions of the current study and the most closely related earlier work on face-to-face design meetings. As will be clear from the rest of this chapter, the social aspects of design meeting work have not been considered in any depth. This is not to say that these issues are unimportant, but they are not central to this consideration of the consequences of distributed design representation.

		This study	Olson, G *et al* (1992)	Olson, J *et al* (1992)	Tang (1989, 1992)	Bekker (1995)	Cross, Christiaans and Dorst (1996a)	Robertson (1997)
Features studied	Process/structure	✓	✓	✓		✓	✓	
	Workspace use	✓		✓			✓	✓
	Gesture/action	✓			✓	✓	✓	✓
	Interpersonal behaviour/role						✓	✓
Study type	Controlled			✓	✓	✓	✓	
	Real World	✓	✓					✓
Domain	Abstract e.g. software design	✓	✓					
	Intermediate e.g. user interface design			✓	✓	✓		✓
	Concrete e.g. engineering design							
	Concrete & physical artefacts						✓	
Meeting type	Single meeting (self contained task)			✓	✓	✓	✓	
	Series of meetings (continuing task)	✓	✓					✓
Task type	Brainstorming/early idea generation			✓	✓	✓	✓	
	Design development	✓	?	✓			✓	✓
	Reporting & reviewing	✓	?					✓
Group type	Peers		?	✓	✓	✓	✓	
	Team with leader	✓	?					✓
Subject type	Students			✓	✓	✓		
	Professionals	✓	✓				✓	✓
Technology type	With meeting technology			✓		✓		✓
	Without meeting technology	✓	✓		✓		✓	

6.1.3 About the 'Mallard' design team

The software designers for this case study were the three members (project manager and two senior software engineers) of the Mallard team at a small software house specialising in research and development. At the time the case study started the project manager (Roger)[29] and one of the team members (Peter) had worked together on the project for almost 4 months. The third person (Matt) while, theoretically attached to the project, had been engaged on other work until about 3 weeks before the start of the case study period. Roger and Matt had worked at the same site for some years, while Peter had moved from another site not long before he started work on the project.

Roger, as the most experienced designer and project manager, took the lead in meetings, setting the aims, keeping discussion focused and allotting tasks for the next stage of work. He also spoke most in the technical discussions. Peter and Matt were also experienced designers. Their roles as team members were notionally identical, although they worked on different design sub-problems. It can also be observed from the video record that Matt tended to speak less than the other two, but was the group's primary note taker. For the two meetings that have been examined for speaker participation, meetings 1 and 3, the percentage of utterances voiced by each of the team are shown in the table below. (Methods used for analysis are described in section 6.1.4.) Since the interpersonal aspects of group interaction were not an area of focus for this study, role issues have not been further analysed.

	% utterances	
Speaker	Meeting 1	Meeting 3
Matt	20.09	6.37
Peter	37.05	43.56
Roger	42.86	50.07

Mallard was a collaborative research project that addressed software systems integration by enabling system testing and integration activities earlier in the project life cycle. The aim was to provide visibility of evolving software products, and an integrated product development process and tools. The team's particular responsibility was a configurable graphical animation workbench for the visualisation of the specification and design notations used in the development process. At the time of the case study, a specification

---

29 All proper names have been changed.

had been completed and the team were working on the high level design of the animation tool: the output of this phase was to be a high level design document for consideration by the wider consortium for a fixed deadline. The organisation of the project entailed cooperation with other partners about some aspects of the design, but this case study concerns work within the team. Their task was reasonably typical of the software industry and while Mallard was a part-funded (by the EC) research and development project, all the team had substantial experience of straightforward commercial work.

Activities during the two weeks of the study consisted of periods of individual work, some of which related to other project matters, interspersed with team meetings. The team's desks were located close together, in part of a large open-plan office. There was frequent informal communication between team members, particularly between Matt and Peter - I could readily observe this since I worked in the same open-plan office. However, prolonged whole-team discussions tended to disturb other people, so these were held in the meeting room and provided the data for this study. This small meeting room was equipped with a large table and chairs and two portable whiteboards and separated by glass partitions from the main office. Interestingly, Roger commented that had the team occupied their own room, as had been the case on previous projects, there would have far more *ad hoc* discussions and far fewer meetings. In this way the team's working practice may be closer to designers cooperating over distance – who have little scope for opportunistic discussions – than to designers working together in the same room.

6.1.4 Data collection and analysis methods

Five Mallard design meetings were video-recorded, resulting in over 7.5 hours of tape which form the basis of the ensuing analysis and discussion. Data could not be collected from all meetings: the team could not be expected to schedule all their meetings for when I was available and indeed this may have distorted work patterns. The choice of meetings was therefore opportunistic, but there is no reason to think that the selection was atypical. Indeed evidence to be presented later suggests that in many aspects the team's behaviour followed a consistent pattern across meetings. The meetings studied were held in the usual room, with the usual range of furniture and equipment, supplemented by the video camera. The camera was positioned to take in the whiteboards and the team members as they sat around one end of the meeting room table. The only special arrangement for the benefit of the study was that the no-one sat at the end of the table nearest the camera. The field of view is as shown in Figure 6.1.

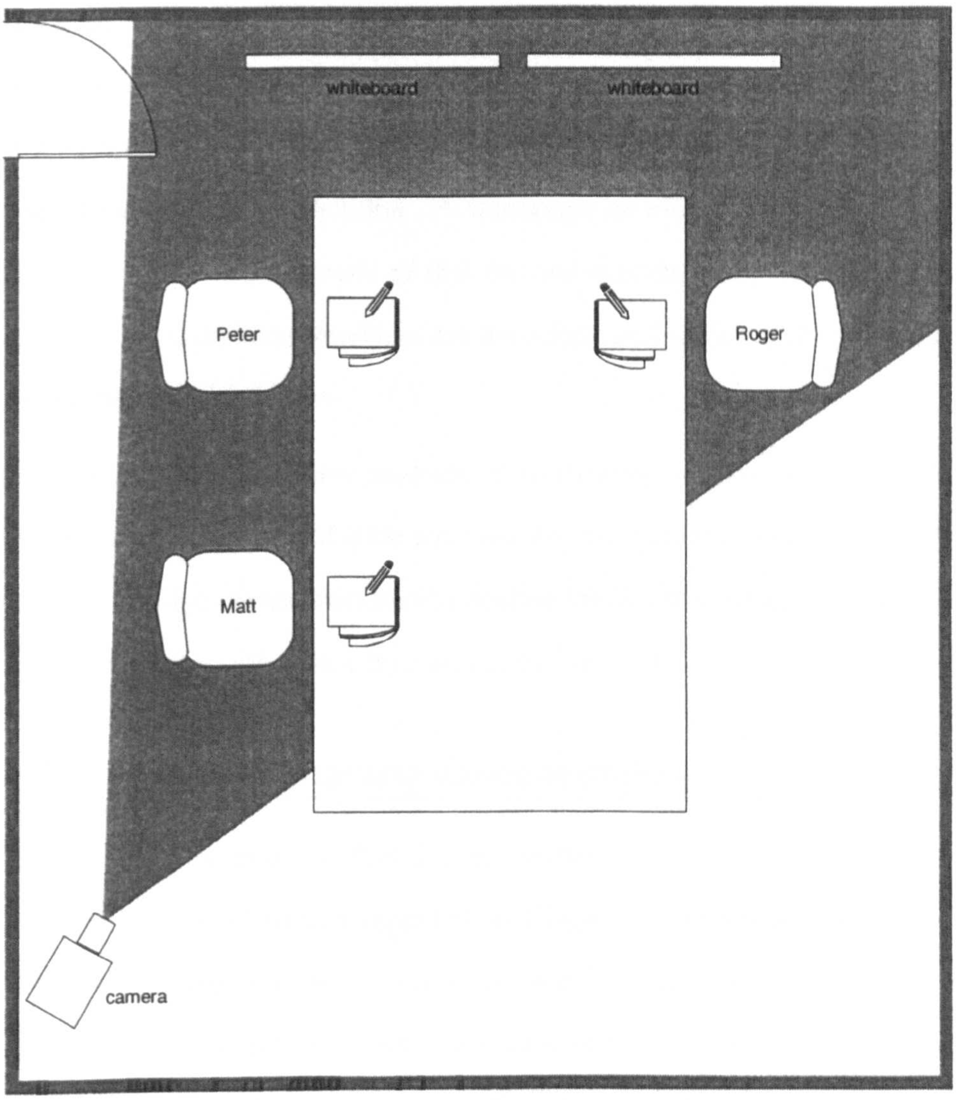


Figure 6.1 Layout of meeting room, showing the habitual positions of team members and the field of view of the camera (shaded in grey).

Apart from an early meeting (not recorded) which I attended for familiarisation, I was not present during meetings except when changing tapes. The team appeared to ignore the camera after its introduction. When asked, they considered that their behaviour had been much as normal, although Roger did observe that in the first recorded meeting (only) there had been some tendency on his part to 'perform' for the camera by choosing words rather more carefully than usual. Tang (1989) supports this, noting evidence in the psychology literature (Kelley and Thibaut, 1969) that the initial effects of being observed fade quickly with time. Of course, the mere fact of being studied may have influenced behaviour but it is difficult to quantify what the consequences of this might have been.

Team members were asked for permission before any recording took place. They also signed a consent form to the effect that the video and audio data could be used for research purposes by myself and others closely associated with the work[30]. This procedure follows the recommendations of Mackay (1995). The team was questioned briefly after each meeting to ascertain the general purpose of the session, its conclusions and the consequential actions. A transcript of each tape was made, thus producing a verbal protocol – a example of this for one meeting may be found at Appendix E. The methods for individual analyses are described and their results reported and discussed in the sections which follow.

In addition to the intensive analysis of the meeting record, each team member was briefly interviewed at the end of each working day to establish what tasks he had been engaged upon, and what communication he had had with other team members. This was the only intrusion into an otherwise normal pattern of work.

6.2 The Mallard meeting series in context

It has already been noted that design meetings are islands in a stream of other activities. Moreover, as the studies reported in Chapter 2 showed, design decisions are rarely resolved in the narrow context of the task-in-hand, but must take account of organisational constraints and work undertaken on other parts of large projects. An evolving design frequently exists before the start of a meeting, and is usually documented, while numerous other documents embody design history or details of related parts of the project. Equally, information must be captured during the meeting for the designers themselves or others in the next stage of development. Mallard displays many of the characteristics of this pattern of working. Protocol fragment 1 shows Roger, towards the end of meeting 3, organising the making of a meeting record. He is referring to the work the team has done on the whiteboards. Note that the record is not just for the

---

[30] The form also provided for consent to using the tape for illustrating conference presentations and similar purposes. One team member exercised his right to refuse this.

team themselves, but forms the basis for co-working in the wider collaborative project. Deadlines for the wider project are driving the design process of the Mallard team.

OK. I'll record all that. What I'd actually like is for somebody not only to record that right hand one but draw it on some tool or even on a piece of paper that we can photocopy, but without any further refinement or thought so that it's a shared record. I think that one's an important one because we might have to come back to it at some point. And then on Monday we have to make some definite plans for actually wrapping this up and getting it into a document to send out on Wednesday I think. Right. It's going to be a rush job but I have to have something. I have to be able to talk about this and make decisions with people at the meeting, which is unfortunate, because it would be nice to involve the team on that but I think we need to have something in writing to give them ample preparation time. And the sort of thing we can have, it can start with the five layer thing we've got here, perhaps six with the meta-model.

Protocol fragment 1. Making a record of a meeting.

In meetings that not entirely mediated by new technology, all this documentation inevitably means many different paper resources are referenced - and often annotated - both individually and collaboratively.

The interweaving of group and individual tasks also influences the type of design work undertaken in meetings. As Figure 6.2 below shows, the Mallard meetings all drew on individual work by team members. The protocols show that typically, one of the team would explain the ideas he had developed, while the others questioned, argued and sometimes suggested alternatives. Protocol fragment 2 shows a sequence of such interactions from meeting 3, where Roger is explaining his treatment of scheduling in the animation tool. Text in square brackets shows murmurs of acknowledgement from other members of the team.

Time	Who	Utterance
1401	R	Then scheduling may be dealt with entirely at this layer [P: mmh] but if it's not, it passes through without any transformation of the [P: mmh] syntax - written on the fly - there's something optional there. I think we've got a user model to concrete model mapper here [P: mmh; M: mmh] and potentially a value mapper if there's a display format which is different to an internal format and I've asserted that. [P: yeah] So we've got user model... abstract user model to concrete user model value and we've got abstract user model to concrete user model element in that loose sense. I know we're calling them entities and symbols for these... [P: mmh] So I think I'll have a similar structure here. The request both has to map values and things [P: mmh] but the result can be entirely in terms of things [P: mmh] and of course...
1402	M	You said you had a...*indistinct*... Does that mean that the display converter has to be at that level?
1402	R	It depends what the display converter is doing.

Protocol fragment 2. Explaining a design concept which had been developed before the meeting.

Figure 6.2 shows the Mallard series of meetings in the context of the individual tasks undertaken by team members as reported at the end of each day. Shaded areas indicate meetings which have been recorded and analysed, while the dotted lines between the columns for each person indicate co-working. Most primary design work is undertaken by Matt and Peter, while as project manager Roger spends more time on other project tasks. (All three team members had significant expertise in the design domain in question.) Matt and Peter were functionally equivalent in the team structure. As can be seen, the team had a regular cycle of individual work interspersed by meetings. Individual tasks generated paper notes (both hand-written or printed) or more formal printed documents which were commonly used during meetings, and referenced meeting notes, document annotations and occasionally the contents of the whiteboards. Other occasional meeting resources were printed documents produced before the period of the study, and the whiteboards with material generated in a previous meeting.

Compare this situation with the environment in which more controlled studies have been conducted. Designers here are often working entirely within the compass of the meeting session: ideas are developed *ab initio*, and there are generally fewer resources for reference. Tang's (1989, 1991) user interface designers worked with large, shared sheets of paper, supplemented in some cases with private notebooks, but with no reference materials; the automatic post-office designers studied by Olson *et al.* (1992b) and Bekker (1995) were supplied with whiteboard and paper, but again no reference resources; however, the product designers who provide data for the studies in Cross, Christiaans and Dorst (1996a) worked in a rather more realistic set-up in that they had a number of briefing documents, but were not concerned with noting results. As for the requirement to record information for use outside the meeting, four of the eight groups observed by Tang copied their paper sheets for use in further development of their project, while the other studies cited used completely self-contained tasks. The section immediately below discusses the implications of the multiple design representations contained in the aforementioned notes, documents and whiteboards.

Date	Meetings	Matt's work	Peter's work	Roger's work
26/9	overall architecture			
		completion of interface definition document	consideration of current design document	consideration of current design document
28/9	design scenario		(recorded series meeting 1)	
		design of objects and actions in light of meeting	design of objects and actions in light of meeting	*other project work*[31]
29/9	early design ideas			
		building software obtained over the net	building software obtained over the net	*other project work*
		design issues discussion	design issues discussion	reading M & P's documents
		document discussing issues raised at meeting	document discussing issues raised at meeting	
		reading P's document	reading M's document	
30/9	object design, scenario refinement		(recorded series meeting 2)	
		events discussion	events discussion	*other project work*
		events design	note about design implications	reading P's note about design implications
		demo of StP by R	demo of StP by R	demo of StP by R
5/10	design details			
		discussion of object modelling	discussion of object modelling	discussion of object modelling
		internal interim design document on own area	internal interim design document on own area	*other project work*
		reading documents	reading documents	reading documents
7/10	scenario refinement		(recorded series meeting 3)	
10/10	progress meeting			
		initial work on final design document	initial work on final design document	documenting conclusions from meeting 3
10/10	document structure		(recorded series meeting 4)	
		design document	design document	reading documents
		reading documents	reading documents	*other project work*
11/10	scenario review		(recorded series meeting 5)	
		modification of design document	modification of design document	*other project work*
12/10	final review			
		final changes to design document	final changes to design document	*other project work*
12/10	design document release			

Figure 6.2. Individual tasks and their relationship to meetings.

31 This included the preparation of progress reports, QA documentation, financial management, attending meetings of the wider project and internal management meetings.

6.3 Design meetings, design workspaces and design process

6.3.1 Representation, cognition and workspace

Where is the design during a design meeting? In the Mallard case, at any given time the current model might be distributed among:

- whiteboard text and graphics, both finished and under construction;
- text and graphics in documents brought into the meeting;
- sketches and notes in notebooks/pads, and on the documents themselves.

All these types of design representation were used in each Mallard meeting. As noted above in section 6.2, the range of media for representation were to some extent determined by the need to develop established parts of the design and to record developments for future work by the team themselves or others working on the wider project. These representations exist in persistent media. However, there are other, transient representations in the ephemeral media of speech and gesture and of course the cognitive representations belonging to each team member. The current design therefore, and transformations upon it, are at any one point spread among a number of distributed representations. Activity in design meetings is thus an instance of distributed cognition, defined by Hutchins (1995) as:

> *... computation observed in the activity of the larger system [that] can be described in the way that cognition has been traditionally described - that is, as computation realized through the creation, transformation and propagation of representational states.*

Here the "larger system" is the design team, the "representational states" the various textual notes, diagrams, sketches, and less concrete forms mentioned above. The consequence is that participants have to maintain awareness of, and navigate between, all these different parts of the cognitive task. Examination of the videotape and the associated verbal protocol shows how the physical representations were distributed among the different physical workspaces, where a workspace is any medium used to develop or represent the design or elements thereof.

The **whiteboard** was used in two modes: (i) to represent design elements deriving from an earlier meeting or session of individual work - individuals might add material to the whiteboard before the others arrived, or transfer it from notes during the meeting, explaining as they drew; (ii) as an exploratory tool for sketching new ideas and modifying old ones. (Contrast this with the observations of Potts and Catledge, 1996, whose designers used whiteboards only as ephemeral shared spaces.)

Printed **documents,** both formal project documents and individual work-in-progress, sometimes with notes made in reviews outside meetings, were used for reference to specifications and to earlier versions of the design. Meetings 2, 4 and 5 all included the current design document itself as an explicit topic of discussion. Sometimes all team members would have a copy of a printed document; sometimes they would all have the same document but different versions of it; sometimes only one or two of the three would have the document in question to hand. As we will see, this meant that sharing of material was necessary from time to time. However, as well as serving as a baseline for discussion, documents were also a resource for active design work, when individuals annotated them with modifications and suggestions during the meeting. When not actually being consulted or annotated, documents were spread out on the table in front, or to the side, of their owner.

Personal **notebooks/pads** were used for comments and notes, often detailed and sometimes explicitly intended for use as a group resource for activities after the meeting.

Alongside multiple representations, several further features of distributed cognition may be observed – the terminology used here derives from the analysis of aircraft cockpit teamwork by Hutchins and Klausen (1996). Work in the group is **socially distributed,** in the form of synchronous collaboration in meetings or less formal discussions, and coordinated independent activities, the existence of which partially determine the organisation of meeting activity. Work is also distributed between the Mallard team and its parent collaborative project, resulting in the task of providing a documented version of the design for outside consumption. There is abundant evidence of **distributed storage and access to information** in the range of shared and private workspaces just noted. This provides a potentially useful degree of redundancy should part of the distributed cognitive system fail – if, for example, the whiteboard is accidentally wiped clear, or one of the team members is called away. The current state of play is stored as **memory in the state of artefacts,** perhaps a manifestly incomplete diagram, as well as in individual memory. Communication within the distributed system relies on **intersubjectivity** – shared understanding – between team members, allowing elliptic references to "what we said yesterday" and a more general mutual awareness. Such mutual awareness is the basis of **shared expectations,** such that when an argument is being developed, for example, questions will be raised if it violates shared understanding of how the model currently works – an instance of self-monitoring in the distributed system. Finally, design proceeds by the transformation and **propagation of representational states** as elements of the developing model move between notebooks, whiteboard and cognitive representations into finished documentation. The rest of this chapter explores just how the interplay of workspace affordances and human action support the distributed activity of design development.

Quantifying the use of workspaces

An analysis of the focus of the team's attention during meetings was carried out to establish the relative proportions of time spent in each workspace and the sequence of transitions. Each tape was viewed and for each 1 minute segment, it was noted whether the group's attention appeared to focus on the whiteboard or on any paper documents. (Analysis by a finer unit of time was precluded by limitations of the timestamping facility of the video recorder.) The focus of attention was inferred from behavioural indicators: the site where writing or drawing was being carried out, the direction of gaze, the referent of gestures or deixis such as 'over there' or explicit verbal reference. Clearly not all gaze, gestures and deixis could be disambiguated, but sufficient information could be extracted to support focus identification. If the focus shifted during the segment, both were noted. Similarly if one team member was consulting documents while the others were looking at the whiteboard, both were again noted. Focus classification was carried out by a single observer (myself). This naturally avoided any inter-observer inconsistencies in the application of the criteria. Each tape was viewed twice and codings checked so as to minimise any intra-observer drift in classification. The foci were then tabulated against elapsed time and real time, for ease of reference back to the tape and the transcript, as shown in Figure 6.3 for the first 90 minutes of meeting 3. Any interruptions to the meeting were also recorded. The representation technique used is an adaptation of that used by Radcliffe (1996).

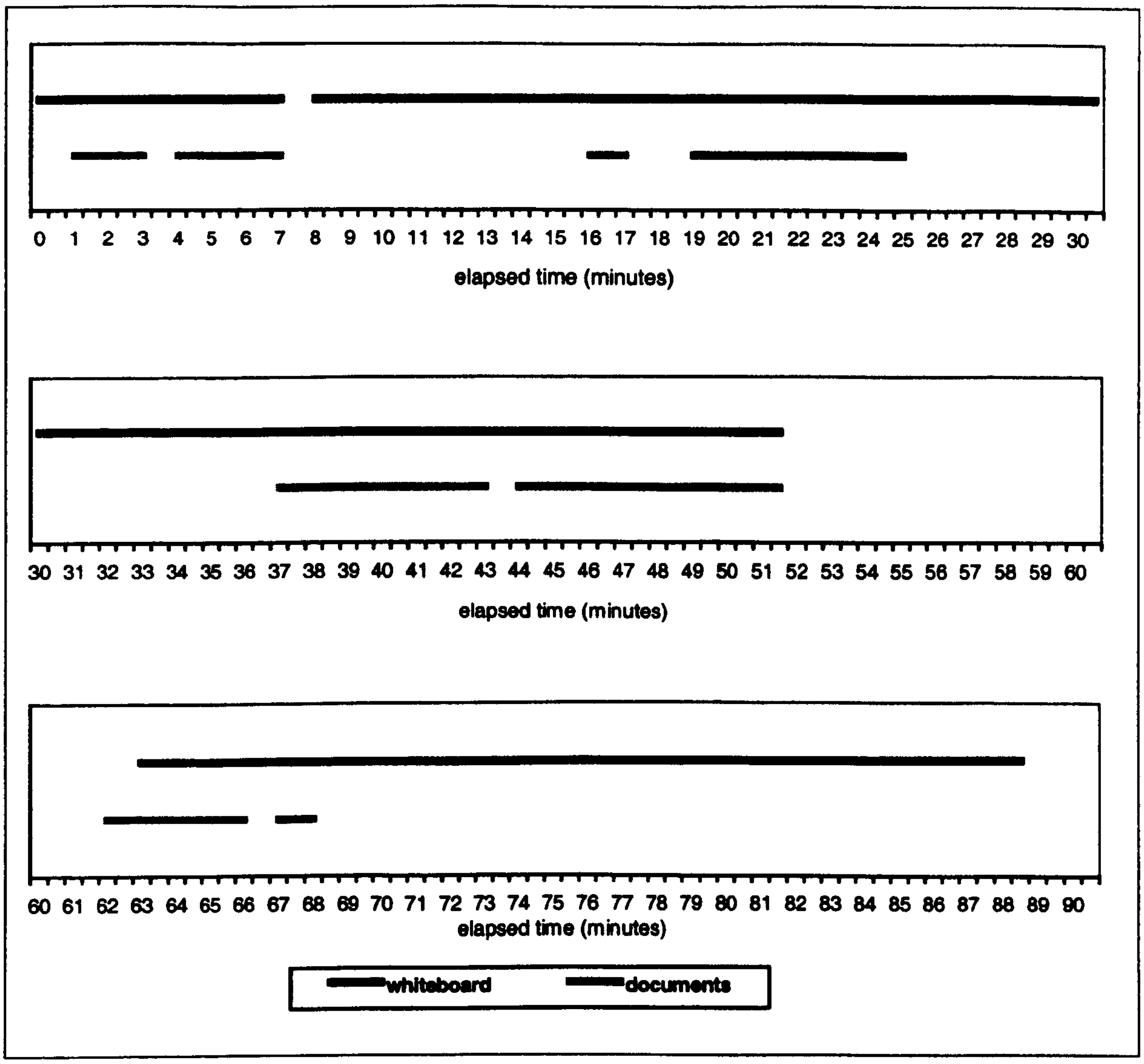


Figure 6.3 Switches in focus over time in meeting 4

When the data is organised in this way, the sequence of switches in focus and the relative proportion of the time focusing on either the whiteboard or paper documents can be seen. It will be observed from this that the focus patterns differ from meeting to meeting. Figure 6.4 shows for the entire series of meetings the percentage of 1 minute segments spent focusing on the whiteboard or on documents and also the percentage of segments ('both') where dual focus on both primary workspaces was recorded. It can be observed that:

- The 'both' category under-represents the number of transitions between workspaces, since there are some segments coded as 'both' where more than one transition occurred, and transitions may in some cases have coincided with segment boundaries;

- Overall, the team spent approximately half the aggregated meeting time in each primary workspace.

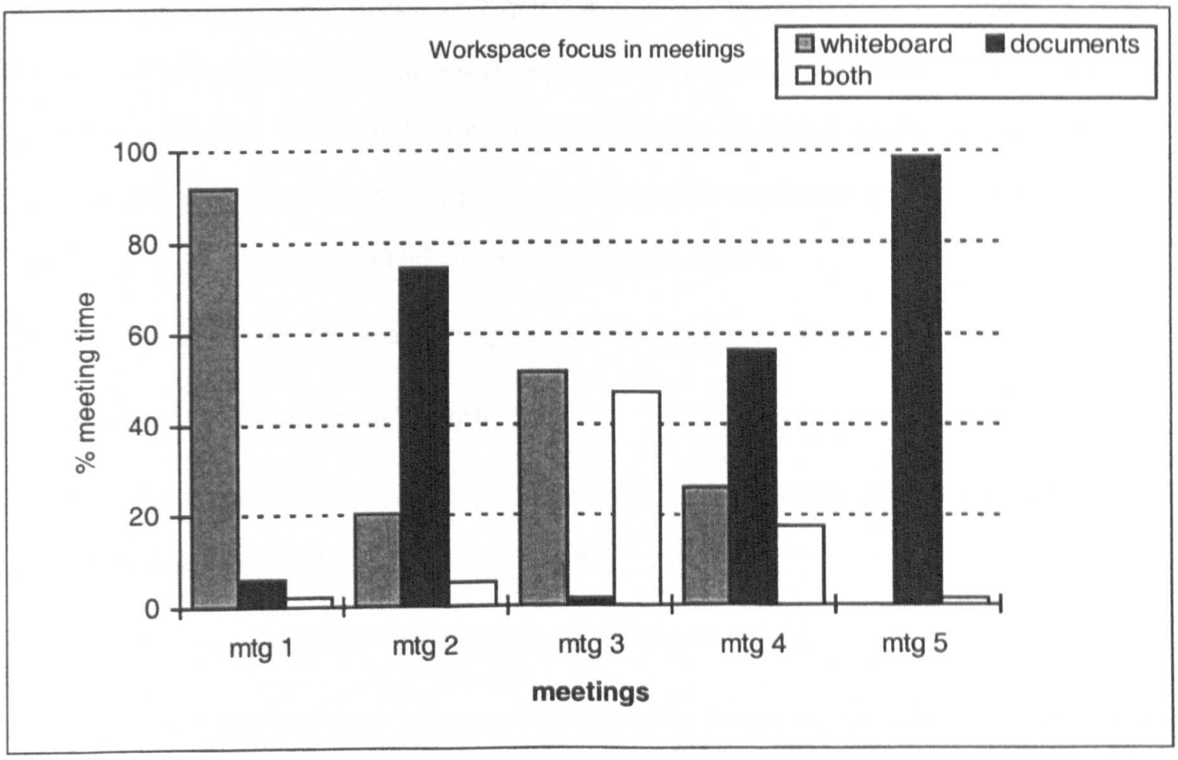


Figure 6.4 Percentage of meeting time spent focusing on each of the primary workspaces

The differences in focus patterns can be explained by the purpose of the meetings and the character of the discussions that took place. (The purpose of each meeting is noted above in figure 6.2.) Unsurprisingly, those meetings where discussion is more exploratory, meetings 1 and 3, tend to focus around the whiteboard, while if documents are under review they claim the major share of the team's attention. Those meetings (2, 3 and 4) where both whiteboard and documents are exploited are of particular interest, particularly in relation to requirements on tools for distributed design. In meeting 3, where the greatest number of instances of dual focus occur, they account for almost 47% of the total. In such cases, attention switches rapidly between the dynamic design representations on the whiteboard and (usually) more static representations in documents. Such switches are common in the protocols, and are further illustrated by the sequence of objects of gestures recorded for the sections of protocol coded as dual focus.

Workspace use in the Delft workshop data

Some indication of the validity of these observations for other domains is provided by a comparison may be made with the pattern of workspace use in the data for the Delft workshop (Cross, Christiaans and Dorst, 1996). This was derived from a product design session undertaken by a group of three experienced designers. The designers had to devise a device to carry a backpack on a mountain bike, starting from a design brief and producing a costed concept design with supporting sketches, within a two-hour time limit. A videotape recording and protocol transcript were made available for analysis to the design researchers at the Delft workshop and to myself.

The group had several shared workspaces at their disposal. These were:

- documents containing the design brief;
- large sheets of drawing paper, placed on the table between the three designers, and sometimes also used for individual work;
- a whiteboard, of the sort from which copies may be made;
- a sample backpack and mountain bicycle (the task was to design a carrier for a backpack on mountain bicycle).

An analysis of focus patterns is provided by Radcliffe (1996), who observes how group and individual work moves easily between different workspaces over the two hours of the design task, and that workspaces are often used in parallel by group members. Early work focuses mainly on the documents, then attention moves to the whiteboard and artefacts, reverting to documents and sketches on the shared paper as the design is finalised. Closer inspection also shows that group members sometimes use corners of the shared drawing paper or the whiteboard for independent work in a similar fashion to the use of private notepads by the Mallard team. The video data also shows that the artefacts provided are sometimes used in tandem with the whiteboard to explore ideas physically as they are developed in sketches or text.

For both groups, then, we have an overall picture of people moving between workspaces in a fluid and flexible way, sometimes focusing entirely on one workspace, sometimes working in parallel using different resources, and sometimes using one medium to supplement the affordances of another.

6.3.2 Analysing gestures as evidence of workspace use

For the purposes of this work, following Tang (1989, 1991) a gesture is any apparently purposive physical action excluding writing and drawing and bearing an apparent relation to the task in hand. A set of gesture categories was derived from repeated viewing of the videotape. These were:

Category	Description
point at whiteboard	Pointing at a specific item on the whiteboard. Cases where the 'point' is prolonged, e.g. in following a data flow, are coded as one instance. 'Points' only occur when the person concerned is in close proximity to the whiteboard.
point at document	Pointing at a specific item on paper notes or printed documents. Cases where the 'point' is prolonged, e.g. in following a data flow, are coded as one instance. 'Points' only occur when the person concerned is within physical pointing distance of the document.
gesture at whiteboard	Any gesture in the general direction of the whiteboard. Examples of 'gestures at' include waves of the hand, pointing movements, and occasionally physical movement towards, the whiteboard.
gesture at document	Any gesture in the general direction of paper notes or printed documents. Examples of 'gestures at' include waves of the hand and pointing movements.
pick up document	As its name suggests, can refer to notes or printed documents. Does not include any observable consultation of the document concerned.
standalone gesture	Any gesture not apparently directed at any of the workspaces. Examples include holding the hand apart to indicate size, moving the hands over and over each other to illustrate a program loop.

As will be evident from the above, the initial classification was at a behavioural level. As for the analysis of the group's focus, I carried out gesture classification as a single observer, and re-viewed the tapes to check consistency. Other systems of gesture categorisation previously used in design research were considered. Tang's (1991) study categorises the purpose of actions in design meetings (which include writing and drawing) into storing information[32], expressing ideas and mediating interaction. While a useful high level classification, this proved to be insufficiently detailed for the particular purpose of investigating the role of gesture in relation to workspace use. Using Bekker's (1995) categories of kinetic, spatial, point and other gestures would have had the

[32]Note that Tang uses this category to include not just writing or drawing, but the use of an illustrative gesture to 'fix' an idea - for example the way a user interface button operates.

advantage of facilitating direct cross-study comparison, but would not directly identify the referent of the gesture[33].

The semantics of gesture in this context are considered further below, but at this point it may be helpful to add a mapping to the psycho-linguistic taxonomies of gesture, as exemplified by McNeil (1992). In McNeil's terms, all the first six categories above are different forms of *deictic*, while the 'standalone gestures' comprise both *iconics* and *metaphorics*. Excluded from the classification are gestures which appeared (from a review of the protocol and videorecording) solely to orchestrate dialogue. These include, for example, rhythmic movements of the pen accompanying the flow of a sentence, nods of the head, turning towards another person - MacNeil's *beats* and *cohesives*. Non-verbal communication of this sort is a very richly researched field; among many other authors are Ekman and Friesen (1969); Argyle and Cook (1976) (as well as McNeil (*ibid*) and, more recently, focussing on the design of communications technology, Fussell and Benimoff (1995). However, the emphasis here is on the interaction of workspace and gesture as specifically related to distributed design representation, so orchestrating gestures are not further considered.

6.3.3 Evidence for focus shifts from the gesture record

These gesture categories were applied to the video record and linked to the relevant atoms of the verbal protocol record. Protocol fragment 3 and Protocol fragment 4 show instances of switching focus and their accompanying gestures.

Time	Who	Action	Utterance
1440	P	gestures at document looks at whiteboard gestures at document	Right. All I'm thinking about here is in some way you've got to define a given set of events, and when you've got those events, what the animation is. I mean, OK you might define it in terms of these frames [R: scenes] things - whatever, who cares. [R: laugh] Somehow you've got to handle that and you've got a set of rules to work out what you're going to do. [R: mm hm] The more composite the events get the more complex that set of rules is and the more you're going to have. [R: mm hm] So there's a balanced trade-off between the granularity of the event structure if you like and the complexity of defining animation rules.

Protocol fragment 3. A sequence of focus shifts between document and whiteboard, showing use of gesture

33 Moreover, this work came to notice after analysis had been completed.

Time	Who	Action	Utterance
13.56	R	gestures at whiteboard gestures at whiteboard	OK. I thought I'd pretty-print that [laughter]. Probably haven't lost too much that was interesting. Construction on the left hand side but I've left a little bit of space around construction because we may well end up breaking it down [P: mm hm] into models of state, configuration type things, [P: yeah] [M: yeah]. Control, well yes, control, scheduling's permeating all layers as a result of some of the things we've said [P: mm hm] user model, value, evaluation of value, only goes up as far as the abstract model, I haven't taken that through [P: yeah] and then there's execution events whose flavour changes as we go up [P: yeah] and everything else whose flavour changes, we really ought to...
13.57	R	looks through document points at document points at whiteboard looks through document	So what I propose we do is partly on the basis of this original breakdown in structuring that we should be able to find a home or an equivalence for functional things in these mapping layers [P: mm hm] like parser, concrete to abstract controller - we had one of those but we haven't got a name for it now [P: mm hm] and so we ought to be able to put in some things, no real attempt to make them object oriented diagrams, just identify what lumps are in the jam [P: yeah] strawberries or whatever it is, so...
13.58	R	drawing on whiteboard	Well, let's do an easy one. We're actually saying that the user model values... they'll just go straight through, yeah. Scheduling... shall we do scheduling? [P: yeah] Any bids for anything else?
13.58	R	drawing on whiteboard points at whiteboard drawing on whiteboard	So what we've got down here is a set of Petri net enablings [P:ah ha] so it's in terms of transition IDs and place IDs and token values, all in this domain we're talking about which is in VSML. [P: yeah] So I assert we've got two things going on here there's VDM user model values [M: mm hm] in that case we need token values [P: yeah] and there's basically transition and place Ids being mapped to... so can we just say kernel model to user model elements?
13.59	M	R drawing on whiteboard	That's a structure translator is it?
13.59	R	points at document	Yeah structure... but it's using the same information whereas... we'll look at the words later. I think this may be more precise or a subset of it.

Protocol fragment 4. A further sequence of focus shifts and use of gesture to clarify reference

Analysis of the referent of gestures also provides a cross-check on the relative amount of time spent considering design representations in the different workspaces. Of the workspace related gestures, i.e. all categories except standalone gestures:

- 52.4% overall were whiteboard related;
- 47.6% overall were document related.

Again, it can be seen that the group's attention was roughly equally divided between these two locations.

Other aspects of gesture are examined in more detail in section 6.4.

6.3.4 Analysing design process

A complementary perspective on workspace use is provided by linking the focus as indicated by the gesture record to design process. Design process analysis is a well-established technique employed by a number of researchers, including Guindon (1990), Olson et al (1992), Dwarakanath and Blessing (1996) and Bekker (1995). Data is obtained by tape-recording subjects 'thinking aloud' as they perform a design task (in the case of single designers) or the natural stream of discourse in the group situation. It is well-recognised that verbal protocols from individuals have significant limitations because of the difficulties of introspection and the interference of this extra task on the primary task, in this case, design. (Cross, Christiaans and Dorst, 1996b, and Lloyd, Lawson and Scott, 1996, comment on the issues in the context of design process analysis.) Group protocols avoid these particular problems, but the evidence they provide is somewhat indirect: it is relatively uncommon to find an explicit comment on the design process itself, except when the organisation of the meeting is under discussion. Rather the nature of the activity in hand has to be inferred by the researcher from the surrounding dialogue, both verbal and non-verbal, and sometimes actions on workplace artefacts. The text of the verbal protocol is divided into discrete utterances or chunks, where a chunk may be anything from a single word to a prolonged discourse. A new chunk begins when another speaker takes over, after a significant pause, or when the category of the discourse changes. The grain of analysis is therefore molecular rather than atomic, by contrast with the finer grain of conversation analysis. Most authors use categories derived from design theory and the work on design rationale for example, Conklin and Burgess-Yakemovic (1991); Lee and Lai (1991); Buckingham-Shum and Hammond (1994); Moran and Carroll (1996). Core design categories usually include:

identifying issues or problems;

proposing solutions to these problems;

identifying criteria for deciding between the solutions;

deciding upon a solution.

In addition some authors add categories for ancillary activities of the following general types:

summarising progress;

walking through solutions.

So far these categories can apply to both individual design work and work undertaken as a group project. In order to include activities specifically related to work in groups, three extra categories are commonly used (and most authors also include 'digression' and a very necessary 'other' category):

seeking and providing clarification;

managing the meeting process;

managing the wider design project.

In the current work, an analysis of the design process is described in relation to:

the exploitation of workspaces (this section);

the functions of gesture (section 6.4)

the nature of the group's work over time (section 6.5);

and to facilitate a comparison with group studies reported in the existing literature (section 6.5). The design process categories used are directly from Olson *et al.* (1992a), thus allowing comparisons to be made between the current results and those reported in that study and by Bekker (1995), who also use the Olson categories. Moreover, the Olson categories had been found by them to be robust between raters and across meetings. The scheme shown below, derived from Olson *et al.* and extended with examples from the current work, explains the categories.

Category	Application as condensed from Olson *et al.* (1992a) and *example* from Mallard.
Issue	The major questions that need to be addressed. They typically focus on the major topics of "Shall we offer this capability?" and "How can we implement that?" *There seems to be something that's missing in that what really isn't clear in the document is the interfaces between different people's bits or whatever. I'm not clear how that's coming out at the moment.*
Alternative	Solutions or proposals about aspects of the designed object, typically either features to offer the user or ways to implement the features. This includes the elaboration of the idea. *It would essentially be creating its own animation event group for that, in some sense cutting out the ones that aren't appropriate, and then it's got to find out how to map that to a display event group.*
Criterion	The reasons, arguments, or opinions that evaluate an alternative or a general evaluative principles. Occasionally these appear in the form of analogous systems. Author's note: Criteria are difficult to disentangle from alternatives. A description of an idea often contains an explicit criterion. *I think the arguments for it are as follows. That in the domain of the abstract model there's a set of primitive changes that you can make to that model like add a process and a flow. But in fact those aren't primitive enough. There'd be things like add the first flow out of a process and the subsequent flow out of a process and the first flow into the process. It's that sort of thing, mm yeah.*
Project Management	Statements in which people are assigned to perform certain activities, decide when to meet again, report on the activity (free of design content), and so on. *That's an issue for the implementation task. We've got to work out how workpackages 2 and 3 work together in this area.*
Meeting Management	Statements having to do with orchestrating the meeting time's activity, indicating that group members are to brainstorm, hold off on discussions and so on. *What I propose we do is look at the concrete model first and that's what we put on the board when we did the concrete model – all this is up-to-date.*
Summary	Reviews of the state of the design to date, restating issues, alternatives and criteria. It is a summary if it is a simple restatement. If it is ordered by steps it is a walkthrough. *Construction on the left hand side but I've left a little bit of space around construction because we may well end up breaking it down into models of state, configuration type things. Control, well yes, control. Scheduling's permeating all layers as a result of some of the things we've said... user model, value, evaluation of value, only goes up as far as the abstract model, I haven't taken that through and then there's execution events whose flavour changes as we go up and everything else whose flavour changes...*
Clarification	Questions and explanations including repetitions for clarification and associations. Author's note: Clarifications and substantive statements are difficult to discriminate, since clarifying explanations often extend the description of an alternative, for example. Clarifications were recorded but analysed with their parent category unless they did not relate to any of these. *Yeah, I wanted to know what the relation was. You have the symbol that's related to a relation, I wasn't sure what it was.*
Digression	Joking, side topics (e.g. how to get the computer to make dotted lines), or interruptions. Author's note: Interruptions were not coded, but deducted from the meeting time. Digressions are recorded but are not included in the analysis. There are few instances. *I think I'm drawing radishes.* *No, they're definitely cherries – look at the stalks!*
Goal	Statement of the purpose of the meeting and some of the constraints to work under, such as time to finish or motivating statements about how important this is. *OK. What I want to achieve is a position where we can keep going forward down this path, assuming that we still think it's sensible – and if it's not, we need to discuss it – and the way forward is to refine these diagrams and put them together in a document and issue it for comment by other people.*
Walkthrough	A gathering of the design so far or the sequence of steps the user will engage in when using the design, used to either review or clarify a situation. It usually follows the user's task or the flow of data or messages inside a system architecture. *... you've got a transformation, let's have it with an external, this is a signal flowing to it on the view that corresponds to the decomposition. Well, let's actually have this as a... Probably we've got a token there and we've now got a single token here. What we're about to show is the consumption of that signal to cause a state transition. Now it depends what you want to term an event, but a set of events would be to remove that token to cause some movement along here and put a token there.*
Other	Time not categorisable in any of the previous categories. Author's note: 'Other' utterances are very rare, and are almost all verbalisations that cannot be deciphered. 'Other' utterances are recorded but are not included in the analysis. *Let's see if X [another partner in the project] take it up. I suspect they won't 'cos they're not into model partitioning at all, it's Y [another project partner] who are the driving force.*

6.3.5 Design process, design workspace and ownership

Figure 6.5 below shows the percentage of document and whiteboard related gestures for the design process categories for all meetings. It can be seen that when discussing alternatives, criteria and clarifying issues, attention is broadly equally divided between documents and the whiteboard. Reference to paper resources held by individuals dominates the raising of new issues, the summarising of discussions, the management of the meeting and, to a lesser degree, the wider management of the project. (Only one gesture was recorded as relating to goal setting, this was pointing to a document.) The only elements of the design process to use the shared whiteboard as their main resource are walkthroughs.

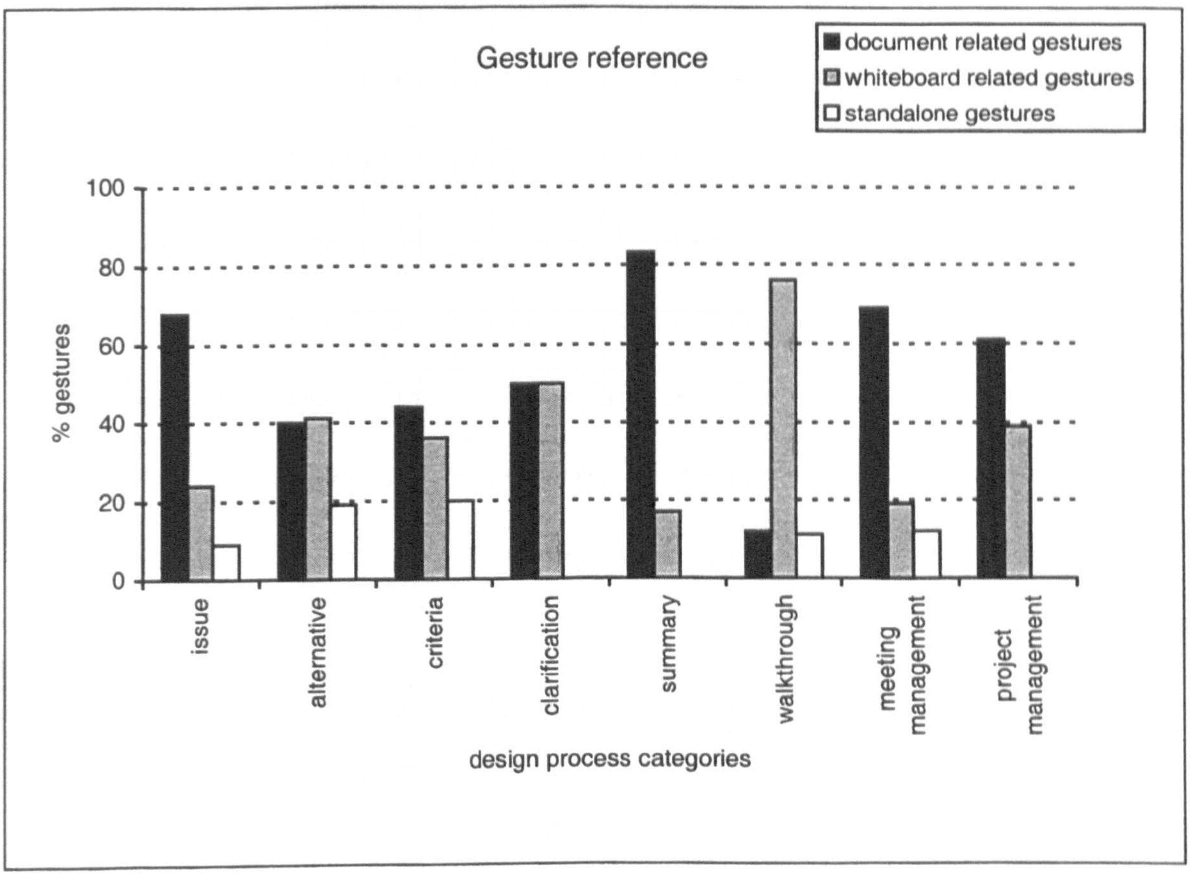


Figure 6.5 Gesture reference and design process

It might be expected that an individual's notes and documents are private to that individual, whereas public workspaces are confined to explicitly shared areas - the whiteboard in this series of meetings, but other instances would be flipcharts, large sheets of drawing paper and so on. And indeed the ownership of the different workspaces appears to follow this pattern for most of the time in Mallard meetings. However, on a number of occasions, the boundaries between public and private are much more fluid. These were particularly common in meeting 2, and in the 173 minutes of the meeting the following incidents occur:

pointing at material on another person's document or notes	(15 instances);
glancing at another person's document or notes	(4 instances);
writing or drawing on another person's document or notes	(2 instances);
picking up another person's document or notes	(1 instance);
writing on own notes and showing the result to another person	(1 instance).

For all meetings, the distribution of boundary shifts is:

pointing at material on another person's document or notes	(25 instances);
glancing at another person's document or notes	(5 instances);
writing or drawing on another person's document or notes	(2 instances);
pointing at own document to show another person	(1 instance);
picking up another person's document or notes	(1 instance);
writing on own notes and showing the result to another person	(1 instance);
converting own document into a group resource	(1 instance).

Private space is thus occasionally shared, predominantly by pointing at someone else's material, and interestingly the sharing is generally not initiated by the owner of the workspace. Sharing is only the subject of comment in the one case where a document is becomes a group resource. Here R remarks "I've got... [pause] ...this is a group exercise", before moving the document to the centre of the table. This suggests an explicit, perhaps semi-permanent, change of status from private to public. In all other instances permission to enter private space appears to be implicit and ephemeral. Protocol fragment 5 shows a typical example.

Time	Who	Action	Utterance
14.48	P	points at R's notes R writes	I think the thing we have to say now is that it has to somewhere, but we're not sure where.
14.48	M	points at R's notes	So this mapping layer we're discussing now, that would build in some knowledge after that list of events occurs and as you say that corresponds to an event that's happened in the kernel.

Protocol fragment 5. The sharing of private workspace

Awareness of what is happening in other people's workspace has a further function more related to meeting process than content: for several periods in different meetings Matt is entirely silent - inspection of the video record shows that he has not withdrawn participation, but is following the discussion, making notes of others' contributions and the contents of the whiteboard. Roger and Peter are of course aware that both this is the case and therefore that Matt is in his turn equally aware of the progress of the design – an example of intersubjectivity, in distributed cognition terms.

6.3.6 Navigation in the workspace - the role of gesture

As suggested above, gesture has a role to play in navigating between and within the various design workspaces and clarifying instances of deixis. Analysis of the Mallard video records and the accompanying protocol suggests the main functions of gesture for navigation are:

(i) to distinguish different representations of the design as, for instance, embodied by a diagram on the whiteboard and an earlier version in a document (broad focus indication);

(ii) to identify a particular piece of design information within, for instance, a diagram (narrow focus indication).

The first gesture and glance in Protocol fragment 4 are examples of function (i), while at time 13.57 examples of both can be seen. Broad focus indicators in the categorisation scheme described at section 6.3.2 are gesturing at whiteboard or documents, pointing gestures are treated as narrow focus indicators.

Focus indicators of either sort accounted for 84.44% of all recorded gestures. 17.05% of all gestures were broad focus indicators and 67.39% narrow focus indicators. The relative proportions of broad and narrow focus indication, broken down according to the design activity embodied in the relevant protocol segment, are shown in Figure 6.6 below. Again, the 'goal' category is omitted, since only one gesture was recorded for this.

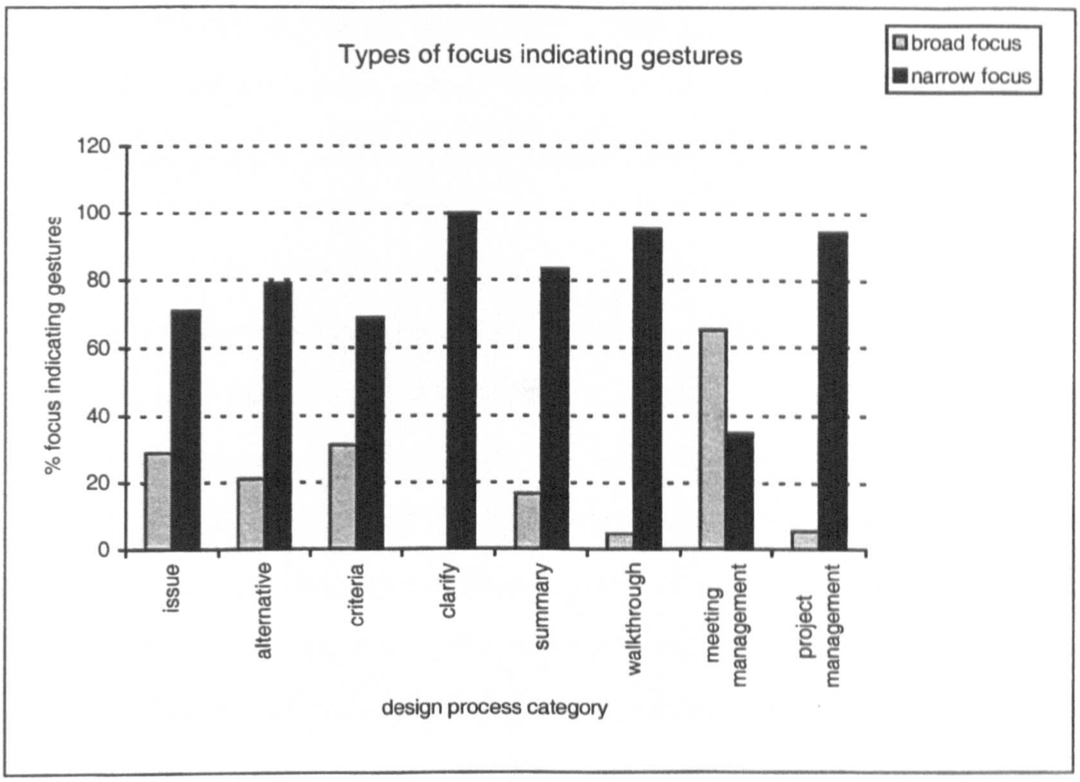


Figure 6.6 Focus indicating gestures in each design process category

From this it can be seen that:

- narrow focus indicators predominate, except for meeting management;
- for the three core design categories of raising issues, proposing design alternatives and decision criteria, broad focus indicators range from a little over 20% to a little over 30% - a not insignificant proportion.

Indications of broad focus appear to be necessary because of the distributed nature of design representations and the shifts in attention between them. Examination of the verbal protocols shows that verbal deictic reference is very rarely unambiguous. In the 128 minutes of meeting 3, for example, there are only eleven explicitly referenced verbal deictics, i.e. phrases of the type 'the top bit of black' as contrasted to 'over there'. For comparison purposes the gesture types for the first 30 minutes (about 25% of the time) of the Delft product design session were analysed. Here the pattern is rather different: focus indicators make up a somewhat lower proportion (70.50%) of all gestures, with 34.43% of the total being broad focus and 36.07% narrow. Broad focus indicators thus play a much more prominent role, perhaps because of the way design work is distributed not only among resources for writing and sketching, but also exploits the physical artefacts of bike and backpack (discussed in section 6.3.1).

Bekker (1995) also reports that gesture is used to direct attention to a piece of information. It is impossible without a comparison condition to state categorically that this type of gesture is non-redundant, and the reported evidence from recent work on deictic gesture is ambiguous on this point. The study by Barnard, May and Salber (1996), where subjects communicated using different video views of their colleagues, strongly supports effectiveness of gesture in clarifying deictic ambiguity. Further evidence is provided by Langton, O'Malley and Bruce (1996), who report an experimental study which concludes that information from pointing gestures and speech is integrated prior to response selection. It appears that the least ambiguous source contributes more to the choice of response. At first sight this is contradicted by the work of Krauss, Morel-Samuels and Colasante (1991) and Krauss *et al.* (1995) who found no added semantic value conveyed by gesture in the description of pictures and abstract images respectively, but the 1995 report does concede that deictic gestures are "communicative both in intent and function" (p.549).

6.3.7 Summary - using design workspace

From the workspace data, it may be observed that:

- design work in meetings is an instance of distributed cognition - design representations are distributed among persistent workspaces (whiteboards, printed documents and personal notes), cognitive resources and the ephemeral media of speech and gesture[34];
- workspaces are commonly used in parallel;
- attention switches between workspaces rapidly and fluidly;

[34] The public workspace in the Mallard case were naturally constrained by the resources available to them and everyday experience suggests that a wider range of shared workspaces are often used in parallel: the product designers, for example made use of a flipchart and large sheets of paper as well as a whiteboard. Any such expansion of the potential for distributed representation will inevitably increase the complexity of navigation among them.

- different elements of the design process make different use of workspaces - but printed documents and personal notes are heavily used in all categories except walkthroughs;
- workspaces may be public or private, but sometimes private workspaces are shared;
- group interaction is supported by an awareness of what is happening in private workspace;
- deictic gesture is prominent in navigating among workspaces.

6.4 Gesture further considered

6.4.1 Standalone gestures

Standalone gestures, i.e. those that did not appear to be oriented to design representations in any of the physical workspaces, corresponding to McNeill's (1992) *iconics* and *metaphorics*, accounted for 15.6% of the gestures recorded for the Mallard team. A typical example is the use of the hands to suggest the flow of control through a Petri net. Such gestures, of course, are part of the distributed representation of the design at the point of their appearance. Figure 6.7 below shows the number of standalone gestures per minute for each design process category compared with the overall rate of gesturing. The rate per minute has been used to facilitate comparison with other studies. It may be seen that standalone gestures are relatively rare, but are made slightly more frequently when the team is carrying out a walkthrough.

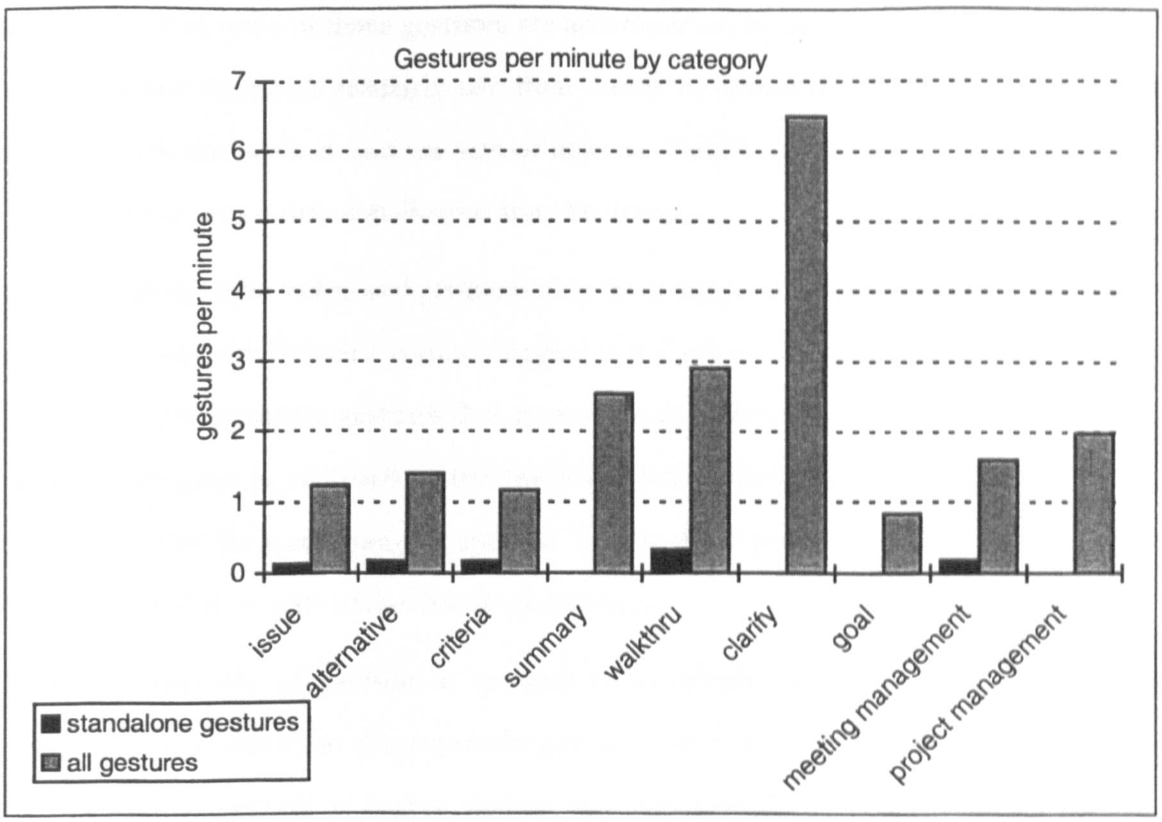


Figure 6.7 Standalone gesture shown as rate per minute, by design process activity

The mean rate for all activities is 0.12 per minute, or 0.002 per second. As can be seen in the table below, this rate of standalone gesture is much lower than that found from the author's own analysis of the product designers' data and Bekker's (1995) study of user interface designers (using a sum of the closest equivalent categories, kinetic and spatial gestures). By contrast, Tang (1989, p. 79) observes that all but one of the gestures recorded in a total of 1 hour 55 minutes of user interface design sessions were made in relation to sketches on the paper.

Design group	Standalone gestures per minute	Standalone gestures per second
User interface designers (Bekker, 1995)	0.90 to 3.60	0.02 to 0.06
Product designers (this study)	0.77	0.01
Software designers (this study)	0.12	0.002
User interface designers (Tang, 1989)	0.01	0.0001

These differences may be a matter of individual communication style - without directly comparable data from more groups it is difficult to know - or in the interpretation and coding schemes used in the different studies. They may also be an effect of the relative degree of abstraction of the domains concerned. The product designers were designing a mountain bike accessory to carry a backpack. Their gestures indicate such things as how a fastening might work, how the backpack would be taken off its carrier, how the centre of gravity would shift on the bike with the backpack attached. (Interestingly, Radcliffe, 1996, notes that some of these gestures are accompanied by appropriate *sounds*.) Bekker's user interface designers similarly use their hands to demonstrate the way users would interact with the product and the size of objects. Tang's results are however anomalous in this respect, so no firm conclusions may be drawn.

Consideration of the video and protocol data from the product designers and the Mallard team, together with Bekker's results, suggest that there are two major types of standalone gesture. These comprise gestures that appear to add information where a concept has no verbal expression at all (such as the fastening action mentioned above) and those that simply parallel the accompanying speech. The former type seems to be a representation, albeit ephemeral, of part of the distributed design.

However, the role of gesture in general in clarifying meaning (rather than deictic reference) is contested in the psycholinguistics literature. The traditional position has been that that gesture is highly salient to communication - the review in Langton, O'Malley and Bruce (1996) cites, among others, Graham and Argyle (1975); Riseborough (1981); McNeil (1992). However, other, very carefully controlled studies suggest little or no added value (Krauss, Morel-Samuels and Colasante, 1991; Krauss et al, 1995). Krauss, Morel-Samuels and Colasante also suggest that gesture may act as an aid to concept formation and its expression in speech. If this is the case, then the low rate of standalone gesture in the current case may be related to the fact that for much of the meeting time, the Mallard designers were not developing ideas *ab initio* but presenting and explaining them to colleagues. At first sight, the position that iconic and metaphoric gesture may be redundant is supported by the body of CSCW and groupware studies (discussed in section 3.1) which have found no added value for video, but this may be explained by the fact that most video tools studied restricted the field of view to head and shoulders only.

In summary then, while the current work fails to find evidence for the role of iconic and metaphoric gesture in conveying meaning, rather than reference, in the software design context, gesture does seem to be much more prevalent in more concrete domains. How far such gesture is redundant cannot firmly be established without controlled experimental work in the design context.

6.4.2 Gesture and design process

Design process categories may be divided into two main groups: design activities (issues, alternatives, criteria and walkthrough) and management activities, (goals, summary, meeting management and project management). Calculating the mean *overall* gesture rate for the Mallard team for these two super-categories gives a rate of 1.68 gestures per minute for design and 1.75 per minute for management. Gesturing then is equally prevalent in both types of activity. This finding is confirmed by a paired two-sample t-test. As for individual design process activities (see Figure 6.7 above), there is no variation in gesturing rate worthy of comment. The relatively high rate for residual clarification may be disregarded, as this comprises only 1.2% of meeting activity. By contrast, Bekker reports significantly more gesturing for design than management, using the same grouping of categories. The explanation for the difference between the two sets of findings may lie with the domain characteristics discussed above.

6.4.3 Summary - the role of gesture in design meaning and process

It has already been established that gesture has an important role in clarifying focus and deictic reference. From further analysis, it can be observed that:

- there is some evidence that gesture <u>may</u> help to amplify verbal explanations, but this can be argued more convincingly in less abstract design domains;
- for this more abstract domain of software design, the rate of gesturing does not differ significantly between design and management activities, or with the type of design activity being undertaken.

6.5 Design group process

6.5.1 Design process across meetings and groups

The primary exploitation of design process analysis in this study is in support of the analyses of workspace use and gesture. However, process analysis also provides an indication of the consistency of group behaviour across meetings and with other groups. Figure 6.8 shows the proportion of time spent on each activity across the five meetings, disregarding 'digression' and 'other'. It can be seen that the profile across meetings is broadly consistent, although a smaller proportion of time was spent on design alternatives in meeting 4, when the main subject was the structure of the design document, and the time devoted to walkthroughs decreases over the series. Roger remarks on the iterative nature of this process during meeting 1, observing "I'm not sure where we'll follow this next. I'm not sure how much we've learnt about structure [of the specification animator] but we've learnt a bit more about requirements.".

The Pearson correlations between the content of individual meetings are high, ranging from 0.84 (between meeting 1 and meeting 4) to 0.99 (between meeting 2 and meeting 3). A two-way analysis of variance confirms the overall consistency in meeting content. Much more time is spent on the exposition of design alternatives than on any other activity, even in meeting 5, when the design concepts are being finalised, but it should not be inferred from this that no progress has been made by the end of the last meeting. An examination of the overall purpose of each meeting and of the contents of the verbal protocol indicates that the different sessions developed different areas of the design and at different levels of detail.

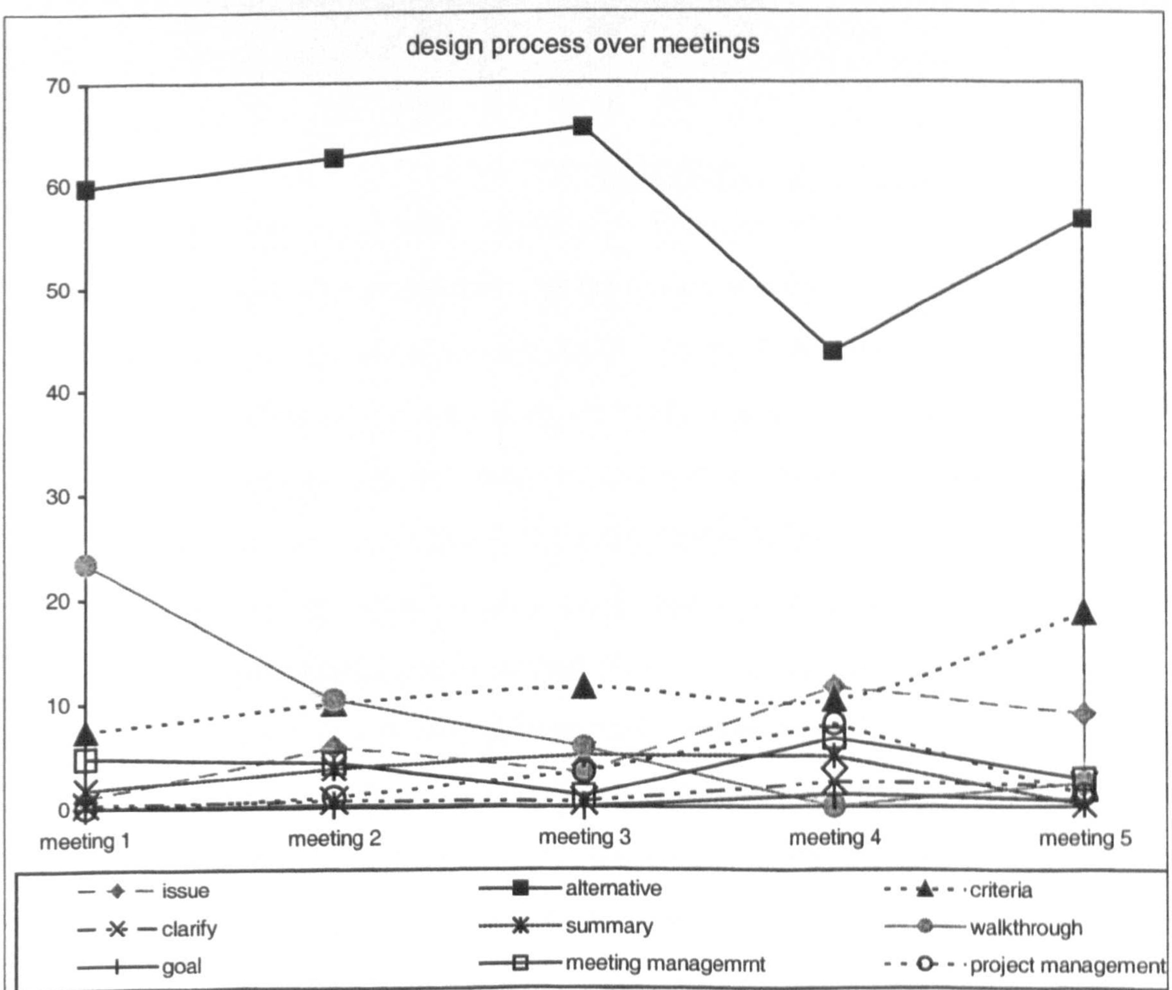


Figure 6.8 The pattern of activity across meetings

The correspondence between the pattern of activity displayed by the Mallard team and the teams of software designers studied by Olson et al (1993) is only moderate. Across all categories the Pearson correlation between the category means obtained for Mallard and the Olson means is 0.63, disregarding 'digressions' and 'other', 0.65. (The corresponding inter-observer correlations obtained by Olson et al ranged between 0.68 and 0.99.) The full comparison is shown in the table below. Both sets of results only include residual, unclassifiable clarifications in the 'clarify' category.

	issue	alternative	criteria	clarify	summary	walkthru'	goal	meeting manage-	project manage-	digression	other
minimum	0.9	43.7	7.4	0.4	0.1	0.0	0.0	1.0	0.0	0.8	0.7
maximum	11.4	65.6	18.4	2.4	4.9	23.4	1.1	4.8	7.9	11.3	1.6
mean	6.0	57.6	11.4	1.2	3.1	8.3	0.4	3.8	2.7	4.6	0.9
Olson *et al* (1992) minimum	0.3	7.7	3.8	0.0	3.8	0.0	0.0	0.0	0.0	0.0	0.0
Olson *et al* (1992) maximum	17.2	44.6	27.6	30.8	50.0	13.6	6.7	20.8	46.4	11.9	3.5
Olson *et al* (1992) mean	6.4	21.5	15.0	8.0	18.5	3.6	1.3	6.6	13.1	5.2	0.7

Figure 6.9. Cross-study comparisons for design process

The most likely reason for these results is a difference in coding practice. Another explanation suggested by consideration of the current material is that of the design style adopted. The Mallard team proceeded depth-first by developing a candidate solution until problems were met, rather than explicitly applying criteria to each solution proposed, or considering a wider range of issues and solutions at the outset. This would appear a natural outcome of what Potts and Catledge (1996) have termed the 'Inquiry Cycle', whereby design ideas are first developed in activity outside meetings, then presented and discussed and finally refined after such discussion. Protocol fragment 2 in section 6.2 illustrates this in action. This could account for the comparatively large amount of time spent on alternatives at the expense of criteria, a ratio of just over 5 to 1, as compared to the Olsons' 1.4 to 1. The Olsons also note a high degree of micro-level structure, observing regular 'sequences of issues raised, followed by suggested alternatives and the application of criteria. A rigorous analysis of the current data in this way has not been applied, but an examination of the protocols indicates a much less regular pattern. However, without the relevant comparison data from the study, whether the cause of the discrepancy is an artefact of coding practice, a real difference in design style or some other factor remains uncertain.

In contrast to Mallard, examination of the product designers' protocol shows that this group set out to take an overall approach of decomposing the problem, generating a number of design alternatives, with the stated intention of evaluating each one. Explicit statements to this effect can be found in the protocols, as shown in Protocol fragment 6.

Time	Who	Utterance
00:12	J	OK so we have an hour 00:12:00 and fortyfive so we need to generate some concepts and I guess refine the concepts and or whaddaya call
	I	information or
	K	yeah we wanna look at the em customer feedback or the users' testing
	J	oh-yeah so maybe yeah wherever that comes in this list and then uh ... em and like evaluate design ideas and decide on a concept select one

Protocol fragment 6. The product designers establish a design strategy.

This is reflected in the analysis of their design process[35] by Dwarakanath and Blessing (1996) which reports an equivalent alternative/criteria ratio of slightly less than 2 to 1. Radcliffe's (1996) analysis of the session notes the explicit attention to structure, but points out that the actual process deviates from the intention, with new concepts appearing very late in the process.

6.5.2 Summary – design process in Mallard

Design process in the Mallard software design meetings has been shown to take a depth-first solution focused approach similar to that identified by Guindon (1990) (analysing software engineers) and discussed by Singley and Carroll (1996) in their proposed taxonomy of design reasoning. The process also appears relatively unstructured in classical design theory terms. It may also be observed that:

- the structure of activity during design meetings seems likely to be related to the way the overall design task is organised, i.e. whether the meeting is discussing solution concepts developed by designers beforehand, or starting the entire process from the beginning;

---

[35] These authors use different process categories, but sufficient information is provided to make a rough comparison possible.

- the Mallard evidence suggests that design meetings over the period of a design project are remarkably self-similar in terms of design process activities;
- cross-study comparisons are hampered by the difficulty of ensuring consistent application of coding schemes.

6.6 An integrated account of small group design meetings

This chapter has discussed the issues of distributed representation and its relationship to workspaces, the role of gesture in navigating between representations and clarifying meaning and issues of design process. The major source of primary data has been a series of software design meetings: this has been supplemented by data from a product design session and discussed in the light of studies reported in the literature. As a result of this, it is possible to identify some fundamental features of cooperative work in *face-to-face* design meetings. The consideration of such features will be crucial if technologies are to support *distributed* design meetings effectively. (Note: the nature of the link between studies of existing cooperative work practice and the design of new technologies is problematic and is discussed in Chapter 7.)

The features identified are now brought together through the medium of a scenario.

6.6.1 Grounding technology design in scenarios

In recent years, scenarios have become a well-used tool in the analysis and design of human-computer systems. They have many roles, from the documentation of existing practice, through validation of understanding with users, to requirements generation and the envisionment or evaluation of future options (Carroll, 1995). Essentially, a scenario tells a story which encapsulates a particular aspect of work practice, or illustrates how the functionality of new technology might be put to use. Scenarios are particularly useful where the context to be supported comprises a relatively small set of short-term, bounded activities, or alternatively, as snapshots of tasks strung out over days or weeks. For example, the technique was used to discuss general possibilities in the stakeholder interviews at Metre, but would also have been invaluable in illustrating for the pilot users how the particular technology implemented might be exploited, had sufficient detail about the engineers' working practice been available. Here a scenario is used in two ways: firstly, to encapsulate a single composite picture of the series of software design meetings analysed; secondly, as an index to features to be taken into account in designing technology to support such a meeting held across different sites. In this way, recommendations for the design of technology remains grounded in the context of the original activities.

6.6.2 Features of design meetings and an illustrative scenario

For the sake of convenience, the design meeting features identified in earlier sections of this chapter are repeated here. This time they are numbered and indexed through those numbers to elements of the scenario. Later, in Chapter 7, the features are integrated with the results of the fieldwork at Metre, and their relationship to technology discussed.

Features relating to workspace use:

1. design work in meetings is an instance of distributed cognition - design representations are distributed among persistent workspaces (whiteboards, printed documents and personal notes), cognitive resources and the ephemeral media of speech and gesture;
2. workspaces are commonly used in parallel;
3. attention switches between workspaces rapidly and fluidly;
4. different elements of the design process make different use of workspaces - but printed documents and personal notes are heavily used in all categories except walkthroughs;
5. workspaces may be public or private, but sometimes private workspaces are shared at the initiation of the owner or others;
6. group interaction is supported by an awareness of what is happening in private workspace;
7. deictic gesture is prominent in navigating among workspaces.

Features relating to gesture in general:

8. there is some evidence that gesture may help to amplify verbal explanations, but this can be argued more convincingly in less abstract design domains;

9. for this more abstract domain of software design, the rate of gesturing does not differ significantly between design and management activities, or with the type of design activity being undertaken. *Not directly illustrated in the scenario.*

Features relating to design process:

10. design activity in the Mallard software design meetings has been shown to be depth-first and relatively unstructured in classical design process terms;

11. the structure of activity during design meetings seems likely to be related to the way the overall design task is structured, i.e. whether the meeting is discussing solution concepts developed by designers beforehand, or starting the entire process from the beginning;

12. the Mallard evidence suggests that design meetings over the period of a design project are remarkably self-similar in terms of design process activities. *Not directly illustrated in the scenario.*

There are two further features deriving from the Metre fieldwork which also appear in the scenario. They relate to the way in which any particular design project is situated in the wider affairs of the parent organisation:

13. designers often work on more than one project, and hence interleave design activities;

14. tools available to support design work are constrained by wider company policies.

The software design meetings scenario

Setting the scene

The three designers in this scenario, Roger, Matt and Peter, have worked for several weeks on this part of the project - the high level design of a graphical animator for software specifications. They also have prior co-working experience, so know each other well. Roger is the most experienced, and the team leader (1). The team has been working separately on complementary aspects of the problem (11). The immediate precursors for their work are the ideas developed in a meeting held two days ago. These have been recorded on a whiteboard and in their individual notes. However, there is a more extensive design history to draw upon, which is represented in official project documents, earlier versions of working documents, and individual notes. The overall output of this phase is to be a high-level design document for consideration by the wider project team.

At the start of the meeting

Each designer has prepared a short working paper, which is circulated (just) before the meeting (11). Everyone has skimmed the contents of the papers, but no one has read them thoroughly. Roger is already in the meeting room, looking at the whiteboard with last time's work. A second, blank whiteboard is next to the first one. Matt and Peter come into the room. All three are equipped with the new working papers, a varied selection of other project documents, and working notebooks. They sit around one end of the meeting table, in front of the whiteboard and place their working materials in front of them (1). Roger briefly outlines the aims for this meeting. Neither of the others comments.

Design in progress

Roger then explains at some length his proposals for the part of the problem he has been working on (11). He refers to text and drawings on his working paper (4), and existing sketches on the whiteboard, explaining his new ideas by sketching on the blank whiteboard, then walking through the software events (1, 4). His focus moves rapidly between these workspaces (3). Gestures, e.g. waving a hand towards a particular document, pointing at a part of a flow diagram (7), together with occasional explicit verbal references ("that red part at the top") and movement around the room (getting up to stand in front of the whiteboard) are used to direct attention. Peter raises questions and puts forward counter-suggestions fairly often. From time to time, he sketches an idea in his notebook before voicing it (2). He also moves between the various design representations, referring also to other project documents (the others have copies of some of these, but not all) (1, 3) and indicating focus in the same way as Roger. Once or twice, he refers to a document only he has brought to the meeting, moving it to the centre of the table to show the others (4, 5). Matt speaks much less than Peter, and is often quiet for five minutes or so. He can be seen, however, to be following the discussion and writing in his working notebook (2, 6). All the members of the team occasionally clarify referents by pointing at fragments of text or diagrams in other people's documents or less frequently, in their notebooks (5, 7). Most of Matt's rare interjections are for clarification, but he does sometimes become involved in more substantive discussions. In one of these, he amplifies an explanation by leaning over the table and annotating a diagram in Peter's copy of a working document (5). Roger appears to accept some of the points raised by the rest of the team and dismisses others, but few explicit decisions are voiced (10, 11). Thirty minutes or so into the meeting it is Peter's turn. The various design representations are used much as before, but Peter also accompanies some of his explanations with gestures - illustrating, for example, how the animator would 'step' through a specification (8). Roger's responses to Peter's ideas are far more frequent than when their roles were reversed. But as before, there are few voiced decisions, and ideas are introduced and explained, but rarely explicitly evaluated against criteria (10, 11). In the course of discussion new issues arise and are pursued, leaving others dangling (10). In the course of discussion it emerges that production of design diagrams is taking longer than expected because of the parent company's reluctance to invest in a particular item of special-purpose software for this relatively short-term project (14). Again, Matt speaks less often than his colleagues. After another twenty minutes Matt takes over the floor and the meeting proceeds as before.

At the end of the meeting

Without any overt conclusion, the team agree that they have done enough for one day. Roger asks Peter and Matt to investigate some of the options further (11). Roger himself will be too busy on other business on this project and some completely unrelated commercial work (13). Matt also agrees to copy down the contents of the whiteboard for future reference, and to distribute a rough version of the notes he has made during the meeting.

6.6.3 Analysing design groups - methodological considerations

The matrix in section 6.1.2 showed the different types of design task and context which have been analysed in studies of design group meetings. As will be evident, these studies populate many different cells in the matrix, making cross-comparisons difficult. Further work is needed to identify those observed features which are constant across these dimensions of task and context, and those which remain peculiar to a design meeting type. As an indication of the sort of differences that can be encountered between instances of design meetings, the 'Delft' designers, by contrast to the Mallard group were much more explicitly structured in their discussion and used fewer different workspaces (both of these aspects have already been considered in more detail). They were also more interactive - in the sense that speaker 'turns' switched more often, and they commented frequently and freely on ideas as they were voiced - and more physically mobile, getting up and down to use the whiteboard or drawing sheets, experimenting with different combinations of bicycle and backpack. To what can these differences be attributed? The list of candidate reasons could include the nature of the design task, the structure of the group, the design domain itself, individual characteristics and many others. It is impossible to be certain which are relevant, and therefore to make well-founded generalisations. Once the dimensions of cooperative design have been identified – and the current study is intended to make some contribution to this – there is certainly a good case for more controlled studies of the 'Delft' type where the design protocol and other data are made available to a number of researchers and variables can be systematically varied.

As has been argued at the beginning of this section, it is vital to supplement controlled studies with studies of cooperative design embedded in its everyday context. Again, identifying relevant sources of variation is critical, since only then can sufficient information be reported for researchers to make comparisons between their own results and those of others.

Further problems arise with the use of coding schemes for gesture, process and other design group phenomena. If researchers invent or adapt schemes so as to be sensitive to their own data, or particular theories of design (coding schemes relating to design rationale are particularly subject to this), cross-comparisons are necessarily difficult. If coding schemes are adopted 'from other studies, the problem is one of ensuring consistency of application. A few examples of each category are simply not sufficient to permit reliable re-application of the scheme. Perhaps design researchers could be persuaded to create a shared archive of coded protocols, but this may be unrealistic given the competitive pressures on academic researchers to publish and on those elsewhere to maintain commercial confidentiality.

7 Overall discussion and conclusions

This chapter

- summarises the main findings of the case study work;
- integrates these with the results of existing work into a framework for cooperative design;
- considers the potential for current and emerging technologies in this domain, using illustrative scenarios;
- indicates directions for further work.

This work has added to the small corpus of case studies of cooperative design, and in so doing has identified a number of features of such activity as practised in organisational settings, an appreciation of which, it is argued, should inform the design[36] of supporting technologies. Some of these findings are new; others confirm or extend the results of existing studies. The features identified include the tension between the traditional culture of engineering design and the underlying assumptions of new technology; the practical difficulties of sharing some types of design artefacts; the way in which the job of design entails an interweaving of individual and group activity, with the ensuing consequences for resource exploitation and navigation in group sessions. Pycock and Bowers, 1996, make a similar point about design in the fashion industry, observing that

> *It is also vital to appreciate the socially distributed and variable nature of design activities. In fact, there is no single activity which organisation members refer to as 'design' and no single organisational locus for it..*
>
> *p.226*

This leads on to perhaps the most salient feature to be observed in this study: the way in which design work is one thread in the web of organisational and group activities which an individual may undertake. The Metre and Mallard 'designers' did not just 'design', they variously submitted management reports and went to the associated meetings, negotiated the sale of their services to other groups, attended wider project co-ordination

[36] The term 'design' is used here as a shorthand for the both the design of special-purpose software and the selection and customisation of off-the-shelf applications.

sessions, claimed and authorised expenses, prepared the documentation for QA procedures, assessed the utility of software for the task in hand and many other things. And in both contexts, all this entailed co-ordination and communication not just within their respective organisations but with project partners elsewhere. The DUCK pilot at Metre, depending on one's viewpoint, either failed or was a very partial success in part because of a lack of support for the varied interactions of organisational life and the information needs entailed. If the Mallard team of software designers were to be moved to different sites, groupware tools of the type trialled at Metre, together with simple, off-the-shelf videoconferencing could apparently have supported (albeit in a very constrained fashion) the team's internal meetings and design documentation. However, they would have integrated neither well with the parent organisation's management information systems, nor with the procedures of the wider project. Nor – given the interplay of group and individual work and hence public and private meeting workspace – would such technology have supported the rich activity of design meetings. Here again, one might predict only partial success, even with this very technologically aware group. All this points to the importance of understanding not just how design is prosecuted, but how this work fits with *what else* designers do, and with *whom else* they work.

The next sections draws these observations together into a framework which forms a basis for considering the computer support of cooperative design and illustrates the practical implications in a set of scenarios.

7.1 Organising the characteristics of cooperative design into a framework.

How then can the diverse observations be organised? One readily available structure is the partitioning of information according to whether it relates to the support of cooperative design at the organisational, the group or individual level. This is a well-established and straightforward means of classifying user data relating to the introduction of new technology and accordingly forms the vertical axis of the table at the end of this section. The horizontal axis partitions material as to whether it primarily relates to the nature of the task itself, to the artefacts used and their representation, to the characteristics of the design domain and designers as a profession or to the way cooperation is organised. It takes an activity theoretic perspective, the reasons for which will now be explained.

In considering ways of structuring the information, there is currently no strong design-specific scheme to be adopted, still less one that systematically inter-relates the different facets of design which have been investigated. A generic framework which suggests itself is that of activity theory. The theory affords an integrated and systematic way of reasoning about context in human activities, and has gained considerable ground in the HCI and CSCW communities (see, for example, Engeström (1990); Blackler (1995); Heeren and Lewis (1997) and the collection in Nardi (1996). The activity theoretic view of the world is an intrinsically collaborative one. The core model, in the widely-referenced version proposed by Engeström, proposes that human activity can be understood in terms of inter-related dimensions of the objective of the activity, artefacts used in achieving this objective, relevant social norms and values, the characteristics of the individuals undertaking the activity and their community and the manner in which the tasks which make up the activity are divided between individuals. It is these dimensions which, tailored to the material in hand, form the horizontal axis of the table.

The framework structured in this way not only organises the diverse findings to be taken into account in systems to support cooperative design, but delineates the dimensions which require consideration in contextually grounded studies in this domain and facilitates the identification of areas where the current state of knowledge is particularly incomplete. This is also the first step in further work to carry forward the current results in a re-consideration of co-working in design from an activity theory perspective.

	Design task	Design artefacts & representation	Rules, norms & values for design	Designers & their domains	Organisation of collaboration
Organisation	Decisions are influenced by organisational factors beyond the problem itself (existing work).	Artefacts may be security classified or commercially sensitive, influencing procedures for their development and communication. (this study).	QA and other working procedures may mandate distributed design history (this study). Norms may conflict with CSCW - "Good design is not done on computer" (this study) Procedures often subject to the control of wider projects (this study).	Uptake of collaborative technologies for design appears to be a special, more complex case of IS uptake in general. (this study)	Designers work with others beyond the immediate design team; raising issues such as the desire to maintain visibility (this study and existing work). Designers commonly work on more than one project at a time (this study).
Group	The evidence for structure in group problem solving is equivocal (this study and pre-existing work); the degree of structure may be related to the nature of the task (this study). Meetings over the period of a project are self-similar in process; new issues continue to be raised and discussed throughout design development (this study). Different elements of the design process make differential use of workspaces. (this study)	Work in meetings is distributed cognition – representation is distributed among persistent workspaces, cognitive resources, speech and gesture (this study and existing work). Workspaces are commonly used in parallel (this study and existing work); shared representations for design may be very large scale, affording quasi-direct interaction (this study) or embodied in concrete objects (existing work) Gesture is salient in conveying meaning and in mediating interpersonal interaction, although there may be redundancy (existing work); deictic gesture is important in navigating workspaces (this study). The act of drawing itself may convey meaning (existing work). Concrete artefacts are a rich source of information and an integral part of communication (this study and existing work).	Peer culture and long-established teams appear to foster sharing of private workspaces. (this study, indirect evidence from existing work) In some circumstances, design quality attracts a lower priority than the preservation of comfortable social relations. (existing work)	In the engineering domain, computers are commonly group resources rather than an individual's desktop. (this study, supporting evidence from elsewhere) Gesture as a resource for meaning is more prominent in less abstract domains (this study)	Some collaboration is only superficial; tasks are decomposed to minimise interaction (this study and existing work). Even in peer groups, designers adopt different task-related and social roles (this study and existing work). Group interaction is supported by awareness of private workspace (this study and existing work).
Individual	For managers, interest in CSCW relates to the co-ordination of distributed teams, for more junior staff, the expressed need is for access to resources (this study) Designers work on a collage of design and non-design tasks (this study); design tasks themselves embody elements which are not prototypically 'designerly' in nature (existing work)	Representations may move from private to public space during face-to-face interaction (existing work); this may be instigated by the owner or others; such status changes are very fluid (this study)		Expectations are a reasonable predictor of uptake for collaborative technologies. (this study and existing work)	Much group activity concerns the integration of the output of individual tasks.(this study and implicit in existing work)

7.2 Matching collaborative technology to design context

The framework just presented brings together a diverse set of findings, but does not illustrate what their implications might be when selecting or designing technology to support designers working together. As observed in section 6.6.1, descriptive usage scenarios are a particularly suitable way of grounding the consideration of technology in a realistic and detailed context.

Accordingly, a set of scenarios which have been structured to reflect the findings of this work and the reported literature will be presented. They encapsulate existing practice, incorporating organisational, group and individual aspects, and taken together, illustrate almost all the features of task, artefact, culture, and the organisation of work identified above. The scenarios are then used to consider how existing technologies can support specific cooperative design activities and to note what organisational issues might arise in conjunction with the introduction of such technologies. The focus is the problem of supporting geographically dispersed groups of designers, the problem which stimulated the current work.

7.2.1 Scenario 1: distributed engineering design

Company A is a long-established engineering design consultancy. Among its functions is the design of large naval vessels. Work is usually carried out under severely constrained resources of time and finance. For this particular project, the 20-strong design team is split over two distant sites, the main company site and a distant satellite, with the manager located at the main site. From time to time, the team also draws on the expertise of domain specialists who are located at the main site. Most of the main team members work full-time on the project, but others are also engaged on other, parallel projects. This is a new team, not all of whom have worked together before. The current project starts with the perusal of a large quantity of paper tender documents, will reference the (paper) documentation from past work of a similar nature, and is required to supply the final specification and associated design proposals in paper form. The client and the company mandate strict QA, design authorisation and design traceability procedures. At present, draft designs and supporting documentation are faxed or hand-carried on paper or floppy disk between sites. PCs are fairly common, but are used only by administrative staff and management. Some of the company's sites (but not this distant site) are networked, and these are linked by email, but such links only extend to PC users. Design engineers themselves work mainly on drawing boards and design journals, using CAD software only to carry out analyses or to produce the final version of the design. Text documents are largely hand-written or dictated and then word-processed by secretarial staff. Meetings – mostly held at the main site – serve many purposes: to allocate and co-ordinate work, to brainstorm initial ideas, and to review the developing design, often presented as full-scale plots. Designers at the distant site are happy to travel for the more significant meetings, and indeed use the opportunity to catch up with colleagues or take care of business on other projects, but resent the overhead of travel for routine co-ordination activities, or simply to transport an urgently needed file. The initiative for change comes from the IS support division, who see an opportunity for demonstrating the potential for new collaboration technologies.

Potential for technological support

Proprietary applications software, the purchase of a certain amount of hardware and more comprehensive networking would deal with most of the issues of facilitating shared use of reference documentation (albeit with significant set-up overheads for data capture) and the emerging design, permitting access/version control and authorisation to whatever level of rigour required. Similarly, standard email and various forms of online discussion utilities could support the routine coordination of cooperative design tasks delegated to individuals. For most companies, such tools are likely to form part of an enterprise-wide intranet. The issues so far are thus organisational rather than technical, requiring close liaison between the IS team and project members to match functionality to working practice, foster ownership and provide training which is linked to the performance of everyday tasks. As noted, for designers in this type of culture computers are tools for particular, delimited purposes. Maintaining consistency with this 'technological frame' (to use Orlikowski's term) suggests that such applications should be available, just as other special purpose tools, on machines shared by co-located groups. This in turn implies a requirement for roving logins and portable user profiles. However, the experience of the Metre pilot indicates that such an initiative will succeed only if new tools and associated procedures are in place at the start of any new project, and if the new technologies also support the designer's work on other projects and with other groups. This latter condition implies migration on an organisation-wide basis, at least for the basic asynchronous coordination and communication tools.

As for meetings of the distributed team, it is at first sight plausible that existing desktop videoconferencing tools would minimise the requirement to meet face to face. But given that video mediated communication has been shown to be deficient in support for the formation of new relationships, that designers use meetings on one particular project to maintain a more general visibility on other sites, and the limitations of videoconferencing in affording the full richness of face to face work (of which more in scenario 2), such an initiative would be misconceived. Where videoconferencing may add value is in the occasional *ad hoc* contacts to clarify tasks, seek specific expertise on design issues and so on. Coupled with shared applications, and availability on readily accessible shared machines as outlined above, this seems a realistic possibility. Finally, and perhaps somewhat ironically, it is the further reaches of shared virtual reality that may eventually offer most potential for the support of synchronous work in such a setting. A shared, immersive 3D environment affording exploration of the full-scale design could provide substantial added value over a 2D drawing. While such systems are not yet sufficiently robust or usable to support everyday use outside the research community, their eventual availability may be the trigger which stimulates the take up of less exotic collaborative media.

One issue remains, that of security. The uptake of collaboration technologies by organisations dealing with highly classified material is entirely dependent upon the provision of data transmission at which both the organisation and its clients are convinced are of an equally high level of security. Currently, this has not been achieved.

7.2.2 Scenario 2: supporting an established software design team

Company B is a small software house with premises spread over several sites. This scenario concerns the work of a team of three engaged on the design of a new case tool.

The project has been running for some weeks, employing a pattern of spells of individual work interspersed with co-ordination and review meetings. These latter draw on original project documentation and paper design documents of varying degrees of formality generated by members of the team. (A detailed scenario of a typical design meeting of this type is to be found at section 6.6.2). Few explicit formal authorisation or sign-off procedures are mandated, although at various specified stages QA documentation is produced which must bear the team leader's signature. Otherwise, design developments or changes are implicitly authorised by the team leader during meetings. Project teams are normally co-located, but in this case one person has recently been seconded to another site to work part-time on another project. The sites are networked, providing email, shared file servers and a company intranet. All software designers are necessarily highly familiar with the PC technology and communication tools which form their everyday working environment, and all have an individual machine. While an IT support department exists, technology is very much owned and managed by the projects themselves. Here it is the absent team member who suggests that the application sharing and videoconferencing technology already owned by the company for the purposes of demonstration to clients could be put to good use in saving him tedious travel through the conurbation where the company is located.

Potential for technological support

Tools already in use by the team will help to support co-ordination when the team are working separately, with increased use of email and phone in substitution for casual face to face requests for information. For more substantial interactions between designers – the equivalent of asking a colleague to run their eye over a paragraph to see if it makes sense – application sharing may be helpful. The sharing and integration of the results of distributed individual work will be facilitated by one of the applications providing structured, shared workspaces. None of this is likely to meet with resistance, since no significant change in working practice is concerned, and members of the team are generally keen to experiment with new tools – it is a normal part of their job. It is opportunistic contact – the casual glance at a diagram over someone's shoulder which triggers a new design suggestion – which will be lost. At present, the only potential means of supporting this would be the type of 'office share' video link which has so far proved to be relatively unsuccessful in the support of technical tasks.

As for meetings themselves, the potential of current systems must be considered in the context of the rich and intricate patterns of workspace use as exemplified in section 6.6.2. Firstly, in real world meetings, the distributed nature of the design itself and of the design process entails the use of multiple workspaces. These are used in parallel and often in rapid succession, the movement between such workspaces being fluid. Consequently, this collective switching of focus must be supported in any technology-mediated meeting but without unduly increasing cognitive load or making unreasonable demands on screen real estate. At present only partial solutions exist and these tend to be neither elegant or sufficient particularly for moving between (rather than within) workspaces (see for example, Fussell and Benimoff (1995) and Smith (1996) who discuss the problems of individual desktop configuration in conferencing systems). Furthermore, for the near future some of these workspaces will be paper based, which raises the perennial problem of integrating paper and electronic media. Secondly, workspaces are by default either shared or private, but occasionally private workspaces are shared. Group interaction has also been shown to be supported by an awareness of what is happening in private workspace, so the problem is compounded by issues of ownership and (read/write/annotate) permissions. So far, technical solutions go only part way to supporting the easy and flexible management of public/private boundaries displayed by groups who habitually work together. There is also the issue of with whom to share the information: in larger meetings sub-groups spontaneously emerge and information is then shared locally, again an issue which is not elegantly served by current tools. Finally, in the absence of clearly articulated gesture, meetings will be more effortful as designers will be required to spell out explicitly change of reference both within and between workspaces. At present deictic gesture, indeed gesture of any sort, is not a medium which is supported effectively by technology, even in the form of collaborative virtual environments, although recent applications such as that described by Reynard *et al* (1998) are beginning to address the issue. Techniques which afford the close integration of directly shared workspace and body image, e.g. Ishii and Ohkubo

(1990); Scrivener, Clark and Keen (1994) appear to have worked well for design dyads, but have not supported larger numbers or multiple workspaces. As Harrison and Dourish (1996) note, in media spaces which lack a common frame of spatial reference for gesture, - "it is not your space that enters mine, but your image", an observation supported by the experimental work of Hindmarsh *et al* (1998) referenced in section 3.3 of the technology review. Thus at the current state-of-the art, meetings entailing substantive design work between remote participants are unlikely to be fully satisfactory, while routine progress reviews may be adequately supported by asynchronous means.

7.2.3 Scenario 3: developing ideas for a new consumer product

Company C is an innovative product design consultancy whose core business is the design of small consumer products. The company has been asked, at short notice, to produce two alternative conceptual designs for a mobile telephone handset for presentation to a client. The client is understood to require a detailed justification of the ideas presented. During the time frame available, the small design team tasked with the project holds a two hour meeting to generate ideas and fix on the two alternatives, but before there is time to generate presentation models of the designs, one designer has to fly to the head office in London (some 300 miles away) to complete another client presentation. Company C has invested heavily in CAD visualisation software, which most designers use to generate realistic representations of finished design concepts; indeed some designers now prefer to work right from the start in CAD, by-passing the use of paper-and-pencil sketches and physical models. All designers have ready access to state-of-the-art CAD workstations, which are networked within and between sites.

Potential for technological support

In this case, where there is a strong requirement for the documentation and retrieval of group design rationale, the advantages of design rationale tools such as gIBIS (Yakemovic and Conklin, 1990) come to the fore. While the concurrent use of such a tool in the meeting environment could be obtrusive and constraining, design rationale documentation immediately afterwards would be a possibility. As for the problem of completing the models for presentation, the exploitation of sophisticated CAD modelling, coupled with application sharing, would afford the team to work together, tweak the model and see the results in real time. In such an instance, the technology represents a real advance on what would have been possible with conventional tools and is used as a supplement to, rather than a replacement for, more established modes of practice.

7.2.4 Matching technology to practice

As will be apparent from the discussion of the three scenarios just introduced, collaborative technology is not yet so advanced as to afford a complete substitute for face-to-face working in cooperative design. Nor, however realistic the shared environments which VR technologies may provide in future, is it clear how the creation and maintenance of working relationships, activities which happen alongside more focused design tasks, can easily be supported. But there is considerably more potential for the augmentation of current practice with utilities which can streamline everyday communication and co-ordination, support more routine meetings where designers cannot conveniently be co-present and extend the capabilities of accepted single user technologies. However, as noted in section 5.4.1, the state-of-the-art in collaborative technologies is subject to rapid change. To assess the match between technology and practice at any given point it is important to understand the characteristics of the practice in question – the framework proposed earlier in this chapter is intended to help in in delineating the issues.

Finally, the relationship between technology and practice is of course dynamic. It is notoriously difficult to predict the impact of new technologies once introduced – the process which Galegher and Kraut (1992), among others, term 'adaptive structuration': individuals and groups adapt goal-directed behaviour in the context of changes in the technological environment. Indeed, such a process may be seen in action in the architectural case study discussed at some length in section 5.4.5. Given this, ample scope for further work remains.

7.3 Scope for further work

During the course of this work the characteristics of cooperative design in different contexts and domains have been further explicated, these characteristics have been gathered together in a framework according to the locus of their impact, and the contribution of current and emerging technologies reviewed in the context of three illustrative scenarios. What then remains to investigate? Several promising avenues present themselves, some of which relate to further micro-analysis of groups of designers at work, while others adopt a wider perspective.

The further micro-analysis of designers at work

(i) Further investigation of the nature of close co-working in design. At one level, this issue has infused much of the discussion in this thesis. It has been shown how, in the cases studied here, design activity was a mixture of individual and team work, and how individuals worked with others outside the immediate group. And as we have seen, projects are often organised to minimise active collaboration, with the consequence that much apparently cooperative work within the group is in reality concerned with coordinating individual effort. Indeed Kvan suggests (Kvan, 1997 and Kvan, West and Vera, 1997) that on closer inspection very little design activity can be characterised as genuine co-design. A close re-analysis of the meetings data presented here is likely to prove worthwhile, ideally augmented by other primary sources, with the specific aim of distinguishing co-design from co-ordination at a micro-level.

(ii) A cross-domain micro-analysis of structure in design meetings, relating the results to the nature of design activity in progress. Understanding the structure of group design process and how best to support it while capturing design rationale remains relatively unexplored. Differentiating different types of activity may help to disentangle this complex issue.

Wider issues in cooperative design

(iii) A review of reported case studies and other meta-reviews, with the aim of clarifying which of the generic dimensions of IS uptake are most salient for CSCW, and for design in particular.

(iv) A complementary investigation of real-world distributed, technologically supported design groups, identifying the features of interpersonal interaction and technology which make for success, and relating them to the framework derived in this study. Such a study would both further elucidate the subtleties of group-based design and produce a better understanding of technological fit in this domain.

(v) Finally, as indicated earlier in this chapter, a meta-perspective for the structuring and analysis of such investigations may be the application of an activity theory model to the design domain. The application of activity theory as a prospective tool in systems design is currently under investigation (Turner and Turner, 1998) and early work is hand to develop the analyses indicated at (ii) and (iii) using an activity theory viewpoint.

8 References

Abbott, K. and Sarin, S. (1994) Experiences with workflow management: issues for the next generation. In R. Furuta and C. Neuwirth, Eds., *Proceedings of CSCW'94*, New York, ACM Press, 113-120.

Abel, M. (1990). Experiences in an exploratory distributed organization. In J. Galegher, R. E. Kraut and C. Egido, Eds., *Intellectual Teamwork: Social Foundations of Cooperative Work*. Hillsdale, New Jersey, Lawrence Erlbaum Associates, 489-510.

Adelson, B. and Soloway, E. (1985) The role of domain expertise in software design. *IEEE Transactions on Software Engineering* **11**, 1351-1360.

Adelson, B. and Soloway, E. (1986) A model of software design. *International Journal of Intelligent Systems* **1**, 195-213.

Agarwal, R. and Prasad, J. (1998) The antecedents and consequents of user perceptions in information technology adoption. *Decision Support Systems* **22**, 15-29.

Ancona, D. G. and Caldwell, D. F. (1990). Information technology and work groups: The case of new product teams. In J. Galegher, R. E. Kraut and C. Egido, Eds., *Intellectual Teamwork: Social Foundations of Cooperative Work*. Hillsdale, New Jersey, Lawrence Erlbaum Associates, 173-190.

Antillanca, H. B. and Fuller, D. A. (1999) Refining temporal criteria to classify collaborative systems. *International Journal of Human-Computer Studies* **50**, 1-40.

Argyle, M. and Cook, M. (1976) *Gaze and mutual gaze*. London, Cambridge University Press.

Austin, L. C., Liker, J. K. and McLeod, P. L. (1990) Determinants and patterns of control over technology in a computerized meeting room. In *Proceedings of CSCW '90*, New York, ACM Press, 39-52.

Avison, D. E. and Wood-Harper, A. T. (1990) *Multiview: An Exploration in Information Systems Development*. London, Alfred Waller.

Ball, L. J., Evans, J. St.B. T. and Dennis, I. (1994) Cognitive processes in engineering design: a longitudinal study. *Ergonomics* **37** (11), 1753-1786.

Ball, L. J. and Ormerod, T. C. (1995) Structured and opportunistic processing in design: a critical discussion. *International Journal of Human-Computer Studies* **43**, 131-151.

Barnard, P., May, J. and Salber, D. (1996) Deixis and points of view in media spaces: an empirical gesture. *Behaviour and Information Technology* **15** (1), 37-50.

Bekker, M. M. (1995) *An analysis of user interface design practice: Towards support for team communication*, PhD Thesis, Technische Universiteit Delft.

Benford, S., Snowdon, D., Colebourne, A., O'Brien, J. and Rodden, T. (1997) Informing the Design of Collaborative Virtual Environments. In *Proceedings of DIS'97*, New York, ACM Press

Bentley, R., Hughes, J. A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D. and Sommerville, I. (1992) Ethnographically-informed systems design for air traffic control. In J. Turner and R. E. Kraut, Eds., *Proceedings of CSCW'92*, New York, ACM Press, 123-129.

Bentley, R., Appelt, W., Busbach, U., Hinrichs, E., Kerr, D., Sikkel, S., Trevor, J. and Woetzel, G. (1997) Basic Support for Cooperative Work on the World Wide Web. *International Journal of Human-Computer Studies* **46** (6), 827-846

Blackler, F. (1995). Activity theory, CSCW and organizations. In A. F. Monk and N. Gilbert, Eds., *Perspectives on HCI - Diverse approaches*. London, Academic Press.

Blythin, S., Hughes, J. A., Kristoffersen, S., Rodden, T. and Rouncefield, M. (1997) Recognising 'success' and 'failure': evaluating groupware in a commercial context. In S. C. Hayne and W. Prinz, Eds., *Proceedings of Group'97*, New York, ACM Press, 39-46.

Bochenek, G. M. and Ragusa, J.M. (1998) Study results: the use of virtual environments for product design. In *Proceedings of the 1998 IEEE Conference on Systems, Man and Cybernetics*, New York, IEEE, 1250-3.

Bødker, K. and Pederson, J. S. (1991). Workplace Cultures: Looking at Artifacts, Symbols and Practices. In J. Greenbaum and M. Kyng, Eds., *Design at Work*. Hillsdale NJ, Lawrence Erlbaum Associates, 121 - 136.

Boeing (1992). Enhanced factory communications. In *CSCW'92 Technical Video Program.*

Booch, G. (1991) *Object Oriented Design with Applications.* Redwood, Ca., Benjamin/Cummings.

Bowers, J. (1994) The Work to Make a Network Work: Studying CSCW in Action. In R. Furuta and C. Neuwirth, Eds., *Proceedings of CSCW'94,* New York, ACM Press, 287-298.

Bowers, J., O'Brien, J. and Pycock, J. (1996) Practically Accomplishing Immersion: Cooperation in and for Virtual Environments. In M. S. Ackerman, Ed., *Proceedings of CSCW '96,* New York, ACM Press, 380-389.

Bowers, J., Pycock, J. and O'Brien, J. (1996) Talk and Embodiment in Collaborative Virtual Environments. In M. J. Tauber, Ed., *Proceedings of CHI'96,* New York, ACM Press, 150-156.

Brave, S., Ishii, H. and Dahley, A. (1998) Tangible Interfaces for Remote Collaboration and Communication. In *Proceedings of CSCW '98,* New York, ACM Press, 169-178.

Brereton, M. F., Cannon, D. M., Mabogunje, A. and Leifer, L. J. (1996). Collaboration in Design Teams: how social interaction shapes the product. In N. Cross, H. Christiaans and K. Dorst, Eds., *Analysing Design Activity.* Chichester, Wiley.

Brunet, L. W., Morrissey, C. T. and Gorry, G. A. (1991). Oral History and Information Technology: Human Voices of Assessment. In *Journal of Organizational Computing.* 1 251-274.

Bucciarelli, L. L. (1994) *Designing Engineers.* Cambridge, Massachusetts, MIT Press.

Buckingham-Shum, S. and Hammond, N. (1994) Argumentation-based design rationale: what use at what cost? *Interanational Journal Human-Computer Studies* 40, 603-652.

Bullen, C. V. and Bennett, J. L. (1990) Learning from user experience with groupware. In *Proceedings of CSCW '90,* New York, ACM Press,

Bullen, C. V. and Bennett, J. L. (1991). Groupware in practice: An interpretation of work experience. In C. Dunlop and R. Kling, Eds., *Computerization and Controversy.* Academic Press, Cahour, B. and Pemberton, L. (1998) When people design: conversational positioning and roles in collaborative design dialogues. In F. Darses and P. Zaraté, Eds., *Proceedings of COOP'98*, Sophia Antipolis, INRIA, 35-46.

Carroll, J. M., Ed. (1995),*Scenario-Based Design: Envisioning Work and Technology in Systems Development*. New York, John Wiley.

Carstensen, P. H. and Sørensen, C. (1996) From the Social to The Systematic. *CSCW: the Journal of Collaborative Computing* **5**, 387-413.

Checkland, P. (1981) *Systems Thinking, Systems Practice*. Chichester, Wiley.

Clarke, A. A., Connolly, J. H., Garner, S. W. and Palmen, H. K. (1996). A Language of Cooperation. In J. H. Connolly and L. Pemberton, Eds., *Linguistic Concepts and Methods in CSCW*. London, Springer-Verlag, 61-77.

Conklin, E. J. and Burgess-Yakemovic, K. C. (1991) A process-oriented approach to design rationale. *Human-Computer Interaction* **6**, 357-392.

Cool, C., Fish, R. S., Kraut, R. E. and Lowery, C. (1992) Iterative Design of Video Communication Systems. In J. Turner and R. E. Kraut, Eds., *Proceedings of CSCW'92*, Washington, ACM Press, 25-32.

Cross, N. (1989) *Engineering Design Methods*. Chichester, Wiley.

Cross, N. (1997) Teamwork in Design. In M. L. Maher, J. S. Gero, and F. Sudweeks, Eds., *Formal Aspects of Collaborative CAD*, IFIP WG 5.2 Workshop, Key Centre of Design Computing, University of Sydney, Australia.

Cross, N., Christiaans, H. and Dorst, K., Eds. (1996a) *Analysing Design Activity*. Chichester, Wiley.

Cross, N., Christiaans, H. and Dorst, K. (1996b) Introduction: The Delft Protocols Workshop. In N. Cross, H. Christiaans and K. Dorst, Eds., *Analysing Design Activity*. Chichester, Wiley, 1-16.

Cross, N. and Clayburn-Cross, A. (1996). Observations of Teamwork and Social Processes in Design. In N. Cross, H. Christiaans and K. Dorst, Eds., *Analysing Design Activity*. Chichester, Wiley, 271-299.

Curtis, B., Krasner, H. and Iscoe, N. (1988) A field study of the software design process for large systems. *Communications of the ACM* **31** (11): 1268-1287.

Darke, J. (1979) The Primary Generator and the Design Process. *Design Studies* **1** (1).

Davies, S. P. (1991) Characterising the program design activity: neither strictly top-down nor globally opportunistic. *Behaviour and Information Technology* **10**, 173-190.

Davies, S. P. and Castell, A. M. (1992) Contextualizing design: narratives and rationalization in empirical studies of software design. *Design Studies* **13** (4): 379-392.

Davis, F. D. (1989) Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly* **13** (3): 319-340.

Dorst, K. (1996). The Design Problem and its Structure. In N. Cross, H. Christiaans and K. Dorst, Eds., *Analysing Design Activity*. Chichester, Wiley, 253-270.

Dourish, P. and Bly, S. (1992) Portholes: Supporting Awareness in a Distributed Work Group. In *Proceedings of CHI '92*, ACM Press, 541-547.

Dumas, C., Saugis, G., Chaillou, C., Degrande, S. and Viaud, M. L. (1998) A 3-D Interface for Cooperative Work. In *Collaborative Virtual Environments 1998*, Manchester, UK, 97-104.

Dwarakanath, S. and Blessing, L. (1996). Ingredients of the Design Process: a Comparison between Group and Individual Group. In N. Cross, H. Christiaans and K. Dorst, Eds., *Analysing Design Activity*. Chichester, Wiley, 93-116.

Eason, K. D. (1989). Patterns of usage of a flexible information system. In S. D. P. Harker and K. D. Eason, Eds., *The Application of Information Technology*. London, Taylor and Francis.

Eason, K. D. (1996) Division of labour and the design of systems for computer support for cooperative work. *Journal of Information Technology* **11**, 39 - 50.

Eason, K.D. and Harker, S.D.P. (1989) *An open systems approach to task analysis.* Internal report, HUSAT Research Centre, Loughborough University

Eason, K. D. and Olphert, W. (1996). Early Evaluation of the Organisational Implications of CSCW Systems. In P. Thomas, Ed., *CSCW Requirements and Evaluation*. Springer, 75-89.

Eastman, C. M. (1970). On The Analysis of Intuitive Design Processes. In G. T. Moore, Ed., *Emerging Methods in Environmental Design and Planning*. Cambridge, Mass., MIT Press.

Egido, C. (1990). Teleconferencing as a technology to support cooperative work: Its possibilities and limitations. In J. Galegher, R. E. Kraut and C. Egido, Eds., *Intellectual Teamwork: Social Foundations of Cooperative Work*. Hillsdale, New Jersey, Lawrence Erlbaum Associates, 351-372.

Ekman, P. and Friesen, W. V. (1969) The repertoire of non-verbal communication: Categories, origins, usage and coding. *Semiotica* **1**, 49-98.

Ellis, C. A., Gibbs, S. J. and Rein, G. L. (1991) Groupware: some issues and experiences. *Communications of the ACM* **34**: 38-58

Elrod, S., Bruce, R., Gold, R., Goldberg, D., Halasz, F., Janssen, W., Lee, D., McCall, K., Pedersen, E., Pier, K., et al. (1992) Liveboard: A large interactive display supporting group meetings, presentations and remote collaboration. In *Proceedings of CHI'92*, New York, ACM Press, 599-607.

Emery, F. E. and Trist, E. L. (1969). Socio-technical systems. In F. E. Emery, Ed., *Systems Thinking*. Harmondsworth, Penguin,

Engeström, Y. (1990) *Learning, Working and Imagining: Twelve Studies in Activity Theory*. Helsinki, Orienta-Konsultit.

Falzon, P., Darses-de-Montmolin, F. and Béguin, P. (1996) Collective Design Processes. In COOP Group, Ed., *COOP'96*, Juan-les-Pins, INRIA, 141-149.

Flor, N. V. and Hutchins, E. L. (1991) Analyzing Distributed Cognition in Software Teams: a Case Study of Team Programming During Perfective Software Maintenance. In J. Joenemann-Belliveau, T. G. Moher and S. P. Robertson, Eds., *Empirical Studies of Programmers: Fourth Workshop*, Ablex, 36-94.

Fussell, S. R. and Benimoff, N. I. (1995) Social and Cognitive Processes in Interpersonal Communication: Implications for Advanced Telecommunications Technologies. *Human Factors* **37** (2): 228-250.

Gale, S. (1991). Adding Video and Audio to an Office Environment. In J. Bowers and S. Benford, Eds., *Studies in computer-supported cooperative work: Theory, practice and design.* Amsterdam, North_Holland, 49-62.

Galegher, J. and Kraut, R. E. (1992) Computer-mediated communication and collaborative writing: Media influence and adaptation to communication constraints. In J. Turner and R. E. Kraut, Eds., *Proceedings of CSCW'92*, New York, ACM Press, 155-162.

Gay, G. and Lentini, M. (1995) Use of Communication Resources in a Networked Collaborative Design Environment, *Journal of Computer Mediated Communication* 1 (1) http://www.osu.edu/units/jcmc/IMG_JCMC/ResourceUse.html

Goel, V. and Pirolli, P. (1992) The Structure of Design Problem Spaces. *Cognitive Science* 16, 395-429.

Graham, J. A. and Argyle, M. (1975) A cross-cultural study of the communication of extra-verbal meaning by gestures. *International Journal of Psychology* 10, 57-67.

Greenhalgh, C. (1997) Analysing movement and world transitions in virtual reality tele-conferencing. In J. A. Hughes *et al.*, Eds, *Proceedings of ECSCW'97*, Dordrecht, Kluwer, 313-328.

Grinter, R. E. (1997) Doing software development: occasions for automation and formalisation. In J. A. Hughes *et al.*, Eds, *Proceedings of ECSCW'97*, Dordrecht, Kluwer, 173-187.

Gross, M. D., Do, E. Y. L., McCall, R. J., Citrin, W. V., Hamill, P., Warmack, A. and Kuczun, K. S. Collaboration and coordination in arachitectural design: approaches to computer-mediated teamwork. *Automation in Construction*, 7 (6), 465-473.

Grover, V., Fiedler, K. and Teng, J. (1997) Empirical Evidence on Swanson's Tri-Core Model of Information Systems Innovation. *Information Systems Research* 8 (3): 273-287.

Grudin, J. (1988) Why CSCW applications fail: Problems in the design and evaluation of organization interfaces. In *Proceedings of CSCW '88*, New York, ACM Press,

Grudin, J. (1994) Groupware and Social Dynamics: Eight Challenges for Developers. *Communications of the ACM* 37, 93-105.

Grudin, J. and Poltrock, S. E. (1997). Computer-Supported Cooperative Work and Groupware. In M. V. Zelkowitz, Ed., *Advances in Computing*. Academic Press, 269-320.

Guindon, R. (1990) Designing the Design Process: Exploiting Opportunistic Thoughts. *Human-Computer Interaction* 5 (2-3): 305-344.

Günther, J., Frankenberger, E. and Auer, P. (1996) Investigation of Individual and Team Design Processes. In N. Cross, H. Christiaans and K. Dorst, Eds., *Analysing Design Activity*. Chichester, Wiley, 117-132.

Gutwin, C. and Greenberg, S. (1998) Design for Individuals, Design for Groups: Tradeoffs between Power and Workspace Awareness. In *Proceedings of CSCW '98*, New York, ACM Press, 207-216.

Gutwin, C., Roseman, M. and Greenberg, S. (1996) A Usability Study of Awareness Widgets in a Shared Workspace Groupware System. In M. S. Ackerman, Ed., *Proceedings of CSCW '96*, New York, ACM Press, 258-267.

Harper, R. and Carter, K. (1994) Keeping people apart. *Computer Supported Cooperative Work* 2, 199-207.

Harrison, S. and Dourish, P. (1996) Re-Place-ing Space: the Roles of Place and Space in Collaborative Systems. In M. S. Ackerman, Ed., *Proceedings of CSCW '96*, New York, ACM Press, 67-76.

Harrison, S. and Minneman, S. (1990) The Media Space: a research project into the use of video as a design medium. In *Proceedings of the Conference on Participatory Design*, Seattle, WA, 51-58.

Harrison, S. and Minneman, S. (1996). A Bike in Hand: A Study of 3D Objects in Design. In N. Cross, H. Christiaans and K. Dorst, Eds., *Analysing Design Activity*. Chichester, Wiley, 417-436.

Harvey, C. M. and Koubek, R. J. Toward a model of distributed engineering collaboration. 35 (1-2), 173-176.

Heath, C. and Luff, P. (1992) Media space and communicative asymmetries: preliminary observations of video-mediated interaction. *Human-Computer Interaction* 7, 315-346.

Heath, C., Luff, P. and Sellen, A. (1995) Reconsidering the Virtual Workplace: Flexible Support for Collaborative Activity. In H. Marmolin, Y. Sundblad and K. Schmidt, Eds., *Proceedings of ECSCW'95*, Dordrecht, Kluwer,

Hebert, M. and Benbasat, I. (1994) Adopting Information Technology in Hospitals: The Relationships between Attitudes/Expectations and Behavior. *Hospital and Health Services Administration* 39 (3): 369-383.

Heeren, E. and Lewis, R. (1997) Selecting communication media for distributed communities. *Journal of Computer Assisted Learning* 13, 85-98.

Henderson, K. (1995). The visual culture of engineers. In S. L. Star, Ed., *The Cultures of Computing*. Oxford, Blackwell Publishers, 181-195.

Hindmarsh, J., Fraser, M., Heath, C., Benford, S. and Greenhalgh, C. Fragmented Interaction: Establishing mutual orientation in virtual environments. In *Proceedings of CSCW '98*, New York, ACM Press, 217-226.

Hix, D. and Hartson, H. R. (1993) *Developing User Interfaces*. Wiley.

Hughes, J. A., King, V., Rodden, T. and Anderson, H. (1994) Moving out from the control room: Ethnography in system design. In R. Furuta and C. Neuwirth, Eds., *Proceedings of CSCW'94*, New York, ACM Press, 13-21.

Hutchins, E. (1995) *Cognition in the Wild*. Cambridge, Ma., MIT Press.

Hutchins, E. and Klausen, T. (1996) Distributed cognition in an airline cockpit. In Y. Engeström and D. Middleton, Eds. *Cognition and Communication at Work*, Cambridge University Press

Ichikawa, Y., Okada, K., Jeong, G., Tanaka, S. and Matushita, Y. (1995) MAJIC Videoconferencing System: Experiments, Evaluation and Improvement. In H. Marmolin, Y. Sundblad and K. Schmidt, Eds., *Proceedings of ECSCW'95*, Dordrecht, Kluwer.

Ishii, H., Kobayashi, M. and Grudin, J. (1992) Integration of inter-personal space and shared workspace: ClearBoard design and experiments. In J. Turner and R. E. Kraut, Eds., *Proceedings of CSCW'92*, New York, ACM Press, 33-42.

Ishii, H. and Ohkubo, M. (1990). Design of TeamWorkStation: A Realtime Shared Workspace Fusing Desktops and Computer Screens. In S. Gibbs and A. A. Verrijn-Stuart, Eds., *Multi-User Interfaces and Applications*. Elsevier.

Jeffries, R., Turner, A. A., Polson, P. G. and Atwood, M. E. (1981). The processes involved in designing software. In J. R. Anderson, Ed., *Cognitive Skills and their Acquisition*. Hillsdale, NJ, Erlbaum.

Jeng, T-S. and Eastman, C. M. (1999) Design Process Management. *Microcomputers in Civil Engineering* **14** (1), 55-67.

Jirotka, M. and Goguen, J. A., Eds. (1994) *Requirements Engineering: Social and Technical Issues*. London, Academic Press.

Joshi, K. and Lauer, T. W. (1998) Impact of information technology on users' work environment: a case of computer aided design (CAD) system implementation. *Information and Management* **34** (6), 349-60.

Karsenty, L. (1996) An Empirical Investigation of Design Rationale Documents. In M. J. Tauber, Ed., *Proceedings of CHI'96*, New York, ACM Press, 150-156.

Kelley, H. J. and Thibaut, J. W. (1969). Group Problem Solving. In G. Lindzley and E. Aronson, Eds., *The Handbook of Social Psychology*. Reading, Ma., Addison-Wesley, 1-101.

Khedro, T. (1999) Collaborative Distributed Building Engineering. *Microcomputers in Civil Engineering* **14** (1), 69-79.

Klein, M. (1997) Capturing geometry rationale for collaborative design. *Journal of Engineering and Applied Science* **1997**, 24-28

Kolarevic, B., Gerhard, S., Hirschberg, U., Kurmann, D and Johnson, B. (1998) An Experiment in Design Collaboration. In *Proceedings of acadia '98*.

Kraemer, K. L. and Pinsonneault, A. (1990). Technology and groups: Assessments of the empirical research. In J. Galegher, R. E. Kraut and C. Egido, Eds., *Intellectual Teamwork: Social Foundations of Cooperative Work*. Hillsdale, New Jersey, Lawrence Erlbaum Associates, 373-404.

Krauss, R. M., Dushay, R. A., Chen, Y. and Rauscher, F. (1995) The communicative value of conversational hand gestures. *Journal of Experimental Social Psychology* **31**, 533-552.

Krauss, R. M., Morel-Samuels, P. and Colasante, C. (1991) Do conversational hand gestures communicate? *Journal of Personality and Social Psychology* **61** (5), 743-754.

Kvan, T. (1997) But is it collaboration? *Keynote address to eCAADe '97*, also at http://arch.hku.hk/people/tkvan/ecaade97.html

Kvan, T., West, R. and Vera, A. (1997) Tools and Channels of Communication. *Third International IFIP WG5.2 Workshop on Formal Design Methods for CAD.*

Langton, S. R. H., O'Malley, C. and Bruce, V. (1996) Actions speak no louder than words: Symmetrical cross-modal interference effects in the processing of verbal and gestural information. *Journal of Experimental Psychology* **22** (6): 1357-1375.

Lee, J. and Lai, K.-Y. (1991) What's in design rationale? *Human-Computer Interaction* **6**, 251-280.

Leiva-Lobos, E., De Michelis, G. and Covarrubias, E. (1997). Augmenting and multiplying spaces for creative design. In S. C. Hayne and W. Prinz, Eds., *Proceedings of Group'97*, New York, ACM Press, 177-186

Lera, S. G. (1983) Synopses of Some Recent Published Studies of Design Processes and Designer Behaviour. *Design Studies* **4** (2)

Lewis, K. (1994) *Markets for Multimedia Groupworking*, (Summary Report, Cortona Consulting.

Lewis, R. (1997) An Activity Theory framework to explore distributed communities. *Journal of Computer Assisted Learning* **13**, 210-218.

Lloyd, P., Lawson, B. and Scott, P. (1996). Can Concurrent Verbalisation Reveal Design Cognition? In N. Cross, H. Christiaans and K. Dorst, Eds., *Analysing Design Activity*. Chichester, Wiley, 437-463.

Loh, L. and Ong, Y.-S. (1998) The adoption of Internet-based stock trading: a conceptual framework and empirical results. *Journal of Information Technology* **13**, 81-94.

Losada, M., Sanchez, P. and Noble, E. E. (1990) Collaborative technology and group process feedback: Their impact on interactive sequences in meetings. In *Proceedings of CSCW '90*, New York, ACM Press,

Lu, I. M. and Mantei, M. M. (1991) Idea Management in a Shared Drawing Tool. In L. Bannon, M. Robinson and K. Schmidt, Eds., *Proceedings of ECSCW'91*, 97-112.

Mackay, W. E. (1995) Ethics, Lies and Videotape. In I. R. Katz *et al.*, Eds., *Proceedings of CHI '95*, ACM Press,

MacLean, A., Young, R. M., Bellotti, V. M. E. and Moran, T. P. (1991) Questions, Options, and Criteria: Elements of Design Space Analysis. *Human-Computer Interaction* **6**, 201-250.

Maher, M. L., Cicognani, A. and Simoff, S. J. (1997) An Experimental Study of Computer Mediated Collaborative Design. *International Journal of Design Computing*, Key Centre of Design Computing, University of Sydney,
http://www.arch.usyd.edu.au/kcdc/cmcd/paper

Maher, M. L. and Rutherford, J. H. (1997) Model for synchronous collaborative design using CAD and database management. *Research in Engineering Design - Theory, Applications and Concurrent Engineering* **9** (2), 85-98.

Maitland, C., Steinfield, C. and Jang, C. Y. (1997) Supporting globally distributed engineering design teams with communication technologies. In *Proceedings of the Global Networking '97 Conference*. Amsterdam, IOS Press, 338-46.

March, L. J. (1984). The Logic of Design. In N. Cross, Ed., *Developments in Design Methodology*. Chichester, Wiley,

Mazijoglou, M., Scrivener, S. and Clark, S. (1996). Representing Design Workspace Activity. In N. Cross, H. Christiaans and K. Dorst, Eds., *Analysing Design Activity*. Chichester, Wiley, 389-416.

McNeil, D. (1992) *Hand and Mind: What Gestures Reveal about Thought*. Chicago, University of Chicago Press.

Medway, P. (1996). Writing, Speaking, Drawing: the Distribution of Meaning in Architects' Communication. In M. Sharples and T. v. d. Geest, Eds., *The New Writing Environment*. Springer, 25-42.

Miller, M. D., Rainer, R. K. and Harper, J. (1997) The unidimensionality, validity and reliability of Moore and Benbasat's relative advantage and compatibility scales. *Journal of Computer Information Systems* **1**, 38-46.

Minneman, S. L. and Bly, S. A. (1991) Managing á trois: a study of a multi-user drawing tool in distributed design work. In *Proceedings of CHI'91*, New York, ACM Press, 217-224.

Moore, G. C. and Benbasat, I. (1991) Development of an Instrument to Measure the Perceived Characteristics of Adopting an Information Technology Innovation. *Information Systems Research* **2** (3), 192-222.

Moran, T. P. and Carroll, J. M., Eds. (1996), *Design rationale: Concepts, techniques and use.* Hillsdale, NJ, Lawrence Erlbaum Associates.

Moran, T. P., McCall, K., v.Melle, B., Pedersen, E. R. and Halasz, F. (1995). Tivoli. In S. Greenberg, S. Hayne and R. Rada, Eds., *Groupware for Real-Time Drawing*. London, McGraw-Hill, 24-36.

Mumford, E. (1983) Participative systems design: practice and theory. *Journal of Occupational Behaviour* **4**, 47 - 57.

Murray, D. (1993) An Ethnographic Study of Graphic Designers. In G. d. Michelis, C. Simone and K. Schmidt, Eds., *Proceedings of ECSCW '93*, Dordrecht, Kluwer Academic Publishers, 295-309.

na-Mora, F. P. and Hussein, K. (1998) Proactive meeting management for distributed collaborative design. *Advances in Engineering Software* **29** (10), 839-849.

Nardi, B. A., Ed. (1996), *Context and consciousness; activity theory and human-computer interaction.* Cambridge, Mass., MIT Press.

NCC (1995) *SSADM 4+: Version 4.2*, Oxford, NCC Blackwell

Newlands, A., Anderson, A. H. and Mullin, J. (1996). Dialog Structure and Cooperative Task Performance in Two CSCW Environments. In J. H. Connolly and L. Pemberton, Eds., *Linguistic Concepts and Methods in CSCW*. London, Springer-Verlag, 41-60.

Nomura, T., Hayashi, K., Hazama, T. and Gudmunson, S. (1998) Interlocus: Workspace Configuration Mechanisms for Activity Awareness. In *Proceedings of CSCW '98*, New York, ACM Press, 19-29.

Obata, A. and Sasaki, K. (1998) OfficeWalker: A Virtual Visiting System Based on Proxemics. In *Proceedings of CSCW '98*, New York, ACM Press, 1-10.

Olphert, C. W. and Harker, S. D. P. (1994). The ORDIT Methodology for Organisational Requirements Definition. In G. E. Bradley and H. W. Hendrick, Eds., *Human Factors in Organisational Design and Management IV*. Elsevier.

Olson, G. M., Olson, J. S., Carter, M. R. and Storrøsten, M. (1992a) Small Group Design Meetings: An Analysis of Collaboration. *Human-Computer Interaction* 7, 347-374.

Olson, G. M., Olson, J. S., Storrøsten, M., Carter, M., Herbsleb, J. and Rueter, H. (1996). The Structure of Activity During Design Meetings. In T. P. Moran and J. M. Carroll, Eds., *Design rationale: concepts, techniques and use*. Hillsdale, NJ, Lawrence Erlbaum Associates, 217-239.

Olson, J. S., Olson, G. M. and Meader, D. K. (1995) What Mix of Video and Audio is Useful for Small Groups Doing Remote Real-time Design Work. In I. R. Katz *et al.*, Eds., *Proceedings of CHI '95*, New York, ACM Press,

Olson, J. S., Olson, G. M., Storrøsten, M. and Carter, M. (1992b) How a Group Editor Changes the Character of a Design Meeting as well as its Outcome. In J. Turner and R. Kraut, Eds., *Proceedings of CSCW '92*, New York, ACM Press, 91 - 98.

Olson, J. S. and Teasley, S. (1996) Groupware in the Wild: Lessons Learned from a Year of Virtual Collaboration. In M. S. Ackerman, Ed., *Proceedings of CSCW '96*, New York, ACM Press, 419-427.

ORDIT (1993) *The ORDIT handbook of tools and techniques*.

Orlikowski, W. J. (1992) Learning from notes: Organizational issues in groupware implementation. In J. Turner and R. Kraut, Eds., *Proceedings of CSCW '92,* New York, ACM Press, 362-369.

Orlikowski, W. J. and Gash, D. C. (1994) Technological Frames: Making Sense of Information Technology in Organisations. *ACM Transactions on Information Systems* **12** (2): 174-207.

Pagani, D. S. and Mackay, W. E. (1993) Bringing Media Spaces into the Real World. In G. di Michelis, C. Simone and K. Schmidt, Eds., *Proceedings of ECSCW '93,* Dordrecht, Kluwer Academic Publishers, 341-356.

Pahl, G. and Beitz, W. (1984) *Engineering Design.* The Design Council/Springer Verlag.

Perry, M. and Sanderson, D. (1998) Co-ordinating Joint Design Work: the Role of Communication and Artefacts. *Design Studies* **19** (3): 273-288

Plowman, L., Rogers, Y. and Ramage, M. (1995) What are workplace studies for? In H. Marmolin, Y. Sundblad and K. Schmidt, Eds., *Proceedings of ECSCW'95,* Dordrecht, Kluwer.

Poltrock, S.E. and Engelbeck, G. (1997) Requirements for a virtual collocation environment. In S.C Hayne and W. Prinz, Eds., *Proceedings of Group '97.* New York, ACM Press.

Popovic, V. (1996). Design Activity Structural Categories. In N. Cross, H. Christiaans and K. Dorst, Eds., *Analysing Design Activity.* Chichester, Wiley, 211-224.

Potts, C. and Catledge, L. (1996) Collaborative Conceptual Design: A Large Software Project Case Study. *Computer Supported Cooperative Work* **5** (4): 415-445.

Powrie, S. E. and Siemieniuch, C. E. (1990). IBC and Cooperative Working in the Automotive Industry. In S. Gibbs and A. A. Verrijn-Stuart, Eds., *Multi-user Interfaces and Applications.* North Holland,

Prescott, M. B. and Conger, S. A. (1995) Information technology innovations: a classification by IT locus of impact and research approach. *Data Base Advances* **26** (2 & 3): 20-41.

Pugh, S. (1981) Concept Selection - A method that works. In *Proceedings of the International Conference on Engineering Design*, Rome, 447-506.

Pycock, J. and Bowers, J. (1996) Getting Others To Get It Right: An Ethnography of Design Work in the Fashion Industry. In M. S. Ackerman, Ed., *Proceedings of CSCW '96*, New York, ACM Press, 219 -228.

Radcliffe, D. F. (1996) Concurrency of Actions, Ideas and Knowledge Displays within a Design Team. In N. Cross, H. Christiaans and K. Dorst, Eds., *Analysing Design Activity*. Chichester, Wiley, 343-364.

Ramage, M. (1999) *The learning way: Evaluating co-operative systems.* PhD thesis, Department of Computing, Lancaster University.

Reeves, B. and Shipman, F. (1992) Supporting communication between designers with artifact-centered evolving information spaces. In J. Turner and R. E. Kraut, Eds., *Proceedings of CSCW'92*, New York, ACM Press, 394-401.

Reynard, G., Benford, S., Greenhalgh, C. and Heath, C. Awareness Driven Video Quality of Service in Collaborative Virtual Environments. In C-M. Karat *et al.* Eds., *Proceedings of CHI'98*. New York, ACM Press. 464-471.

Reynolds, M. (1994) Decision-making using computer conferencing: a case study. *Behaviour & Information Technology* **13** (3): 239-252.

Riseborough, M. G. (1981) Physiographic gestures as decoding facilitators: Three experimenst exploring a neglected facet of communication. *Journal of Nonverbal Behaviour* 5, 172-183.

Rittel, H. W. J. and Webber, M. W. (1984). Planning problems are wicked problems. In N. Cross, Ed., *Developments in Design Methodology*. Chichester, Wiley.

Robertson, T. (1996) Embodied Actions in Time and Place: The Cooperative Design of a Multimedia, Educational Computer Game. *Computer Supported Cooperative Work* **5** (4), 341-367.

Robertson, T. (1997) Cooperative Work and Lived Cognition: A Taxonomy of Embodied Actions. In J. A. Hughes *et al.*, Eds., *Proceedings of ECSCW'9*, Dordrecht, Kluwer, 205-220.

Robson, C. (1993) *Real World Research: A Resource for Social Scientists and Practitioner-Researchers*. Oxford, Blackwell.

Rogers, E. M. (1983) *Diffusion of Innovations*. New York, The Free Press.

Rogers, Y. (1993) Coordinating Computer-Mediated Work. *Computer Supported Cooperative Work* 1, 295-315.

Sanderson, D. (1992) The CSCW implementation process: An interpretative model and case study of the implementation of a videoconference system. In J. Turner and R. Kraut, Eds., *Proceedings of CSCW'92*, New York, ACM Press, 370-377.

Sanderson, D. (1994) Mediating Collaborative Research? *Computer Supported Cooperative Work* 2, 41-65.

Sanderson, D. (1998) Joint technology design at the end of the second millenium: designing to get yoghurt into containers. In F. Darses and P. Zaraté, Eds., *COOP'98*,, Sophia Antipolis, INRIA, 153-163.

Scott-Morton, M. S., Ed. (1991),*The Corporation of the 1990s: Information Technology and Organizational Transformation*. New York, Oxford University Press.

Scrivener, S. A. R., Clark, S. M. and Keen, N. (1994) The LookingGlass Distributed Shared Workspace. *Computer Supported Cooperative Work* 2, 137-157.

Sellen, S. and Buxton, B. (1992) Using Spatial Cues to Improve Videoconferencing. In *Proceedings of CHI'92*, New York, ACM Press, 651-652.

Sharrock, W. and Anderson, R. (1996). Organisational Innovation and the Articulation of the Design Space. In T. P. Moran and J. M. Carroll, Eds., *Design rationale: concepts, techniques and use*. Hillsdale, NJ, Lawrence Erlbaum Associates, 429-451.

Shu, L. and Flowers, W. (1994) Teledesign: Groupware user experiments in three-dimensional computer-aided design. *Collaborative Computing* 1 (1): 1-14.

Siegel, S. (1956) *Nonparametric statistics for the behavioral sciences*. New York, McGraw-Hill.

Simon, H.. A. (1957) *Models of Man*. New York, Wiley.

Simon, H. A. (1984). The structure of ill-structured problems. In N. Cross, Ed., *Developments in Design Methodology*. Chichester, Wiley.

Singley, M. K. and Carroll, J. M. (1996). Synthesis by Analysis: Five Modes of Reasoning that Guide Design. In T. P. Moran and J. M. Carroll, Eds., *Design rationale: concepts, techniques and use*. Hillsdale, NJ, Lawrence Erlbaum Associates, 241-265.

Smith, G. (1996) Cooperative virtual environments: lessons from 2D multi user interfaces. In M. S. Ackerman, Ed., *Proceedings of CSCW '96*, New York, ACM Press, 390-397.

Sonnentag, S. (1996) Planning and knowledge about strategies: their relationship to work characteristics in software design. *Behaviour and Information Technology* **15** (4): 213-225.

Straub, D., Keil, M. and Brenner, W. (1997) Testing the technology acceptance model across cultures: A three country study. *Information & Management* **33**, 1-11.

Streitz, N. A., Rexroth, P. and Holmer, T. (1997) Does "roomware" matter? In J. A. Hughes *et al.*, Eds, *Proceedings of ECSCW'97*, Dordrecht, Kluwer, 297-312.

Stults, R. (1988) *Experimental Uses of Video to Support Design Activities*, No. SSL-89-19), Xerox PARC.

Suchman, L. A. and Trigg, R. H. (1991). Understanding Practice: Video as a Medium for Reflection and Design. In J. Greenbaum and M.Kyng, Eds., *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ, Lawrence Erlbaum Associates, 65-89.

Swanson, E. B. (1994) Information Systems Innnovation Among Organizations. *Management Science* **40** (9): 1069-1088.

Tang, J. C. (1989) *Listing, drawing, and gesturing in design: A study of the use of shared workspaces by design teams*, PhD thesis, Department of Mechanical Engineering, Stanford University.

Tang, J. C. (1991) Findings from observational studies of collaborative work. *International Journal of Man Machine Studies* **34** (2): 143-160.

Tang, J. C. and Isaacs, E. (1993) Why do Users Like Video. *Computer Supported Cooperative Work* **1**, 163 - 196.

Tang, J. C., Isaacs, E. A. and Rua, M. (1994) Supporting Distributed Groups with a Montage of Lightweight Interactions. In R. Furuta and C. Neuwirth, Eds., *Proceedings of CSCW'94*, New York, ACM Press, 23-34.

Tang, J. C. and Minneman, S. L. (1990) VideoDraw: a video interface for collaborative drawing. In *Proceedings of CHI '90*, 313-320.

Thomas, J. C. and Carroll, J. (1978) The psychological study of design. *Design Studies* **1** (1): 5-11

Tornatzky, L. G. and Klein, K. J. (1982) Innovation Characteristics and Innovation Adoption-Implementation: A Meta-Analysis of Findings. *IEEE Transaction on Engineering Management* **EM-29** (1): 28-45.

Turner, P. and Turner, S. (1997) Supporting Cooperative Working Using Shared Notebooks. In J. A. Hughes *et al.*, Eds., *Proceedings of ECSCW'97*, Dordrecht, Kluwer, 281-295.

Turner, P. and Turner, S. (1998) Structuring Scenarios with Activity Theory. In *Proceedings of the First International Workshop on the Design of Work Artefacts*, University of York, September 1998.

Turner, S. and Turner, P. (1995) *Best Practice in the Management of Technical Information*, (Report for ICI Wilton Research Centre. Internal report, MARI Computer Systems.

Ullman, D. G., Dietterich, T. G. and Stauffer, L. A. (1988) A model of the mechanical design process based on empirical data. *Artificial Intelligence for Engineering Design, Analysis, and Manufacturing* **2**, 33-52.

Van de Ven, A. H. and Delbecq, A. L. (1976) Determinants of Coordination Modes within Organizations. *American Sociological review* **41** (April): 322-338.

Vera, A. H., Kvan, T., West, R. L. and Lai, S. Expertise, Collaboration and Bandwidth. In C-M. Karat *et al.* Eds., *Proceedings of CHI'98*. New York, ACM Press, 503-510.

Whitefield, A. (1986) An analysis and comparison of knowledge use in designing with and without CAD. In *Knowledge Engineering and Computer Modelling in CAD - Proceedings of CAD86*, London, Butterworths, Whitefield, A. (1989). Constructing appropriate models of computer users: the case of engineering designers. In J. Long and A. Whitefield, Eds., *Cognitive Ergonomics and Human Computer Interaction*. Cambridge, UK, Cambridge University Press,

Whittaker, S., Geelhoed, E. and Robinson, E. (1993) Shared workspaces: how do they work and when are they successful. *Int. J. Man-Machine Studies* **39**, 813-842.

Wolf, C. G., Rhyne, J. R. and Briggs, L. K. (1992) Communication and information retrieval with a pen-based meeting support tool. In J. Turner and R. E. Kraut, Eds., *Proceedings of CSCW'92*, New York, ACM Press, 322-329.

Yakemovic, K. C. B. and Conklin, E. J. (1990) Report on a Development Project Use of an Issue Based Information System. In F. Halasz, Ed., *Proceedings of CSCW '90, New York, ACM Press*, 105-118.

Yetton, P. W., Johnson, K. D. and Craig, J. F. (1994) Computer-Aided Architects: A Case Study of IT and Strategic Change. *Sloan Management Review* **Summer 1994**, 57-67.

Yourdon, E. (1989) *Modern Structured Analysis*. Englewood Cliffs, NJ, Prentice Hall.

Appendices

Appendix A: Complete list of requirements by application domain

The tables below show the preliminary list of requirements organised by application domain and stakeholder group. The following elements are included:

- 'raw' requirements, expressed as far as possible in technology-neutral terms;
- related current practice;
- the raw requirements mapped to CSCW technology where appropriate;
- any additional comments.

This was a working document and the contents are therefore in note form.

Proposal Preparation (Senior managers)			
Requirement	**Current Practice**	**Technology**	**Comments**
Effective negotiation with other partners	Face to face meetings with the associated heavy burden of travel.	Videoconferencing	Will not meet all requirements for negotiation e.g. "seeing the whites of their eyes", and informal chat. Medium must be very high quality.
Easy retrieval of material to be included in tenders	Individual's knowledge / memory	Shared library of materials	Need for librarian?
Easy viewing of large documents / spreadsheets on screen	Printing or scrolling.	Large screens.	Particularly acute for users of Mac Classics & SEs.
Reduction in the amount of travel.	Not met. Although managers tend to 'wrap up' a number of meetings in one trip.	Videoconferencing, application sharing.	
Better dissemination of information.	Personal contact and file sharing / file transfer.	Email, and or the use of a bulletin board.	
Cost savings.		Reduction in travel & time savings.	
Proposal Preparation (Project team)			
Consistency of styles, templates etc	Physically distributed but not always used. Text has to be reformatted manually.	Application sharing?	Needs everyone to be a skilled WP user.
Improved interaction and co-ordination among authors.	Meetings	Email, telephone conferencing, video conferencing. Editor with annotation. Shared whiteboard / graphics for story-boarding & diagramming.	Telephone conferencing already in place but not widely used. Concern about as to who is listening out of camera shot.
Approval for large docs. quickly & at short notice.	Meetings.	Electronic signatures. Application/doc sharing.	Legal problems? Acceptance by clients?
Need for naming convention & version control	Manual system?	Shared documents with version control.	
Better support for planning the preparation process.	Meetings	Shared planning tool.	Also useful for charts in the document itself.
Need for a 'Please call me' messaging system.	Mac broadcast facility	Electronic alert	
Critical mass of users of email seen as essential, this to be achieved through 1 per desk.	n/a		
A scanner and or OCR for integrating material from external sources.	Retyped.	Supply scanner / OCR.	
More reliable network		More reliable network	
Quick and easy way of recording decisions	Meeting minutes - but decisions go unrecorded in last minute rush.	Decision recording tool.	Only causes problems if there are disagreements etc later.

Software engineering (Project managers)			
Requirement	Current Practice	Technology	Comments
Tools should run across heterogenous platforms & O/S	Use of PC, Sun, Vax, Mips, Alpha, IBM RS600, HP. O/Ss: Solaris 2, SunOS 4.1, DECnet, OSF/1, open VMS, Mac.	Tools should run across heterogenous platforms & O/S as specified.	
Easier communication with rest of company - including the ability to leave messages when recipient not available	Email only within xxxxxx group, otherwise phone, fax, etc.	Company-wide email	Need to encourage awareness and use. Online forms e.g. for progress reports, travel may help.
Enable staff who time share between sites to stay in touch.	Phone, if at all.	Company-wide email	As above
Easier communication external to company - including the ability to leave messages when recipient not available	Email only from xxxxxx, otherwise phone, fax etc	External email and link to Internet	Security implications
Confirmation that messages have been read (not just received).	Manual confirmation	Read receipt on email	
Meetings with external partners without travel but preserving social cues, NVC etc	Travel to meetings	Videotelephony (VT), i.e. point-to-point; or videoconference suite (VCS); or desktop videoconferencing (DVC). By conferencing one means multi-point to multi-point.	May not meet all requirements for negotiation e.g. "seeing the whites of their eyes", and the informal chat over coffee. Whatever the medium it must by very high quality.
Reduce travel to other sites	Travel to other sites	Email, telephone conferencing (TC), VT, DVC. Shared applications including drawing tools for early design sketches. Annotation tools, possibly voice. Snapshots and indexing of videorecording.	TC already in place but not widely used.
Minimise interruptions.	Not.	Electronic comms management tools i.e. sophisticated answering machine facilities.	
Distance working for domain experts	Travel to other sites	Email, telephone conferencing (TC), VT, DVC. Shared applications.	
Multi-site configuration control	Not.	Extension to Lifespan to add multi-site configuration control.	
Reduce time spent on project administration.		Electronic forms and authorisation e.g. for travel	
Shared project log book for important decisions especially early in the design cycle, including diagrams as well as text.	Individual day books (paper)	Shared electronic day book with graphics and possibly pen input, extension of CASE tools to early design decisions.	Use of this must be "natural" and not impose additional overhead. Must not assume typing proficiency.

Software engineering (Chief Systems Engineer, Principal Consultants & Software Development Consultants)			
Requirement	Current Practice	Technology	Comments
On-line processing of change request	Manually(?)	Electronic forms. Electronic signatures?	
Support for occasional home working	n/a		
Improved time for internal approvals		Electronic forms. Electronic signatures?	
Easier access to information stored in project log books.	Manually	Electronic log book - must be as portable as its paper equivalent, and support sketches, graphical information, key word sharing, colour-coding & formatting, possibly numbered pages, printing of all or part.	Paper based searches more effective because of the use of context information, e.g. remembering a particular page layout. Not everyone agrees with the need for an electronic day book.
Support for document reviews.	Manually. Documents marked up & comments minuted.	Shared applications. Support for named, time & date stamped annotation to documents / code etc.	
Support for conferencing to replace all types of travel.	International travel.	VT, DVC for key technical people.	Would not replace all meetings. Concerns raised over loss of informal contacts. Also concerns regarding security.
Picctures / diagrams with non-native English speaking partners / clients.	Travel & physical whiteboard.	Shared electronic whiteboard.	The interactive nature of this application is important.
Improved access to paper documents.		Scan documents?	
Continued need for informal contact in cross-site working.	As is	?	Unlikely to support this. Need to know the working environment & overhearing informal technical exchanges.
Email - required by some but not all.	Some availability.	Improve access to, and awareness of email.	Must be available across platforms & OSs. Security must be addressed. Phone may be quicker .
Access to shared software model across sites.	Work at other site.	Shared applications.	Security an issue.
Reduced travel for this group working on remote design reviews.		Email, TC, VT, DVC - ideally hands-off. Shared applications including drawing tools & whiteboard. Annotation, possibly voice.	
Reduced travel for this group working as remote experts.		Email, TC, VT, DVC - ideally hands-off. Shared applications including drawing. Annotation. Possibly voice. Whiteboard important.	

Software engineering (Senior software engineers & team leaders)			
Requirement	Current Practice	Technology	Comments
All team members to have an email account	n/a	All team members to have an email account	
Easier communication external to company - including the ability to leave messages when recipient not available	Email only from xxxxxx, otherwise phone, fax etc	External email and link to Internet. Simple fast text link?	Security implications
Improved information retrieval from personal log books.	Manually.	Electronic log book - which must be as portable as its paper equivalent, and support sketches and graphical information	
Improved communication when off-site.	Phone & vodaphone	?	
Maintain informal contact with members of distributed teams.	Co-location.	?	
Central library on software routines	Personal libraries.	Central library on software routines	
Will use email if everyone else does.			
Shift in attitude towards collaborative technologies	n/a	dissemination	

Software engineering (Remote/consulting experts)			
Requirement	Current Practice	Technology	Comments
Preserve security of sensitive material	Removable hard disks, access control	Secure network, and current practice.	
Desktop access to MIS	Find, then use dumb terminal.	Link to MIS	
Review documents on-line	On paper	Annotation tools.	
Replace infrequent long distance trips.	Travel.	DVC	(One expert only)
Better communications with other sites.	Travel; meetings; fax.	Email.	Travelling experts tend to 'roll up' a number of meetings in one trip.
Preserving informal contacts & maintain visibility. Personal touch very important.	Travel.	?	DUCK is unlikely to support this. There is a need to maintain a presence. The need for remote experts to sell their skills to their clients.
Better communications with other sites.	Travel; meetings; fax.	Email; TC & application sharing.	
Access to own resources etc remotely.	Not done - or pocket full of floppies!	Universal network; scanner for paper based information.	
Access to Internet.	n/a	Access to Internet.	Speculative.
Running demonstrations remotely.	n/a	Shared apps; network.	Will save on technical support staff travelling. However demo alone rarely sole purpose of visit. May improve internal PR, as consultants seen to be thrifty.
Need for document transfer at anytime.	Shared folders in Macs but cannot when Macs are off-line.	Central server with public access - permanently available. Email.	
Improved information retrieval from personal log books.	Manually.	Electronic log book - which must be as portable as its paper equivalent, and support sketches and graphical information	May keep a paper version too.
Better dissemination & training.	Visits, word of mouth, etc	Email, bulletin board, shared apps.	
Remote software installation & diagnostics.	Travel.	?	
Email used by all.	Patchy use & availability.	Universal email.	
Save time on routine admin of expense, travel etc forms.	Paper & surface mail, & fax.	Electronic forms & signatures.	
Improved document management & availability of paper docs, post, invoices electronically.	Retyped?	Scanner, OCR. Document management tools.	
Improved network bandwidth.		Improved network bandwidth.	

Missing pages are unavailable

Engineering - various (Design Facilities Manager / Project Safety)			
Requirement	**Current Practice**	**Technology**	**Comments**
Reduce time on routine admin, e.g. capital expenditure.	Paper	Electronic forms, electronic signatures.	
Engineering - various (Chief Systems Engineer / Counter-measures Consultant)			
Facilities for engineers off-site (specifically in the US) to maintain contact with home site.	Will happen in the near future.	?	Security paramount.
Maintain security, esp. for external access.	Secure network. Email not used externally. Hand carried documents, secure fax, diplomatic bag.	?	
Improved communication without travel.	File transfer, phone & travelling.	Possibly DVC, whiteboard & shared apps.	
Engineering - various (Project Manager - Ship Design)			
Simplify routine interaction	Fax, phone & memos	Email.	Email provides a record..
Improved access to, and organisation of project records.	Paper based, manual search. Stored centrally, but individuals kept own copies of key docs.	Central server with public access - permanently available.	
Save expensive meetings which are not always productive.		Possibly DVC, whiteboard & shared apps.	Some meetings need to be face-to-face esp with subcontractors.
Engineering - various (Senior Manager - Aquatics)			
Improved communication with other sites without travel. Supervision of remote team.	Meetings	Possibly DVC, whiteboard, shared apps & email with read-receipt & broadcast facilities.	"10% of interaction depends on NVC. "Also tends to roll-up several meetings in one trip.
Electronic diary.	Paper diary & secretaries.	Electronic diary.	
Greater dissemination of available facilities.		Greater dissemination of available facilities.	
Engineering - various (Drafting unit manager)			
Streamline QA procedures.	Paper based.	Electronic forms & authorisation.	
Integrate symbol libraries	Individual libraries.	Rationalise libraries	Reuse very common.
Avoid unnecessary client meetings.		DVC, whiteboard & shared apps	Some f-to-f required e.g. with subcontractors.
Engineering - various (Chief Naval Architect)			
Improved traceability, esp. multiple sites.	Signatures & records of meetings & reviews.	Electronic forms and authorisation?	
Efficient communication		Email.	
Avoid time wasting in meetings.		DVC, whiteboard with snapshot recording & shared apps.	
Preserve security of sensitive material		Secure network, and current practice.	
New tools should be simple to use.		New tools should be simple to use.	

Appendix B: Questionnaire text

This appendix contains the text of the questionnaire as distributed to potential CSCW users at Metre. Note that because of reformatting for the thesis, the layout is not as originally presented.

(Designers Using Co-operative Knowledge)

The DUCK project is part of the DTI sponsored "Computer Supported Co-operative Work" (CSCW) programme.

CSCW is the use of computer technology to enable groups of people in different locations to work more effectively together. DUCK is investigating the potential for CSCW in the practice, management and administration of engineering design, with the emphasis on meeting the needs of users. The project will both exploit existing groupware applications (e.g. e-mail and application sharing) and develop new groupware applications where required.

The DUCK project is committed to "user-centered design": the user requirements will determine the technical specification for DUCK. XXXXXX are providing information about user requirements and later, pilot sites to test out the usefulness of the technology. This questionnaire is a follow up to user interviews conducted with XXXXXX employees across different parts of the marine division. The purpose of this questionnaire is to extend this information gathering process to as many people as possible.

XXXXXX are one of three organisations involved in DUCK; the others are MARI Computer Systems and Paisley University.

The questionnaire has 2 parts to it:

Part 1 asks about your current working practice.

Part 2 asks about your opinions on possible, future groupware applications.

It is important that you answer all questions on the questionnaire, as clearly and as accurately as possible. If you have any queries with the questionnaire or the DUCK project in general, please contact either Susan Turner (MARI group) on xxx xxxx xxxx OR XXX XXXX (XXXXXX) on ext. xxxx

Thank you for your time and effort

Part 1 - Current Working Practice

Personal details

(1) What is your job title, (e.g. System Engineer, Secretary, etc.)?

Please state:____

(2) Where do you work (e.g. XXX 2)? (Please state):____________________

(3) What division and group are you part of? (Please state):__________________

(4) How long have you been with XXXXXX? (Please tick):

0-3 yrs ❐ 3-5 yrs ❐ 5-10 yrs ❐ 10-15 yrs ❐ > 15yrs ❐

(5) Please state briefly, what your main duties and responsibilites are:_______________

____

____

Computer equipment

(6) What make and type of computer do you use most often (e.g. AppleMac Classic, PC 386, etc.)?

Please state:____

(7) How often do you normally use this computer? (Please tick):

At least every day ❐

Every 2-3 days ❐

Once a week ❐

Once a month ❐

Less than once a month ❐

(8) Is this computer networked? (Please tick): Yes ❐ No ❐ Don't know ❐

(9) Where is this computer located? (Please tick):

On your own desk ❐

On someone else's desk ❐

On a 'free' desk ❐

Other (please state):____

(10a) Is this computer shared with others? (Please tick): Yes ❐ No ❐ Don't know ❐

(10b) If "Yes", then how many people share this computer (approximately)?

Please state:____

(11) How frequently do you use this computer for the following? (Please tick):

	Daily	Every 2-3 days	Once a week	Once a month	Less than once a month	Never
Word-processing						
Spreadsheets						
Statistics/Calculations						
Drawing/Graphics						
File transfer						
Accessing MIS						
Diary						
Others (please state):						

Communications

(12) Approximately how often do you normally use the following methods for communicating with clients? (Please tick):

	Daily	Every 2-3 days	Once a week	Once a month	Less than once a month	Never
Telephone						
Fax						
Telephone conferencing						
Face-to-face discussions						
Group meetings						
Postal mail						
Others (please state):						

(13a) Have you ever experienced any of the following communication related problems? (Please tick any that apply):

	Yes	No	Don't know
People unavailable (by phone)			
People unavailable (face-to-face)			
Unable to assess reactions (e.g. by telephone)			
No feedback that communication has been received			
Errors in file transfer			
Slowness of file transfer			
Unable to gain access to expertise/advice			
Other (please state):			

13b. If "Yes", then how serious have these problems been? (Please circle):

	Not at all serious				Very serious
People unavailable (by phone)	1	2	3	4	5
People unavailable (face-to-face)	1	2	3	4	5
Unable to assess reactions (e.g. by phone)	1	2	3	4	5
No feedback that communication has been received	1	2	3	4	5
Errors in file transfer	1	2	3	4	5
Slowness of file transfer	1	2	3	4	5
Unable to gain access to expertise/advice	1	2	3	4	5
Other:____________________	1	2	3	4	5

Travel

(14a) Do you ever have to travel to communicate with clients? (Please tick).

Yes ❒ No ❒ Don't Know ❒ [If "No", please go to question 15.].

(14b) If "Yes", then how often do you normally travel? (Please tick).

At least once a week ❒

Every 2-3 weeks ❒

Once a month ❒

Every 2-3 months ❒

Less than once every 2-3 months ❒

(14c) Please indicate your reasons for travelling. (Tick any that apply).

Need to see others at same location ❒

Need to consult paper documents not available at home location ❒

Need to consult electronic documents not at same location ❒

Prefer face to face contact ❒

Like travelling ❒

Client requests it ❒

Practical reasons (e.g. to see piece of equipment) ❒

Others (please state):____

(14d) Have you experienced any of the following travel-related problems? (Please tick).

Too time consuming ❐

Being away from own office/files ❐

Others (please state):____

(14e) If "Yes", then how serious have these problems been? (Please circle).

	Not at all serious				Very serious
Too time consuming	1	2	3	4	5
Being away from own office/files	1	2	3	4	5
Others:____________________	1	2	3	4	5
____________________	1	2	3	4	5

8.1.1 Day books

(15a) Do you keep a day book or a log book? (Please tick).

Yes ❐ No ❐ Don't Know ❐

15b. If "Yes", do you use your day book for any of the following? (Please tick any that apply).

	Yes	No	Don't know
Making notes of discussions			
Making text notes			
Making sketches			
Performing calculations			
Making lists (of things to do)			
Pasting in items from computer print outs			
Pasting in other items (e.g. contact cards)			
Other (please state):			

PART 2 - Groupware Applications

This section describes 6 different groupware applications. Each description includes examples of how the application could be used, however, the examples are not exhaustive and there are certainly many uses not mentioned.

Please answer the 5 questions (a-e) for each groupware application example by giving your own, personal opinion.

Internal e-mail

E-mail allows you to send and receive text messages to and from other people or groups of people that have e-mail facilities, from a computer. The messages are stored until their recipient logs onto the e-mail system. They can be read on-screen, printed, saved to a file or forwarded to someone else. Files can also be sent using e-mail. Full internal e-mail would allow you to use this service with anyone in the company.

Examples of usage: to find out if anyone has the answer to a technical query, to let other members of your team know of a meeting date, to send comments about a document, etc.

(16a) Have you ever used internal e-mail ? (Please tick).

Yes ☐ No ☐ Don't know ☐

(16b) How useful do you think internal e-mail would be to you if it was available? (Please circle).

Not at all useful				Very useful
1	2	3	4	5

(16c) How often do you think you would use internal e-mail if it was available? (Please tick).

- At least once a day ☐
- Every 2-3 days ☐
- Once a week ☐
- Once a month ☐
- Less than once a month ☐
- Never ☐

(16d) If you had external e-mail, do you think your use of any of the following would decrease? (Please tick).

	Yes	No	Don't know
Telephone			
Fax			
Paper memos			
Face-to-face discussion			
Own site meetings			
Meetings at other sites			
Others (please state):			

(16e) Do you have any additional comments about internal e-mail? If so, then please write them in the space below:

External e-mail

The same as internal e-mail, but with coverage extended outside the company to other organisations or individuals with e-mail. Because messages are stored, e-mail can be used to communicate with colleagues in different time zones. Examples of usage: Communicating with colleagues on collaborative projects

(17a) Have you ever used external e-mail ? (Please tick).

Yes ❐ No ❐ Don't know ❐

(17b) How useful do you think external e-mail would be to you if it was available? (Please circle).

Not at all useful				Very useful
1	2	3	4	5

(17c) How often do you think you would use external e-mail if it was available? (Please tick).

At least once a day ❐

Every 2-3 days ❐

Once a week ❐

Once a month ❐

Less than once a month ❐

Never ❐

(17d) If you had internal e-mail, do you think your use of any of the following would decrease? (Please tick).

	Yes	No	Don't know
Telephone			
Fax			
Paper memos			
Face-to-face discussion			
Own site meetings			
Meetings at other sites			
Others (please state):			

(17e) Do you have any additional comments about external e-mail? If so, then please write them in the space below:

Application sharing

This allows you to share software running on your machine, e.g. a word-processor with one or more other people anywhere on the network in real time. The others are able to see the same view of the document on their computer screens and if desired, each person can take it in turns using the mouse, keyboard, etc. to input to the application. Only one person needs to have the software running on their computer to share it with others.

Examples of usage: suggesting and agreeing amendments to a text documents e.g. part of a bid proposal, or design drawing, with one or two other people, asking for, or providing advice on a problem e.g. in running a particular piece of software.

(18a) Have you ever used application sharing? (Please tick).

Yes ❐ No ❐ Don't know ❐

(18b) How useful do you think application sharing would be to you if it was available? (Please circle).

Not at all useful ... Very useful

1 2 3 4 5

(18c) How often do you think you would use application sharing if it was available? (Please tick).

At least once a day ❐

Every 2-3 days ❐

Once a week ❐

Once a month ❐

Less than once a month ❐

Never ❐

(18d) If you had application sharing, do you think your use of any of the following would decrease? (Please tick).

	Yes	No	Don't know
Telephone			
Fax			
Paper memos			
Face-to-face discussion			
Own site meetings			
Meetings at other sites			
Others (please state):			

(18e) Do you have any additional comments about application sharing? If so, then please write them in the space below:

Shared electronic whiteboard

This provides a shared whiteboard window on the computer screen. Two or more people, using their own computers, can each draw on the whiteboard without having to take turns (just like a real whiteboard where everyone has a marker). In the electronic version each person's input is shown in a different colour or line type and the "pens" (the cursors) are different shapes. You can take a "snapshot" of the whiteboard at any time and save it to a file. It is also possible to use a page of a document e.g. a spreadsheet as the base layer of the whiteboard - you can then draw on top of it, but not change the document itself (like annotating a paper document).

Examples of usage: brainstorming alternative solutions to a design problem, commenting on a proposed budget held in a spreadsheet, explaining a design feature.

(19a) Have you ever used a shared electronic whiteboard? (Please tick).

Yes ❐ No ❐ Don't know ❐

(19b) How useful do you think shared electronic whiteboards would be to you if they were available? (Please circle).

Not at all useful				Very useful
1	2	3	4	5

(19c) How often do you think you would use shared electronic whiteboards if they were available? (Please tick).

At least once a day ❐

Every 2-3 days ❐

Once a week ❐

Once a month ❐

Less than once a month ❐

Never ❐

(19d) If you had a shared electronic whiteboard, do you think your use of any of the following would decrease? (Please tick).

	Yes	No	Don't know
Telephone			
Fax			
Paper memos			
Face-to-face discussion			
Own site meetings			
Meetings at other sites			
Others (please state):			

(19e) Do you have any additional comments about shared electronic whiteboards? If so, then please write them in the space below:

Desktop video-conferencing

This consists of a camera mounted on your computer, together with (typically) a hands-off telephone and a video window of the person to whom one is speaking. Depending on the system, connections can be simple point to point (2 person) or multi point conferencing where there would be a video window for each of several people. Such systems typically run over ISDN and provide a reasonably good level resolution and freedom from screen jitter. Often desktop video-conferencing is integrated with shared applications and an electronic whiteboard so all these facilities can be used together. A photograph* of a monitor showing desktop video-conferencing is shown below.

Examples of usage: in any of the situations suggested for application sharing and electronic whiteboards, but particularly where it is useful to put a "face to a name", e.g. when talking to a previously unknown colleague at another site. Video may also be useful for gauging reactions in a negotiation situation.

*Photograph originally included no longer available

(20a) Have you ever used desktop video-conferencing? (Please tick).

Yes ☐ No ☐ Don't know ☐

(20b) How useful do you think desktop video-conferencing would be to you if it was available? (Please circle).

Not at all useful				Very useful
1	2	3	4	5

(20c) How often do you think you would use desktop video-conferencing if it was available? (Please tick).

At least once a day ☐

Every 2-3 days ☐

Once a week ☐

Once a month ☐

Less than once a month ☐

Never ☐

(20d) If you had desktop video-conferencing, do you think your use of any of the following would decrease? (Please tick).

	Yes	No	Don't know
Telephone			
Fax			
Paper memos			
Face-to-face discussion			
Own site meetings			
Meetings at other sites			
Others (please state):			

(20e) Do you have any additional comments about desktop video-conferencing? If so, then please write them in the space below:

Electronic day book

This could be anything from a personal digital assistant (PDA) like the Apple Newton, through to am integrated electronic notebook, diary, sketchpad and calculator working seamlessly with your current environment. the day book could replace current paper day books by providing the same functionality (and convenience?) as paper whilst adding cut and paste and search indexing annotation facilities. Pen input would would be available for sketches and hand-written notes. You could also allow other people to access your electronic day book as necessary.

Examples of usage: if all members of a design team kept an electronic day book, the material could easily be searched for early calculations, notes of important conversations and so on if this information became important later in the project life cycle, or for re-use of ideas etc. in subsequent projects.

(21a) Have you ever used an electronic day book? (Please tick).

Yes ☐ No ☐ Don't know ☐

(21b) How useful do you think electronic day books would be to you if they were available? (Please circle).

Not at all useful				Very useful
1	2	3	4	5

(21c) How often do you think you would use electronic day books if they were available? (Please tick).

At least once a day ☐

Every 2-3 days ☐

Once a week ☐

Once a month ☐

Less than once a month ☐

Never ☐

(21d) If you had an electronic day book, do you think your use of any of the following would descrease? (Please tick).

	Yes	No	Don't know
Telephone			
Fax			
Paper memos			
Face-to-face discussion			
Own site meetings			
Meetings at other sites			
Others (please state):			

(21e) Do you have any additional comments about electronic day books? If so, then please write them in the space below:

8.1.2 Additional comments

(22) Do you have any additional comments about DUCK or CSCW that you would like to make? If so, please write them in the space below:

Once you have completed this questionnaire please return it by internal mail to:

XXXXXX XXXXXX

XXX 3

Please return your completed questionnaire by **Friday 24th June 1994**

Thank you for completing this questionnaire.

195 - 217

Appendix C: Additional general comments from the Metre user questionnaire

The comments below were contributed to the 'additional comments' section at the end of the DUCK user questionnaire.

Job title (as stated by respondent)	Comment
Analyst programmer	Perhaps people from different sites would actually work together more if some of these facilities were available. I do not so the questionnaire has been difficult for me to fill in.
Business manager	Without having used a number of these services it is difficult to assess the improvement they could make. We need a prototype implementation soon.
Consultant systems engineer	Interesting facilities which would necessitate all personnel to have dedicated PC facilities. I personally think that they could improve efficiency though transmission of commercially sensitive or security sensitive information might be problematic.
Engineer	CSCW is not about having facilities constantly available to users i.e. machines on every desk. There is much emphasis to be placed on the work culture and organisational set up. Where body shopping persists, CSCW will remain of limited value because of cost trade-offs to be made (at least with the initial expense).
General manager	I think that it is becoming increasingly more important to enable groups of people in different locations to work together. People seem to be more and more reluctant to work away from home and I believe CSCW will make it easier and prove that it is both acceptable and entirely possible.
Principal consultant	Cooperative working sounds like a great way of finding even more ways to lose information on a computer system. The requirements for data bandwidth and computational power would strain any IT infrastructure currently in place. More effort on finding ways of transferring text/graphics in slow time would be infinitely more useful.
Principal consultant	You need some widespread lo-tech benefits soon or you will lose your audience.
Principal systems engineer	The most useful electronic tool is a shared access library with good config. control and index/browse so you can see the current baseline state of design etc.
Project manager	I have high hopes that CSCW will enable me to manage multi-site projects successfully in the future. We cannot easily overcome our geographic dispersion so this promises a means of reducing travel costs, personal disruption etc.
Project manager	Suitable training would be required to accompany the introduction of CSCW technology, user training for existing tools has been inadequate. A particularly valuable tool would be a decent project management tool that is integrated with the MIS and is accessible to all project staff and is usable!
Project manager	The first step towards making any of this work is high availability - i.e. a computer on everyone's desk.
Project manager	I don't believe that the cost of the technology is low enough yet to make CSCW cost effective except where circumstances dictate that people cannot be co-located. Full email would be a good first step.

Secretary	Transferring so much on to a PC would I believe increase health and safety problems. A variety of means to communicate - telephone, fax, meetings etc., allows me the opportunity to have a break from the Mac.
Senior analyst	Since it is beyond Metre to equip me with a decent PC/Mac, this questionnaire is somewhat academic. Let's catch up with today's technology before we look at tomorrow.
Senior consultant	The suggested uses are idealistic in the extreme and suggest a lack of pragmatism. "Pie in the Sky" image is exacerbated by the fact that it's difficult to answer "How often would you use..." when at present it's difficult to get even a free Mac or a free telephone tie line.
Senior design engineer	This questionnaire suggests an infatuation with 'problem-solving by technological overkill' that ignores the major problems (cost and time) involved in trying to make the things work. They are mostly expensive toys for the upper management to show off to clients (and their own bosses) to show how clever/up-to-date/on-the-ball they are, and eventually end up in a cupboard.
Software engineer	This sort of technology needs to be introduced at the lower technical levels. If only high level managers get hold of it, it will never be routinely used and the culture won't change sufficiently to see the benefits.
Software engineer	Most of the applications are only useful in conjunction with other applications.
Systems analyst	As application sharing seems to be the crux of the project (all the other facilities are commercially available, it is crucial to establish if there is a definite market (or in-house) need for this aspect of the product. I answered no to the question of potential use. This applies to my current project. My last project theoretically could have used it but what about the security limitations? I cannot currently say whether I would use it in the future but experience would imply yes.
Systems consultant	These systems are only of use if everyone has the same system and on-demand access to the system. A great deal of the information which we deal with is classified - this would present problems.
Systems engineer	As far as Metre is concerned CSCW represents a 'fix' for the underlying problem. There are too many sites! Rationalisation to 1 site north and 1 site south would ease the current problems significantly. However if we adopt Desktop Videoconferencing and I get a decent machine (c.f. SE!) the I'm all for it.
Systems engineer	External Internet would greatly benefit the work I do, as clients are in the US. This would allow me to demo software developments to them without leaving Glasgow. This has two major advantages. 1. It is far cheaper and less risky than flying to the US with software on a computer tape. 2. It would allow frequent demos, keeping the client closely involved with developments.
Systems engineer	What about protection of classified information?
Technical consultant	Prior to any of the glossies mentioned here being procured, it would be nice to have my Mac SE upgraded.

Appendix D: Cost benefit tables

This appendix hold those cost benefit tables not included in section 4.5. of the main body of the report. The explanatory text below is repeated from that section for convenience.

The ORDIT Organisational Impact Cost Benefit Analysis tool provides a technique for assessing user and organisational acceptability of a proposed system by focusing on major work roles and organisational match. The proposed social and technical system is reviewed against a pre-supplied checklist of issues, with the aim of identifying benefits which must be realised, and costs which must be managed, if the new technology is to succeed. Costs and benefits are not viewed in strictly financial terms, but as potential changes for the worse or better. Costs should be viewed as pointers to areas where either the system specification needs modification, or where special care should be taken with user support, training etc., or both. The technique is used in the early stages of specification and highlights any organisational changes or modifications to the technical system which may be required. Each cost or benefit identified is scored on a three-point scale, where 1 represents some benefit/cost, 2 a clear benefit/cost and 3 a major benefit/cost. A change may be result in both benefits and costs, so may be scored in both categories. (A full account of the use of user cost benefit analysis for prospective evaluation in CSCW may be found in Eason and Olphert, 1996.)

Stages 1 and 2 of the procedure record information gathered during the requirements elicitation process about the proposed system, the organisational background and roles of key stakeholders. Stages 3 and 4 assess organisational impact for each user group and the organisation as a whole.

The process was carried out towards the end of the requirements phase with a view to clarifying the requirements and assessing the appropriateness of the technology as then envisaged (including both off-the-shelf tools and applications to be developed later by DUCK e.g. the online design journal). Ideally Stage 3, User Cost Benefit Analysis, should be carried out with a group of representative users. Resource constraints at Metre precluded this: the assessment is based on information gained from the requirements interviews and with the participation of the Aquatics Technical Manager of Metre as user representative. It was hoped that once the pilot project had been identified it would be possible to carry out Stages 3 and 4 again for those particular users, but once again this proved impossible because of time constraints at Metre.

Stage 1: Outline Technical Systems Specification

1. Purpose and Overall Configuration	
Overall rationale	To facilitate group working between remote sites by enhancing communications and information sharing.
Scale (how much of the organisation will be affected)	Initially one or two pilot projects within Marine Division, if successful the whole of the Division and beyond.
Type of system (e.g. mainframe, networked PCs, standalone w/stations etc.)	Networked PCs at first, eventually networked Macs and workstations.
Relation to existing technical system (replace, extend must be compatible with etc.)	Extend existing networks and range of communications tools, introduce groupware, must be compatible with existing technology.

2. Planned Benefits		Priority
What benefits are used to justify the planned expenditure?		
Resource reduction	Reduce time/money spent on travelling.	high
Resource effectiveness	Make better use of staff time by avoiding travel. Facilitate more effective use of existing tools e.g. email.	high
Individual enhancement	Remove overhead of travelling and relocation, improve communication with colleagues, access to information, design history etc.	high - will not succeed without tangible benefits of this type.
Organisational enhancement	Cost savings, improved quality of team work.	medium

3. System Functionality	
What, in global terms, are the main categories of service to be offered to users?	
Reports/enquiry facilities	Electronic forms processing, shared access to local and remote information resources, document management system.
Communication facilities	Telephone conferencing, possibly desktop videoconferencing, shared applications, electronic whiteboard, extension of email, file sharing, shared electronic day book.
Text & graphics processing facilities	As now, but with annotation and application sharing. Shared electronic day book.
Special purpose tools, e.g. CASE tools, CAD software	As now, but with application sharing. Shared electronic day book.
Functions for monitoring/controlling equipment and processes	Possibly workflow software.
What requirements, in global terms, will this place upon users?	
Requirements for using the system	For most facilities, a critical mass of users is essential for their success. Therefore all individuals within the target area who have a role in the design process, its management or administration need to use the system.
Requirements for standardisation of procedures across the organisation	Workflow, electronic forms and possibly BPR may require increased standardisation. Document and application sharing may impose at least *de facto* standardisation of styles and harmonisation of individually customised applications.
Requirements related to security, timing and pace of work etc.	Security is vital in many projects. Users will need to consider security and privacy implications of making information available for others. Additional comms modalities may mean more interruptions, but time management may be easier with less travel.

4. Management Development and Control	
From where will the completed system be managed both on a routine and a developmental basis?	In first instance by DUCK project and Metre Technical Director (PM). Later by Central Facilities?
From where will the system be developed and what plans exist for the management of the project?	By DUCK project, but existing technology to be extended by PM/Central Facilities. Plans are part of DUCK project planning.
What kind of development and implementation strategy is envisaged (e.g. prototypes and trials, phased implementation, big bang, etc.) ?	First trial with off-the-shelf tools in 2 pilot sites. Second trial with DUCK software e.g. electronic day book & such off-the-shelf tools as have been found useful. Full implementation dependent on results & resources.

Stage 2: Organisational Description

1. Organisational Structure and Work Roles	
What is the reporting structure in the part of the organisation to be affected by the system?	For projects: team-member ☞ team leader ☞ project leader ☞ project manager ☞ business manager ☞ business area manager ☞ LOB meeting ☞ divisional meeting ☞ divisional director ☞ Metre company meeting. Detailed monthly reports and forecasts produced on project technical progress, manning and finances. Technical specialists report to business managers? Proposal preparation team responsible to a co-ordinating committee (director level). Approvals for text of bid: specialists (contractual, financial, technical and sometimes risk) ☞ LOB level, ☞ divisional level ☞ top management level (e.g. Chief Naval Architect).
Who are the users of the system?	
Primary users	project team members and their managers technical specialists (remote experts) proposal preparation team members secretaries admin staff finance staff Central Facilities staff
Secondary users (occasional or indirect users)	senior managers (some will be primary users) clients via email
Tertiary users (those affected but without direct access)	clients may have some restricted access to project information store
Non-users but affected by implications, e.g. job displaced by the system	
What changes are envisaged in the work roles or reporting structure by the time the system is implemented (either as a result of the system or because of other organisational changes)?	Introduction of the new system will allow collaborative work between people on different sites. There should be less travel and relocation of staff.

2a. Overall Allocation of Relevant Tasks to Work Roles		
A top level task description of the way in which the activities that the technical system is to support are currently handled in the organisation.		
Task	*Existing Work Role*	*Existing technology*
management reporting system	project leaders and all levels of management	paper reports produced on WP & spreadsheets, MIS system, shared folders over network, meetings[37]
proposal authoring & co-ordination	full-time proposal team & domain experts as part-time contributors	WP, shared folders over network, exchange of disks, Mac broadcast facility, phone, meetings
proposal review & approval	proposal team & senior managers (some as members of co-ordinating committee)	comments on paper, meetings
mechanical engineering design	team members, team leaders, project leaders, tech. specialists	CAD software (PCs & w/stations), calculation software, WP for reports (PCs & Macs), day books, paper drawings, reference manuals, standards etc.
software design	team members, team leaders, project leaders, tech. specialists	CASE tools & usual software engineering tools (PCs, Macs & w/stations), (PCs & Macs), day books, reference manuals, etc.
design team co-ordination and management	team members, team leaders, project leaders	informal communication arising from being in the same office, meetings, Work Instruction Forms, phone if people are off-site, weekly paper reports, project repository of drawings code etc. (on-line, with access control, version control etc.)
high level project management	senior managers	meetings, often involving travel to other sites
liaison with clients, partners	usually project leaders and above	email where available, phone, fax, meetings, paper and electronic documents.
provision of specialist advice	technical specialists acting as remote experts	usually travel to other sites, phone, fax
document production	secretarial staff if not done by authors themselves	WP (Macs & PCs)
technical support	Central Facilities staff	site visits or located on site, phone, memos?
processing of travel requests, expenses etc.	admin. staff	manually

[37] Most meetings will involve staff from several sites - this applies throughout the table

2b. Allocation of functionality table				
	Work roles, access to new functionality, & new tasks implied			
System Functionality	Team member	Team leader/project leader	Manager	Technical Specialist/ Remote Expert
Shared applications	May need to harmonise styles of working. Use shared applications for collaborative tasks where not colocated.	May need to harmonise styles of working. Use shared applications for collaborative tasks where not colocated.	May need to learn WP applications etc. previously handled by secretary. Use shared applications for collaborative tasks where not colocated.	May need to become familiar with a wider range of applications. Use shared applications for collaborative tasks where not colocated.
Document preparation and management system including annotation and version control	Collaborative document preparation especially for proposals.	Collaborative document preparation & review especially for proposals.	Collaborative document preparation & review especially for proposals.	Collaborative document preparation & review especially for proposals.
Shared electronic whiteboard	Use as ordinary whiteboard for group discussions where not colocated.	Use as ordinary whiteboard for group discussions where not colocated.	Use as ordinary whiteboard for group discussions where not colocated.	Use as ordinary whiteboard for group discussions where not colocated.
Email	Use to replace phone, fax etc. in communication with internal & external colleagues. Need to consider security implications in external use.	Use to replace phone, fax etc. in communication with internal & external colleagues. Need to consider security implications in external use.	Use to replace phone, fax etc. in communication with internal & external colleagues. Need to consider security implications in external use. May undertake comms tasks previously handled by secretary.	Use to replace phone, fax etc. in communication with internal & external colleagues. Need to consider security implications in external use.
Bulletin board	Information dissemination & gathering.	Information dissemination & gathering.	Information dissemination & gathering.	Information dissemination & gathering.
Telephone/ Videoconfere ncing	Use instead of meetings for collaborative tasks where not colocated.	Use instead of meetings for collaborative tasks where not colocated	Use instead of meetings for collaborative tasks where not colocated. May need to cope with communications previously filtered by secretary; if video-conferencing replaces contentious meetings will need to develop new negotiation skills.	Use instead of meetings for collaborative tasks where not colocated

2b. Allocation of functionality table - cont'd				
	Work roles, access to new functionality, & new tasks implied			
System Functionality	Team member	Team leader/project leader	Manager	Technical Specialist/ Remote Expert
Shared electronic day book	Use in place of paper daybook. Will need to bear in mind that use by others becomes more likely; may need to conform to project conventions	Use in place of paper daybook. Will need to bear in mind that use by others becomes more likely; may need to conform to project conventions.	Use in place of paper daybook. Will need to bear in mind that use by others becomes more likely.	Use in place of paper daybook. Will need to bear in mind that use by others becomes more likely.
On line information resource and shared document repository	Use for project documentation and consult for up-to-date project & company material. Maintain material in own technical area.	Use for project documentation and consult for up-to-date project & company material. Will need to develop access/version control as necessary. Maintain material in own technical area.	Consult for up-to-date project & company material.. Maintain material in own technical area?	Use for project documentation and consult for up-to-date project & company material.. Maintain material in own technical area.
Workflow software for electronic processing of forms etc.	Use for submission of travel requests, expenses etc.	Use for authorisation & submission of travel requests, expenses etc.	Use for authorisation & submission of travel requests, expenses etc.	Use for submission of travel requests, expenses etc.
Electronic 'signature'		Use for authorisation of travel requests, expenses etc.	Use for signing off drawings etc. & for authorisation of travel requests, expenses etc.	

2b. Allocation of functionality table - cont'd				
	Work roles, access to new functionality, & new tasks implied			
System Functionality	Central Facilities	Secretarial	Admin.	
Shared applications	<u>All new functionality brings additional requirements for installation, training and support</u> Use shared applications for training & remote user support.			
Document preparation and management system including annotation and version control		*Possibly* collaborative document preparation & review especially for proposals.		
Shared electronic whiteboard	For occasional group discussions.			
Email	Communication, help requests and information dissemination.	Own comm-unication & may need to handle for manager	Communication	
Bulletin board	Information dissemination e.g. FAQs.	Information dissemination & gathering & may need to handle for manager.	Information dissemination & gathering.	
Telephone/ Video-conferencing	For occasional group discussions.	?		
Shared electronic day book				

2c. Allocation of functionality table - cont'd				
	Work roles, access to new functionality, & new tasks implied			
System Functionality	Central Facilities	Secretarial	Admin.	
On line information resource and shared document repository[38]	Provision of systems information.	Access to standard company documents, reports etc. as necessary.	Access to standard company documents.	
Workflow software for electronic processing of forms etc.	Use for submission of travel requests, expenses etc.	Use for submission of travel requests, expenses etc.	Use for submission of travel requests, expenses etc.	
Electronic 'signature'			Processing of travel requests, expenses etc.	

[38] Will also be used by functions such as Marketing, PR & QA and other bodies such as the Social Club.

Stage 3: User Cost-Benefit Assessment

3a. Probable outcome

Benefits and costs[39] are scored on a 3 point scale.

1 = some benefit/cost

2 = clear benefit/cost

3 = major benefit/cost

Note that a change can be both a benefit and a cost. For example:

Issues	Change	Benefit	Cost
d. Work pace/deadlines	Easier access to information means more control over pace, but may be more interruptions.	2	1

[39] Costs should be viewed as pointers to areas where either the system specification needs modification, or where special care should be taken with user support, training etc., or both.

User Group: Team Members (Software & Mechanical Engineers)

Issues	Change	Benefit	Cost
1. Job security	May increase with new skills.	1	
2. Job content			
a. Task variety	Increased.	1[40]	
b. Effort required	Initial learning curve. May be more effort to input to EDB[41] but less to retrieve data.		2
c. New skills/old skills lost	New skills acquired for new tools.	1	2
d. Work pace/deadlines	Easier access to information means more control over pace, but may be more interruptions. Increase in speed and efficiency of comms may lead to expectations of shorter deadlines.	1	1
e. Workload			
f. Satisfaction	Routine work may be made easier, allowing more time for more interesting creative tasks.	2	
g. Interdependence	Increased.	2	1
3. Organisational procedures			1
a. Discretion autonomy			1
b.Standardisation/ formality	Information and application sharing *may* mean increased standardisation.	2	1
c. Power and influence	May increase with access to information.	1	1
d. Privacy	Increased ease of access by others to individual's information.	1	3
e. Communications	Much better access to rest of company and external contacts.	3	2
f. Status			
4. Personnel issues			
a. Basic pay			
b. Other rewards	Less need for travel and relocation.	3	1
c. Career prospects		3	1
d. Industrial relations			
	totals	21	17

---

[40] some roles

[41] (Shared) Electronic Day Book

User Group: Team Leaders/Project Leaders

Issues	Change	Benefit	Cost
1. Job security	Unlikely	1	
2. Job content			
a. Task variety	May lose some travel to meetings - this can be both a nuisance and a welcome break in routine	1	
b. Effort required	Initial learning curve. May be more effort to input to EDB but less to retrieve data. Benefits may be more significant for this group.		3
c. New skills/old skills lost	New skills acquired for new tools.	1	1
d. Work pace/deadlines	Easier access to information means more control over pace, but may be more interruptions. Increase in speed and efficiency of comms may lead to expectations of shorter deadlines.[42]	2	2
e. Workload	May be some initial increase in supervising use of DUCK, but DUCK will ease the burden of project admin.	1	1
f. Satisfaction	Routine work may be made easier, allowing more time for more interesting creative tasks.	2	1
g. Interdependence	Increased.	2	1
3. Organisational procedures			
a. Discretion/autonomy		1	1
b.Standardisation/ formality	Information and application sharing may mean increased standardisation.	2	1
c. Power and influence	May increase with access to information.	2	1
d. Privacy	Increased ease of access by others to individual's information.	1	3
e. Communications	Much better access to rest of company and external contacts.	2	
f. Status			
4. Personnel issues			
a. Basic pay			
b. Other rewards	Less need for travel and relocation.	2	1
c. Career prospects		1	
d. Industrial relations			
	totals	21	16

[42]Clashes between meetings may still occur, especially with more senior staff

User Group: Managers

Issues	Change	Benefit	Cost
1. Job security	Unlikely		
2. Job content	Design history more traceable - should be firmer basis for approval, signing-off etc.	2	1
a. Task variety	May lose some travel to meetings - at this level this is welcome.	3	1
b. Effort required	Initial learning curve. (EDB of minor relevance for this group).		3
c. New skills/old skills lost	New skills acquired for new tools.	1	2
d. Work pace/deadlines	Easier access to information means more control over pace, but may be more interruptions. More time available as off-site less often.	3	2
e. Workload			
f. Satisfaction	May increase if higher quality work from staff.	2	1
g. Interdependence	Increased.	2	1
3. Organisational procedures			
a. Discretion/ autonomy			
b.Standardisation/ formality		2	1
c. Power and influence	May decrease if more junior staff have greater access to information. If travel to other sites decreases, may become less visible. May increase if become more accessible.	1	2
d. Privacy	Increased ease of access by others to individual's information.	1	2
e. Communications	Better access to rest of company and external contacts.	3	2
f. Status			
4. Personnel issues			
a. Basic pay			
b. Other rewards	Less need for travel.	3	
c. Career prospects		2	1
d. Industrial relations			
	totals	25	19

User Group: Technical specialists / remote experts

Issues	Change	Benefit	Cost
1. Job security		3	1
2. Job content	Application sharing etc. may make the consultancy role easier.		
a. Task variety	May lose some travel - this can be both a nuisance and a welcome break in routine	3	
b. Effort required	Initial learning curve. May be more effort to input to EDB but less to retrieve data.	1	
c. New skills/old skills lost	New skills acquired for new tools. Information may shift from being in the head to in the machine.	2	3
d. Work pace/deadlines	Easier access to information means more control over pace, but may be more interruptions. Increase in speed and efficiency of comms may lead to expectations of shorter deadlines.	3	1
e. Workload	May generate more frequent informal consultancy - workload may increase but profile may be raised.	3	2
f. Satisfaction	Increased - potentially more use made of expertise.	3	2
g. Interdependence	Increased.	3	1
3. Organisational procedures			
a. Discretion/ autonomy			
b.Standardisation/ formality			
c. Power and influence	May gain higher profile because of greater accessibility, but less on-site presence may counteract this. If expertise is made available on-line, input may be seen as less crucial.	3	2
d. Privacy	Increased ease of access by others to individual's information.		3
e. Communications	Much better access to rest of company and external contacts.	2	2
f. Status		2	1
4. Personnel issues			
a. Basic pay			
b. Other rewards	Less need for travel and relocation.	2	1
c. Career prospects		3	
d. Industrial relations			
	totals	33	19

Appendix E: A review of evaluation methods in CSCW (as of Summer 1994)

This appendix summarises a review of 34 reported evaluations of CSCW technology, ranging from controlled laboratory experiments to field studies of technology in use in real organisations. The review was undertaken in Summer 1994 as part of the preparation for the pilot evaluation at 'Metre'.

The tables contain brief details of the systems under evaluation and the methods uses, together with explanatory notes.

1. Field studies

Authors	Subject of evaluation	Methods used	Notes
Abel 1990	Video/audio link between research labs (Xerox Parc)	Collection of email accounts of experiences & impressions of users, journal kept by author.	
Bikson and Eveland 1990	Year long field study of distributed groups of retirees & employees planning retirement policy with & without email.	Group activities checklist; technological expertise; work status; physical proximity; nature & extent of relationships (detailed account of metrics assessing knowledge & interaction) at start and over the year; group centralisation & roles; participation; modes of contact; log of email message headers (sender, receiver, time & date, not topic); perceptions of technology, task effectiveness, anticipated contact, expectations of retirement - 5 point rating scale used; means & ANOVA carried out. Quotes factors in task collaboration success: high skill & ability; good training & lots of experience; autonomy, participative decision making, co-op working conditions; mutual liking; high level of intra-group communication.	Identifies dimensions of change: formal org. structures; roles; physical & social barriers to communication; electronic islands; social density, integration, centralisation & communication; perceptions & evaluations; use of other media. Notes effects of mutual adaptation, also relationship between high levels of intra-group communication & task group success
Bullen and Bennett 1990; Bullen and Bennett 1991	25 enterprises using various groupware tools	Semi-structured interviews in the workplace. Nothing quantitative.	
Dourish and Bly 1992	Account of the use of Portholes, a video link supporting informal contact between and within Xerox PARC and EuroPARC.	Informal and anecdotal observations; group of 15 users asked to note usage of system over 3 day period and completed questionnaire, including open ended questions about liked & disliked features.	

Finholt et al. 1990	Email usage and task effectiveness in teams of student software engineers	Prior performance level (grades), computer science experience. Pre and post trial questionnaires about likely participation:. *Pre trial*: 7 point Likert scale for level of motivation; Stanford Shyness Survey; McCroskey's Personal Report of Communication Anxiety; prior computer mail use and computer access. *Post trial*: 10 point Likert scales for level of commitment to group 7 point Likert scales for perceived quality of communication; 5 point Likert scales to measure perceived quality of communication. *Communication data*. F-to-F meetings observed at 1 month intervals - content analysis, participation patterns. Self report at monthly intervals of phone calls, memos, f-to-f and email. Email also collected. 10 point Likerts for client evaluation of project success. Individual ratings of each other's performance (% contribution).	Sources referenced for the various measures.
Fish et al. 1990	3 month field experiment evaluating the VideoWindow conferencing system across two common areas in same building.	Dimensions evaluated: transparency (how different it is from face to face), reciprocity (that people across the space can both see and hear each other), privacy, environmental context, social context, conversational regulation, and social relationships. *Video* of f-to-f interaction & interaction over VideoWindow: no of opportunities to interact, no of conversations which actually took place, related to circumstances at the time e.g. if someone was within camera range. Qualitative analysis of the character of interactions. *Questionnaire*: familiarity & liking of each other & with each other's work.	
Galegher and Kraut 1992	Investigation of contingency theory v adaptive structuration; 3 person groups of students with two week deadlines-collaborative writing task. Media: computer/phone, f-to-f, free choice.	*Daily work reports:* checklist of types of work; communication with other group members (time of day, type of contact, duration, who involved, topics, how enjoyable, how productive - 7 point Likert scales); evaluations of difficulty of coordinating work, fairness of division of labour, quality of communication procedures, attachment to other group members - 7 point Likert scales. *Retrospective data - questionnaires:* level of interdependence with rest of project - 7 point scales[43]; extent to which specific tasks completed independently - 7 point scale. *Performance data:* each groups project independently rated.	

[43]Van de Ven, A. H. and A. L. Delbecq (1976). "Determinants of Coordination Modes within Organizations." *American Sociological review* 41(April): 322-338.

Gutek 1990	Discussion of the structural contingency approach as a framework for assessing fit between organisations & technology.	References to other studies where organisational or group effectiveness has been measured as a variable. Examples specifically quoted are: morbidity (operating theatres) computer use, if discretionary user satisfaction sales volume no of forms filled lines of code written (self) perceptions of productivity	Notes that job satisfaction & productivity shown to be unrelated. Also that measures probably have to be tailored to the individual case.
Mackay 1988	Workplace based study of small group of experienced email users prior to introduction of a mail filtering tool (Information Lens).	*Structured interviews:* email usage on that day - no of messages sent/received; no of mail folders, size of inbox, no of distribution list subscriptions. Asked if that day was typical. Open ended questions about problems and successes. *Check on email*: on the above items, allowed to delete confidential messages. *Second set of interviews:* perceived degree of control over mail (allocated to 3 categories); preferences about prioritising/archiving mail (allocated to 2 categories). These results compared against actual and perceived usage from first part of study.	
Nunamaker et al. 1989	Field study of electronic meeting room support system in use at IBM.	System log files, pre and post use questionnaires, observations of sessions, follow up interviews, time for completing project, no of meetings required, length of meetings, administrative and man-hour costs, time/cost estimates for equivalent projects. Outcome variables: *Effectiveness:* quality of session process - evenness of participation from log files, perceptions of aspects of effectiveness from questionnaire, interview questions about results of meetings. *Efficiency*: projected manhours v actual manhours for project and estimate by independent experts, perceptions of effectiveness from questionnaire, interview questions about efficiency and outstanding features of system in use. *User satisfaction:* Utilisation rates & plans to install more meeting rooms, self reports from questionnaire, interviews.	Detailed report of measures.
Orlikowski 1992	Introduction of Lotus Notes in large management consultancy	Unstructured interviews; review of documents; observations of meetings, work & training. Pre & post implementation. No use of logging or quantified data.	
Reynolds 1994	Case study of a six month trial of computer conferencing to replace monthly meetings in an academic department.	*Comparison between equivalent 3 month periods off- and on- line*: No of items of information posted, discussions held, decisions made; no of entries in each of these topics for on-line condition. *Comments from members of department:* initial expectations (collection of written comments), and from tape recorded meetings after 3 and 6 months use of the system.	

Sanderson 1994	Field study of desktop videoconference system for distributed academic research group (5 sites). Already had email & computer conferencing.	Semi-structured interviews before & during implementation period (by telephone & site visits); observations of f-to-f meetings; logbook of events, dates & discussions. (Unclear whether this logbook was kept by each participant or by the author). Also questionnaire exploring collaborative aspects of the project & perceptions of conferencing system.	Qualitative data only.
Sanderson 1992	Use of roll-around video conferencing equipment in large decentralised organisation	Observation of meetings, video recording of one meeting, repeated semi-structured interviews. Noted use of facilities, attitudes, behaviour of participants. No quantified data reported.	
Tang and Isaacs 1993	1. Survey of perceptions of videoconferencing system (videoconferencing suite) 2. Work group across 3 sites with desktop video, email, shared drawing tool. Both real teams at Sun.	1. Questionnaire sent by email: best aspects of VC; worst aspects; ranking of desirable additional features. 2. Pre desktop conferencing: video taped & analysed f-to-f meetings, videoconferences, & phone conference. Noted turn transitions (& collisions, gestures, missed glances, length of turns, back channelling, tone of meetings & humour. Not clear how some of this quantified. With DTC: phone calls logged (no, av. duration), email collected, DTC sessions logged (start & stop, who, conference parameters...), f-to-f logged by team. Videotapes of sample sessions analysed using interaction analysis.) Data analysed for patterns or changes between conditions.	
Watabe et al. 1990	Informal evaluation of Mermaid - a multiparty multimedia desktop conferencing system "in daily use". Specifically meetings in up to 4 locations, 2 to 8 participants, for planning, software development etc.	Performance, perceptions of transmission delays, informal evaluations of the usefulness and usage of each service.	No quantitative data (except transmission speeds)
Yakemovic and Conklin 1990	18 month field study of the use of the IBIS tool for recording decision rationale in software development.	No of IBIS nodes created, no of people using IBIS, identification of problems which probably would have been missed otherwise. Observation of IBIS in use and collection (interviews?)of views of its users.	Data collection & analysis methods not documented.

2. Laboratory experiments

Authors	Subject of evaluation	Methods used	Notes
Austin et al. 1990	Groups of 4-6 students who had previously worked together - 30 minute priortisation task in a computerised meeting room.	*Prior to use:* questionnaire - computer experience and proficiency, influence of each group member, satisfaction with & perceptions about group process (5 point Likert) *During use:* Video recordings made. Note of who took control, time & any verbal exchange. % of total task time each user controlled screen. *Post use:* same questions	
Koven and Radhakrish-nan 1990	Lab based experiment with groups playing Distributed Blackbox game (aim to maximise group score) comparing group with individual performance.	*Quantitative:* time, score, shots, errors etc. " *Qualitative:* adaptation to control mechanisms, confidence level, roles within group.	
Krauss and Fussell 1990	Discussion of relationship between mutual knowledge & communication effectiveness.	Largely referential communication tasks, variables measured are identification accuracy & length, type etc. of descriptions.	
Losada et al. 1990	Groups of students who had previously worked together, 30 & 50 minute management tasks, computer-supported meeting, with & without feedback on socio-emotional interaction & with & without technology.	Detected *patterns* of interactive sequences using time series analysis. On-line coding of behaviour using Bales' coding scheme - dominant/submissive, friendly/unfriendly, task-oriented/emotionally expressive. Focuses on socio-emotional interactions.	
McCarthy and Miles 1990	Case study of the iterative design of a(text-based) conferencing system - group design task.	Audio tape recordings of debriefing sessions, of cooperative design sessions and comprehensive system logs. Qualitative analysis of these to identify usefulness of existing features and requirements for new ones.	

Olson et al. 1992	Lab study of 38 groups (previously worked together) solving a design problem - using a shared group text editor or normal paper based documents & a whiteboard.	*Outcome*: how completely output covered all aspects of assigned task; ease of understanding ideas reported in the document; quality of design, including rating on an 80 point scale. *Satisfaction:* post-session questionnaire: satisfaction with process; satisfaction with results; assess evenness of contributions; identify a leader if one emerged. *Process:* transcripts coded for types of task activity; topic under discussion and time to organise the group; time to clarify ideas; time discussing difficulties with technology and engaging in side discussions	Compared results to findings in field studies: problem solving aspects of tasks were roughly comparable.
Scrivener et al. 1994	Trial of 3 pairs of designers using a shared drawing surface & video overlay. Design tasks.	Questionnaire about perceptions of the system, its usefulness and its ease of use. Possibly also video of the trials though this is not explicitly stated.	
Tan et al. 1991	Effects of 'intellective' and preference tasks and GDSS or other means of support on group consensus in a lab based evaluation.	Consensus change (expressed mathematically, based on pre-meeting and post-meeting comparisons)	
Whittaker et al. 1993	Lab based studies of pairs of subjects executing joint text editing and design tasks using audio only & audio + shared workspace.	*Outcome measures*: time to complete task, number of post-hoc changes, independent ratings/measures of product quality; number of paragraphs completed. *Analysis of oral interaction*: number of turns, content of turns. *Subjective measures*: (rating scales about satisfaction with task and process) Also logged errors & comments.	Experimental design and analysis very well documented.
Wolf et al. 1992	Lab based study of meeting support tool - pen based shared drawing space (all users in same room).. 30 minute design task. 4 groups, 3 of which had previously worked together.	Questionnaire and discussed reactions to system. Meeting was observed and notes taken - communication styles, work styles, use of the drawing space. Time stamped recording of events as documented in the drawing space. Screen and meeting videotaped.	No quantitative measures.

3. Mixed studies

Authors	Subject of evaluation	Methods used	Notes
Beard et al. 1990	Laboratory (students) and field (university department) evaluation of VS, a meeting scheduling tool.	*Laboratory*: sample scheduling task - time to completion of task, error-free response time, % correct answers. *Field*: interviews after 4 weeks' use and a 'focus group' - what sort of meetings scheduled, did visual representation help, what method of scheduling was used, was VS helpful & would user continue with it. Also notes no of users still using the system 2? years after the experiment.	
Brothers et al. 1990	ICICLE, a system to augment the process of formal code inspection within f-to-f environment. Laboratory tests & design of field study reported.	*Laboratory*: simulated code inspection cycle. Pre & post use questionnaires, including general impression of system on 5 point scale, video recording of sessions, analysis of changes in group dynamics. *Field*: bugs found & 'other measures of code quality' + measures as in lab tests.	
Kraemer and Pinsonneault 1990	Meta-study of GDSS and GCSS systems	Numerous - measures themselves not quoted, but variables measured are.	
Nunamaker et al. 1991	Meeting Support Systems developed at University of Arizona - lab & real world settings. Includes group process gains and losses.	Post-use questionnaires including rating of effectiveness & satisfaction compared with non-EMS. Some (undefined) measures of participation, no of ideas generated, and no of critical remarks. Probably interviews.	Evaluation techniques only vaguely described. Need to refer to references for individual studies included in the paper.
Twidale et al. 1994	Multi-user interface generator tool for air traffic control. Informal, formative evaluation with students and real users.	Identified problems with the system through following techniques: *Students:* Observation, analysis of conversation between subjects carrying out an interface design task *Users:* Transcripts of conversations between users exploring the prototype and designers.	Argues that opportunistic, informal evaluation is powerful and cost effective. Emphasis on context.
Vogel and Nunamaker 1990	GDSS at University of Arizona & other sites. Includes case studies with real users & problems.	Time spent on each part of the decision process, feedback about reactions to process (productivity, no of ideas generated, ability to reach consensus). Also interviewed 4 months later to discover how many of decisions implemented. Also on-line pre & post use questionnaires: reactions to process & how it compares to normal meetings (reductions in no of meetings, saving in project time, ideas generated, level of achievement, efficiency). Plans to capture demographic data, subjective evaluations, use of specific modules & time spent in each	

4. References

Abel, M. (1990). Experiences in an exploratory distributed organization. In *Intellectual Teamwork: Social Foundations of Cooperative Work,* J. Galegher, R. E. Kraut and C. Egido, Ed. Hillsdale, New Jersey, Lawrence Erlbaum Associates. 489-510.

Austin, L. C., Liker, J. K. and McLeod, P. L. (1990). *Determinants and patterns of control over technology in a computerized meeting room.* Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90), Los Angeles, California, ACM Press.

Beard, D., Palanappian, M., Humm, A., Banks, D. and Nair, A. (1990). *A visual calendar for scheduling group meetings.* Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90), Los Angeles, California, ACM Press.

Bikson, T. K. and Eveland, J. D. (1990). The interplay of work group structures and computer support. In *Intellectual Teamwork: Social Foundations of Cooperative Work,* J. Galegher, R. E. Kraut and C. Egido, Ed. Hillsdale, New Jersey, Lawrence Erlbaum Associates. 245-290.

Brothers, L., Sembugamoorthy, V. and Muller, M. (1990). *ICICLE: groupware for code inspection.* Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90), Los Angeles, California, ACM Press.

Bullen, C. V. and Bennett, J. L. (1990). *Learning from user experience with groupware.* Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90), Los Angeles, California, ACM Press.

Bullen, C. V. and Bennett, J. L. (1991). Groupware in practice: An interpretation of work experience. In *Computerization and Controversy,* C. Dunlop and R. Kling, Ed. Academic Press.

Dourish, P. and Bly, S. (1992). *Portholes: Supporting Awareness in a Distributed Work Group.* Proceedings of CHI '92, ACM Press.

Finholt, T., Sproull, L. and Kiesler, S. (1990). Communication and performance in ad hoc task groups. In *Intellectual Teamwork: Social Foundations of Cooperative Work,* J. Galegher, R. E. Kraut and C. Egido, Ed. Hillsdale, New Jersey, Lawrence Erlbaum Associates. 291-326.

Fish, R. S., Kraut, R. E. and Chalfonte, B. L. (1990). *The VideoWindow system in informal communications.* Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90), Los Angeles, California, ACM Press.

Galegher, J. and Kraut, R. E. (1992). *Computer-mediated communication and collaborative writing: Media influence and adaptation to communication constraints.* Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'92), Toronto, Ontario, ACM Press.

Gutek, B. A. (1990). Work group structure and information technology: A structural contingency approach. In *Intellectual Teamwork: Social Foundations of Cooperative Work,* J. Galegher, R. E. Kraut and C. Egido, Ed. Hillsdale, New Jersey, Lawrence Erlbaum Associates. 63-78.

Koven, C. d. and Radhakrishnan, T. (1990). IBC and Cooperative Working in the Automotive Industry. In *Multi-user Interfaces and Applications,* S. Gibbs and A. A. Verrijn-Stuart, Ed. North Holland. 61-76.

Kraemer, K. L. and Pinsonneault, A. (1990). Technology and groups: Assessments of the empirical research. In *Intellectual Teamwork: Social Foundations of Cooperative Work,* J. Galegher, R. E. Kraut and C. Egido, Ed. Hillsdale, New Jersey, Lawrence Erlbaum Associates. 373-404.

Krauss, R. M. and Fussell, S. R. (1990). Mutual knowledge and communicative effectiveness. In *Intellectual Teamwork: Social Foundations of Cooperative Work,* J. Galegher, R. E. Kraut and C. Egido, Ed. Hillsdale, New Jersey, Lawrence Erlbaum Associates. 111-146.

Losada, M., Sanchez, P. and Noble, E. E. (1990). *Collaborative technology and group process feedback: Their impact on interactive sequences in meetings.* Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90), Los Angeles, California, ACM Press.

Mackay, W. E. (1988). "Diversity in the Use of Electronic Mail: A Preliminary Inquiry." *ACM Transactions on Office Information Systems* 6(4): 380-397.

McCarthy, J. C. and Miles, V. C. (1990). Elaborating communication channels in conferencer. In *Multi-user Interfaces and Applications*, S. Gibbs and A. A. Verrijn-Stuart, Ed. North Holland.

Nunamaker, J. F., Dennis, A. R., Valacich, J. S., Vogel, D. R. and George, J. F. (1991). "Electronic Meeting Systems to Support Group Work." *Communications of the ACM* **24**(7): 40-61.

Nunamaker, J. F., Vogel, D. R., Heminger, A., Martz, B., Grohowski, R. and McGoff, C. (1989). "Experiences at IBM with Group Support Systems: A Field Study." *Decision Support Systems* **5**(2): 40-61.

Olson, J. S., Olson, G. M., Storrosten, M. and Carter, M. (1992). *How a Group Editor Changes the Character of a Design Meeting as well as its Outcome.* CSCW '92, New York, ACM Press.

Orlikowski, W. J. (1992). *Learning from notes: Organizational issues in groupware implementation.* Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'92), Toronto, Ontario, ACM Press.

Reynolds, M. (1994). "Decision-making using computer conferencing: a cas study." *Behaviour & Information Technology* **13**(3): 239-252.

Sanderson, D. (1992). *The CSCW implementation process: An interpretative model and case study of the implementation of a videoconference system.* Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'92), Toronto, Ontario, ACM Press.

Sanderson, D. (1994). "Mediating Collaborative Research?" *Computer Supported Cooperative Work* **2**: 41-65.

Scrivener, S. A. R., Clark, S. M. and Keen, N. (1994). "The LookingGlass Distributed Shared Workspace." *Computer Supported Cooperative Work* **2**: 137-157.

Tan, B. C., Wei, K. and Raman, K. S. (1991). Impact of GDSS and task type on consensus in small group meetings. In *Collaborative Work, Social Communications and Information Systems*, R. K. Stamper, P. Kerola, R. Lee and K. Lyytinen, Ed. Amsterdam, Elsevier Science Publishers. 33-53.

Tang, J. C. and Isaacs, E. (1993). "Why do Users Like Video." *CSCW* **1**: 163 - 196.

Twidale, M., Randall, D. and Bentley, R. (1994). *Situated evaluation for cooperative systems.* CSCW '94, New York, ACM Press.

Vogel, D. R. and Nunamaker, J. F. (1990). Design and assesment of a group decision support system. In *Intellectual Teamwork: Social Foundations of Cooperative Work*, J. Galegher, R. E. Kraut and C. Egido, Ed. Hillsdale, New Jersey, Lawrence Erlbaum Associates. 511-528.

Watabe, K., Sakata, S., Maeno, K., Fukuoka, H. and Ohmori, T. (1990). *Distributed multiparty desktop conferencing system: Mermaid.* Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90), Los Angeles, California, ACM Press.

Whittaker, S., Geelhoed, E. and Robinson, E. (1993). "Shared workspaces: how do they work and when are they successful." *Int. J. Man-Machine Studies* **39**: 813-842.

Wolf, C. G., Rhyne, J. R. and Briggs, L. K. (1992). *Communication and information retrieval with a pen-based meeting support tool.* Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW'92), Toronto, Ontario, ACM Press.

Yakemovic, K. C. B. and Conklin, E. J. (1990). *Report on a Development Project Use of an Issue Based Information System.* CSCW90: Proceedings of the Conference on Computer-Supported Cooperative Work, ACM.

Appendix F: Sample raw verbal protocol

This is the unedited and uncategorised protocol from Meeting 3 of the Mallard series

Notes:

1. Prior to the meeting, Roger has drawn a set of diagrams on the whiteboard.
2. Times are included in this raw version of the protocol primarily as an index back to the video record. The transcript was subsequently annotated with more precise timestamps.
3. Open brackets " (" indicate utterance fragments which overlap with the speech of another participant.

Time	Spea-ker	Utterance
13:56	R	OK. I thought I'd pretty-print that [laughter]. Probably haven't lost too much that was interesting. Construction on the left hand side but I've left a little bit of space around construction because we may well end up breaking it down [P: mm hm] into models of state, configuration type things, [P: yeah] [M: yeah]. Control, well yes, control, scheduling's permeating all layers as a result of some of the things we've said [P: mm hm] user model, value, evaluation of value, only goes up as far as the abstract model, I haven't taken that through [P: yeah] and then there's execution events whose flavour changes as we go up [P: yeah] and everything else whose flavour changes, we really ought to *indistinct*
13:57	R	So what I propose we do is partly on the basis of this original breakdown in structuring that we should be able to find a home or an equivalence for functional things in these mapping layers [P: mm hm] like parser, concrete to abstract controller, we had one of those but we haven't got a name for it now pmh, we had one of those but we haven't got a name for it now [P: mm hm] and so we ought to be able to put in some things , no real attempt to make them object oriented diagrams, just identify what lumps are in the jam [P: yeah] strawberries or whatever it is, so
13.58	R	Well, let's do an easy one. We're actually saying that the user model values and they'll just go straight through, yeah Scheduling shall we do scheduling [P: yeah] any bids for anything else?
	R	So what we've got down here is a set of Petri net enablings [P:ah ha] so it's in terms of transition IDs and place IDs and token values, all in this domain we're talking about which is in VSML. [P: yeah] So I assert we've got two things going on here there's VDM user model values [M: mm hm] in that case we need token values [P: yeah] and there's basically transition and place IDs being mapped to so can we just say kernel model to user model elements
13.59	M	That's a structure translator is it?
	R	Yeah structure but it's using the same information whereas we'll look at the words later. I think this may be more precise or a subset of it.
14:00	M	Are these two circles on the way up? Or have you deliberately done them on the way up. You *indistinct* the values, but you need to know what elements those values map to.
	R	Well, that's that. Sorry, that was meant to be up and that was meant to be down. [M: right] So what I'm about to do, perhaps I'll do this in a different colour. I mean this is not decent syntax, there's something like that giving us that [P: yeah] [M: mm] and on the way back down I assert we, basically we can say, you know, its a Chinese Restaurant, 3 number 7s [P: yeah] but I guess we can say in principle that it's an element idea [P: yeah] that gets mapped, we're not interested in values.
14:01	P	Yeah *indistinct*
	R	Then scheduling may be dealt with entirely at this layer [P: mmh] but if it's not it passes through without any transformation of the [P: mmh] syntax - written on the fly - there's something optional there. I think we've got a user model to concrete model mapper here [P: mmh] [M: mmh] and potentially a value mapper if there's a display format which is different to an internal format and I've asserted that. [P: yeah] So we've got user model - abstract user model to concrete user model value and we've got abstract user model to concrete user model element in that loose sense. I know we're calling them entities and symbols for these [P: mmh] So I think I'll have a similar structure here. The request both has to map values and things [P: mmh] but the result can be entirely in terms of things [P: mmh] and of course
14:02	M	You said you had a *indistinct* Does that mean that the display converter has to be at that level?
	R	It depends what the display converter is doing.
		BREAK - R GOES OUT OF THE ROOM
14:04	R	Yes. What I was about to say was that I expect to find reuse of these things in different paths you know [P: *indistinct*] right well I'm probably not going to draw it in the same ones, I'm going to run out of space actually... execution?
14:05	P	Execution events?
	R	Well, execution all the way through - say execution events, yeah, yeah, right, right, values, elements and the role of the transitions is going to come out, the role of the things is going to come out [P: yeah] That's sort of coming out, that's part of the element mapping I
	P	I would suspect it would probably be useful to have something that was concerned with that
	R	I think we actually have to do that on the scheduling as well [P: mmh] like it's useful to know not merely that this transition *indistinct*
14:06	P	Yes I
	R	You can extract more than one transition from one thing so that I think it's implicit here but let's just [P: *indistinct*] and *indistinct* role coming here whereas we don't I think have that there
	P	No
	R	Cos there's not a breakdown in that sense. Not different components, there is different views of the same thing [P: mmh] I think. OK. So there's only two things here? These are when it comes out of the kernel
	P	*indistinct* It could be a different structure, possibly, of its own but
	R	Yeah, but the sort of conversions that go on so the *indistinct* kernel model..... user model...... role.... this is going up and it takes both [P: mmh]...... diagram, this.....
14:07	P	Cherry diagram

	R	Don't know what to do with those, let's leave them dangling for the moment. You're not actually showing..... that's OK...
	R	Right. So what comes out then is one ore more because we're not saying they go straight through but I think we have this marshalling of events at the abstract user model layer
14:08	P	Mmm yeah
	R	But what is going to come through ultimately [P: mmh] is events which are typed [P: mmh] they have affected elements at the abstract layer, so that's the process, the flow, the stores, and they have values [P: yeah] of data and what have you and that's it. Well, there's time of course [P: yeah] so it looks to me like we've got those again.
	P	Yeah, looks like it
	R	It's going to end up a very boring diagram
	R	I think I'm drawing radishes.
	P	No, they're definitely cherries. Look at the stalks.
14:09	R	Right. Now if we now look at the construction we're going to be looking at elements and in some cases we're going to be looking at values [P: yeah] and it's values we're going to be doing things with. State type things isn't' it? [P: mmh] So whether we should break construction down or put both things in for the moment I'm not sure [P: mmh]
	R	So going down from construction we've now got concrete to abstract element mapping. Is this the way it's going to work? The way it works now?
14:10	P	Yeah, it has a mapping
	R	Cos its the mapping rules that put it in the repository..... mapping rules (right and...
	M	(*indistinct* We're also going to have concrete user model to abstract user model values. [R: right] Don't care what happens in there for the moment
	P	*indistinct* and where's the text?
14:11	R	So what we've also got in here is a parser - can I just write parser in here?
	P	Yeah
	R	That's just to keep it in the same bubble at the moment [P: mmh yeah] the parser's in some sense closely related to that [P: yeah] and the values are in the same language. They're not the same thing at all, but they're related. When we get down here on construction we say well.. Where's the abstract to graph grammar thingummy jig. That's a good question, whatever it's called.
14:12	M	The rule selector.
	R	The rule selector, yes.
	M	It's not in the diagram, it's this one here.
	R	Good. *indistinct* I suspect the construct messages you get coming out of the abstract user model are after the selection's taken place [P: mmh] yeah that's concrete to abstract which is different because the sort of message you get coming out here is add a process in circumstance 7.3. or whatever [P: mmh] [M: yeah] and that's actually effectively what you've done is selected a graph grammar rule.
14:13	P	Mmh yeah, that's reasonable
	R	And it's sort of in the domain of the abstract user model still isn't it?
	P	Mmh
	R	So this means - I'd like to record somewhere that this functionality belongs in one particular place or another, or perhaps it should be in the text associated with the abstract user model. Could we agree that's where it should be? Does that seem sensible?
	P	Mmh
	M	Yeah
	R	It's nothing to do with the kernel model. OK that's fine, it's just what's been done to the user model.
	P	It requires the mapping layer
	R	It doesn't seem..... otherwise what do you pass through
	P	Mmh
	R	OK. So we've got abstract user model to kernel model elements and abstract user model to kernel model values cos we're going to mark the net, it has to be marked to give the VSL values and we don' t have anything corresponding to the parser because they stop there [P: yeah]
14:14	M	So where's the rule selector?
	R	The rule selector is in here. What I'm saying is that these messages are the rules. Each message is a rule and therefore the rule selector's here. Now that's, you know, it's just popped out as we've drawn this [M: yeah] so I'm not saying it's right. But what we've....
14:15	R	I think the arguments for it are as follows. That in the domain of the abstract model there's a set of primitive changes that you can make to that model like add a process and a flow. But in fact those aren't primitive enough [P: mmh]. There'd be things like add the first flow out of a process and the subsequent flow out of a process and the first flow into the process [P: mmh] It's that sort of thing, mm yeah
	P	(*indistinct*

	R	(*indistinct* things that we... the knowledge to sort out what is entirely in the domain of the abstract model. I think that it's nothing to do with mapping, personally. And that fits in rather well with Polimi's view that what they've got sitting there is a graph grammar converter and you've actually got to tell it [M: *indistinct;* P *indistinct*] which rule to use and give it the parameters associated with that rule so it can actually fire it. [M: *indistinct;* P *indistinct*] I have a 'shove the problems off onto somebody else' type of feeling. [P *indistinct*] Of course the rule application has to be configurable, customisable because the rules are. There has to be some *indistinct* of concepts, actually, because *indistinct*
14:16	P	Using graph grammars to translate is easy. Deciding which grammar rules to use is difficult.
	R	*indistinct* user interface, yes
	P	*indistinct*
	R	We could make it configurable
	P	Yes, we could *indistinct*
14:17	R	I don't think we've properly covered, though what is configuration? Configuration - perhaps it could use the element mapping all the way through [P: yeah], it may be some ancillary information about machines and things [P: yeah] what
	P	That might be defined in terms of the attributes of particular types of elements
	R	Possibly, possibly
	P	I don't think it's excluded here, it's just
	R	I'm a little worried [P: yeah] about the things we've got up here, perhaps we'll look later.
	R	Control, that's rather a bundle [P: mmh] like those breakpoint type things [P: mmh]. Breakpoints when you need conversion, if you
	P	Is that control?
	R	Well, where's set breakpoint? Turn the question round.
	P	Well, at create monitor *indistinct* So is it actually executing control through or is it something else?
14:18	R	It's executing or a translation. [P: yeah] You can do it during execution, something we haven't actually..... we haven't actually separated that out yet. Can I... in this area there's possibly things in two categories. There's those things which apply to the model as a whole in some sense, like run, stop, pause [P: mmh] save, store [M: yeah] which typically I think won't need much if anything in the way of conversion [P: mmh] but here's those things which apply to components of models [P: yeah] like set breakpoint, clear breakpoint, don't know what else. Obviously doing the conversion.
	P	I think.... some of the... set breakpoint I don't think is a primitive at all. I think
	R	No *indistinct*
	P	In the concrete layer you would have a concrete layer, create a monitor, though you may need to translate that monitor into something appropriate at the abstract user model layer, and then that's not an element, because it's not part of the modelling
	R	absolutely
	P	but it's something that you need
14:19	R	It's a component of the execution, but not a component of the model.
	P	And I don't think it's an execution control, because I think that is things like run, stop, etc., when the monitor detects something, it may well choose an execution control.
	R	It's interesting, what you've said about the monitor, creating a monitor. again it's this thing. The user creates a model at the concrete layer [P: mmh] that layer says 'is that a monitor that is purely at this level, or is it one tat we can't handle. If we can't handle it, pass it down.'
	P	Yeah
	R	Which is interesting, this sort of thing.
	P	This sort of thing's come along before and I think that's the sort of thing that has to occur.
	R	Right. so. If we just put in create monitor, then we're doing monitor type operations. [P: yeah] Is that everything we'll be left with to do with the model as a whole and not to do with deeper elements?
14:20	P	No, there are filters as well which you may or may not... co show do filters get into the abstract model?
	R	Yeah
	P	(Maybe these things
	R	(Is the filter not a monitor, in some sense? It's like, if that happens, so tell me
	P	Possibly
	R	It's not the way that we've been looking at it. It's doing
14:21	P	It could be looked at that way
	R	It's doing the same sort of thing
	P	mm hm Yes it is
	R	If you like, it's a filter, it's an action
	P	Yeah. The default is to actually pass things on, but yeah, a filter could be an action. Either pass it on or pass it on with a few changes
	R	Yeah, well why can't you have... the language in theory *indistinct* [P: yeah] so your condition would be either collect it or collect it and throw away. It depends what you want [P: mm hm] and your action can be *indistinct*

	P	There is.... yeah.... I think you can probably do it that way. Something occurred to me which you mentioned before, which you mentioned before, which was about passing events through even though they were filtered and I don't know whether we resolved that. So that was to do with the model layer, whichever model layer needs to know about the event even though the user doesn't update its structure or whatever.
14:22	R	Yeah, we said that if the abstract user model was going to maintain its state, [P: yeah] then we shouldn't throw away events [P: yeah] at the (kernel layer until we'd got to that
	P	(Until we'd got to that. So that may be just a positional thing, it just *indistinct*
	R	But in some sense I see this monitor thing giving us a way of connecting the tools, cos what you do is create a monitor for the third party tool [P: yeah] which says the condition, says what it's interested in [P: mm hm] and the action says, basically says 'hide me' or
	P	Yes, that's possible isn't it.
	R	Is this bordering on a concept *indistinct*
	P	It doesn't seem that unreasonable as long as it does mean... I think that at this level the document might have to be a bit more explicit about what we mean by conditions and actions and what sort of *indistinct*
14:23	R	The one thing against that is.... OK I see two things. One is we're actually not... We have not just got listening 3rd party tools but talking 3rd party tools that could drive the abstract layer. So we threw away our concrete layer for an execution *indistinct* which drove *indistinct* user interface. The concrete layer's not really there. [P: mm hm] Well, it might be involved in setting things up but not really very much once running [P: mm hm] so we actually want control coming from the user interface, it's rather like control... with its own mapping of course.
	P	Well, we're trying to build that into the layers anyway, this *indistinct* could well be that - it doesn't mean it has to be the only exclusive way of doing it.
14:24	R	And the other thing about it is why are we treating these things in a different way to we're treating our own layers in the architecture i.e. why is the communication across these layers monitor driven?
	P	Yeah. I actually had one or two things in the back of my...
	R	I don't think I actually meant that, but it's a, you know, putting an argument up. If we really are saying that, OK where the information comes from and where we're sending it to is just our task.
	P	That means essentially that the monitor, all the converter rules and so on have to be written in terms of monitor language, if we're saying that monitors do all that *indistinct*
14:25	R	I'm not sure that the monitors fulfil the total role of the mapping layer.
	P	But if it had something to do with it...
	R	But it should mean - the pipe line between the two. That is actually an interesting alternative [P: yeah] If we have this fairly general concept of monitor [P: mmh] it could actually be the mapper.
	P	It could be.
	R	I'm not sure I like it cos it's *indistinct* two-way mapping and things, perhaps doesn't
	P	But you could stick them on the arrows there couldn't you or something like that?
	R	Or maybe we could say or own communication across layers is a special case of what we're designing for. We're just building these things in as specials for third-party tools.
	P	I think it's an interesting idea, but I don't think we should put it in. I think we should mention it and say 'one possibility is...' and then think about it because I think its...
14:26	R	It's a bit off the wall for this stage.
	P	Yeah, it is really.
	R	Maybe we should build a more limited concept of monitors in here.
	P	Yeah, monitors need to come and go through and we need to deal with them.
	R	And we've also got control which we said *indistinct* it was element specific it wasn't just general and I've lost it. The answer to the question.
	M	*indistinct*
	R	OK. That we're going to cover under monitors I think. Can I ask a question? When you said towards the end does that mean that everything that's left is
	M	Yeah. *indistinct* the only other thing that I came up with is filters and we just send that *indistinct*
	R	Well, yeah.
	P	I think that filters is quite apt with monitors, it's the more general thing.
	R	The more general communication across the layers
	P	Sounds like a good idea, but not one that's obvious [R: yeah] so I think we'll just leave it out for now and think about it later.
14:27	R	So what we should have had - about to run out of space - so this is model control so there's - no - so there's a
	P	Is it model control or is it just execution control?
	R	OK. What I'm trying to say is that it applies to the whole model and not just the elements. So this is probably not the right term but it can be interpreted as that for the moment, it's the whole model [P: mmh] So it doesn't need any element conversion, that's the point I'm making [P: mmh] and probably no value conversion. [P: right] There should probably be a non-customisable set of things that we can do [P: yeah] run, stop, save and load, quit.
	P	Are save and load part of the construction or not
14:28	R	I meant save and load in terms of state
	P	Yeah, I know, but
	R	But, pass..

	P	*indistinct* which route it takes
	R	*indistinct* this same node is purely in the domain of the concrete model, of course, the same node in the diagram, the same node in the piece of text
	P	Maybe it is as well *indistinct*
	R	Possibly. And then what I want to add here is monitor - creation really, isn't it, what we're talking about.
	P	Yeah, yes it is.
	R	Although let's not *indistinct* just for the moment.
	P	Although can we not edit it once we've created it?
14:29	R	Pass - conceptually at least there's no reason why we can't have monitors at the kernel level [P: mmh] *indistinct* .... Alright I'm asserting for the moment that control doesn't mean conversion under this limited definition of it, but that may prove to be wrong.
	P	Yeah, it depends on whether you include *indistinct* or not doesn't it?
	R	Well, take state. You need to know what the state is [P: yeah] you're dealing with such a chunk
	P	Yeah, OK
	R	But yeah but monitors are about things that can be monitored, elements and values, you know, for the condition and the action. we're not quire sure about the action, but I suspect it can't involve any more than those things anyway.
	P	But coming back to the values and elements thing
	R	Yeah, we've got
14.30	P	We've got those. Is the language the same all the way down? I mean it's nice if it is, but is it?
	R	That's the action side if you like.
	P	Yeah, I mean the condition side, to an extent. Or you've probably expressed conditions in a different sort of way.
	R	OK. I really don't know.
	P	Yeah
	R	I would have though there'd have been a clear correspondence on actions.
	P	You can probably actually just have a parse tree somewhere that represents the action and where you do it depends on where you are and you need to translate the elements of it. But whether translating the elements of it affects the parse tree I'm not sure.
14:31	R	Let's just note that as a... I mean monitors as a whole need more investigation [P: yeah] and they can be pretty primitive things to start with.
	P	They at least have some elements of value conversion and maybe some other sorts of conversion [R: yeah] which we can call voluntary conversion, for want of a better name.
	R	Let's start in this space here. Element value - do we need more than one - as a placeholder for the moment.
	P	As a placeholder that's OK.
	R	Nothing happens between They may stop here [P: mmh] In fact some of them don't even come out of here [P: no] but they're to do with other *indistinct*
14:32	P	The other thing is, it's more than that, actually, because if the monitor monitors condition on a symbol. I know we don't have - we don't necessarily see any use for that at this stage, but if that was it
	R	OK the user has to be able to distinguish between a condition on a symbol and a condition on a thing represented by a symbol.
	P	Yeah, that's true.
	R	and it can be done by calling it the same sort of thing.
	P	Yeah, so it's tricky. Yeah, maybe it's just a view thing - we need to decide what we want there. We'll probably ask the user actually *indistinct* worked out that the user does something, pops it up and says 'Well, what do we want to do with this?' [laughter]
		Pause - R writing
14:33		So actually we've identified element mapping and its inverse [P: yeah] value mapping and its inverse [P: mmh] parsing of text but no inverse [P: yeah] at the moment. Element value again, monitor - bit of an unknown - and *indistinct* value. So there's extra information coming through here which is this sort of type, event type. We're asserting at the moment that's merely a label.
	P	I presume we're actually going to put these into objects at some stage, I would hope, because we need to say we can't just map it.
	R	Yeah. Maybe *indistinct* or are they just operator overloads - are they (conversion operators, there again - they are conceptually
	P	(*indistinct* it's one of these things where
	R	It's better to put them in as transformers
14:34	P	Yeah, and then if they do turn out to be the same you can do that as implementation tricks.
	R	I think operator overload is perhaps an implementation trick, but in some sense it's something we ought not to worry about at the object level. But I think the arguments about persistence and whatever [P: mmh] are probably sufficiently strong to put them in as objects at the moment.
	P	I think so because I think to even do the inverse of that you need the map that you had to start off with. If you put them in as objects you can put them together in some sense.
	R	(And down here *indistinct* all this stuff of course

	P	(Of course
	R	It's not just information it's customisation
	P	Everything's connected.
	R	And down here we've got elements, values both ways and monitors.
	P	Yeah. Easy isn't it *indistinct*
	R	So can we look through, lets... Perhaps we should look at all levels in here, all blocks and see that between here and the things we've already got we've actually found a home for everything.
14:35	P	Yeah.
	R	Well, we haven't addressed customisation as such, we're only talking about use and what will be addresses explicitly in customisation is debug mode where the user has these extra facilities, or things having been interpreted rather than compiled etc - the compiler's virtually ignorant,
	P	Yeah
	R	And so we've got all five layers here. We haven't talked about animation particularly, we've talked about third party tools. We need to identify those things which are *indistinct* - all of those red things, with the possible (exception of monitor conversion.
	P	(*indistinct*
	R	Yeah, monitor conversion's an unknown, but all the rest are rule driven in some sense.
	P	Yeah, the safest thing with monitor conversion is to assume that it does (need *indistinct*
	R	(So here we're actually up in the concrete model
14:36	P	Yeah
	R	Well there's a model, there's an icon repository, that's a set of symbols.
	P	*indistinct*
	R	This is the one I've read. We've got edit functionality, we've got visualisation functionality - they'll both be in the views - we've got animation rules and we've got a few other things that we managed to put in as well.
	P	What's the icon repository?
	R	Well, I guess it's a set of symbols - well, no it's shapes isn't it?
	P	Yeah.
	R	You don't need the icon repository as such - it's the symbols - shapes- you're using. Well it's a stupid thing to have in her anyway. It really was
	P	Good
	R	An icon by the way doesn't mean ... icons. I think it was *indistinct* who talked about these things as glyphs [P: right] They're shapes which - an icon is like iconise window, in that it represents something other than itself [P: yeah] [M: mm hm] I know symbols, shapes do as well, but *indistinct*
14:37	R	This is the concrete to abstract mapping. We've got a parser, driven by syntax rules, which we haven't shown, concrete to abstract controller, which does the concrete to abstract element mapping [P: yeah] and there's the concrete user model. Event controller, we haven't got any such thing in this layer [P: yeah] but we have an animation conductor in the concrete layer, we've got an event controller in the abstract [P: yeah] and this always was a bit of an unknown, but we've got display conversion which is the abstract to concrete value [P: value] converter again driven by display *indistinct*
14:38	R	In the user model, looking at the abstract user model, there's an abstract syntax tree which we haven't put in yet, but we discussed this morning [M: right] there's consistency checkers, though not shown as explicitly [M: mm hm] and there's syntax rules and semantic rules, and we haven't really opened it up to that level [P: mmh] but we've actually got a home for all those things [P: yeah]
	R	And in the mapping from the abstract to the kernel we had an event translator, which basically we've thrown away.
	P	Yeah, we've just said how do you do that *indistinct*
	R	Let me just think that one through.
	P	OK its an event translator.
14:39	R	We've got rules against the elements. Maybe if there is... OK the event that comes through from the kernel [P: mmh] is, for example it's a process start, transition has fired and so the concrete model to the abstract user model event is processed, started executing
	P	Mm hm. There's a slight difference
	R	There's a difference, but it is actually one-to-one.
	P	Yeah, however you could make it configurable one-to-one if you wanted to
	R	Its a bit like... I think what's happening at the concrete layer [P: mmh] where you said I'm going to treat this as a string [P: yeah] I can string match on it [P: yeah] but I don't really have any concept of its meaning [P: yeah].
	P	So you could possibly translate *indistinct* event rules of some sort.
14:40	R	*Indistinct*. The translation of the rôle this things represents a process stating [P: yeah] and therefore an event on this rôle can only be is always is process start.
	P	You only know that by reading something don't you? You only know what the allowed types of things that you can have coming up are [R: yeah] by reading rules.
	R	Yeah there is probably a map there
	P	It may be a trivial map, but
	R	So, I mean, but it seems to me that the event mapping is very, very close to the rôle mapping if not identical [P: mm yeah] *indistinct* , but we probably need to get some words round that

	P	Well, we probably justified putting a rôle mapper in there, in sort of a rôle mapping
	R	We need it for scheduling, and we need it for event translation if we are saying that's the same sort of thing.
14:41	P	In those two cases yeah
	R	Is what we're saying in scheduling terms we're saying well you can schedule process start or you can schedule process end there [P: mm] or whatever [P: yeah] and if that happens is happens is what you've got is process start [P: mmh]. It's identical. There may be other uses for it [P: mmh]. The rôle used on the way down to give you transition specification and again it's the same language. It's the same language.
	P	It's the same thing.
14:42	R	So event translator has gone up into the rôle translation [P: mmh] We've got display rules, that's value conversion, I think. Well, yes, I think there was just an uncertainty which level it should be at [P: right]. Structure-event relationships, well that's just the posited data that's being used structure translator we've got it, that's the element mapper, there's rules that drive it *indistinct* and there's the kernel model. [P: *indistinct*] Down at the kernel level there's the kernel model, the abstract syntax tree isn't there anymore, there's the executor ... Petri net translation executor, schedule's floated up and there's the use model to kernel model, you know the OSL value conversion [P: mmh] which is sitting in this layer. Alright-yes - there's something wrong here I think.
	R	I mean it's not major wrongness, it's alright. When we want to evaluate the user model specification [P: mmh] it's going to have some inputs [P: yeah]. Those inputs are going to come from Petri net places [P: yes]. Therefore the values are in VMSL. I think I actually had this in my first object model down at the bottom. So first we actually have to get the VMSL values [P: mmh] translate them into user model values then evaluate the user model specification [P: mmh] getting a result which is the user model values, then translate that back into VMSL so the VDM value conversion is coming in this both ways here in that mapping layer, in the input layer.
	P	Good, that's good.
14:43	R	*indistinct* on Monday morning when this all falls apart. If we don't touch it, it'll work, so
	P	*indistinct*
	R	Yeah, you're right. Anyway we had VDM value to user model oughtn't we. Actually I hadn't used 2 way arrows in the past, I should separate it out [P: yeah], user model to VDM it's here it's VDM to user model we must have had it.
	P	Yes. We skipped over it.
	R	We skipped over it. It's there
	P	Yeah
14:44	R	And oh it's here as well, it is here
	P	It's there
	R	Yes, that's that one, we had it before
	R	So we'll call it *indistinct* to VDM user.
14:45	R	model values, *indistinct* abstract user model to kernel model value - tidy this up later - so that one goes through these, that one comes to there [P: mmh] actually there's no ... these have rules in common or rules and their inverses if you like, they don't actually share any data I don't think, there's no model specific shift, the value conversion isn't relation specific.
	R	And then down in the bottom here in the repository we've got scheduler, rules, transition rules, value conversion rules, semantic rules of various sorts. Well
	R	That looks quite nice, don't you think we can all go home then, turn that off, that's
14:46	R	At one point this morning I thought this was obvious and trivial, but it wasn't [P: mmh] but it's come out sufficiently cleanly [P: mmh] and with what appears at the moment to be sufficient completeness [P: yeah] to look good.
	P	Do you think that we are probably ready to have objects in there
	R	[R: mmh] looking at that either now or in 5 minutes or whatever 'cos it doesn't seem to be trivial what objects you map those to and what the communication paths between objects are up to the top layer
	R	[R: mmh]
	P	Do you see what I mean? [R: yes] The functions and where you put those functions.
	R	Yeah, OK but my intention was to hereby capture things that had to happen [P: yeah] rather than any of this extra [P: yeah], right but ***Tea Break***
	P	Yeah, seems sensible
		Break
		R has erased one board and made a note
14:59	R	What's the colour code?
	R	dunno,
	P	What's an object look like d' you think?
	R	Rectangular with curved corners *indistinct*
	P	I don't know what are of those is
	R	An object going places, perhaps
	P	I'm going to call this animator even though its *indistinct*, because its shorter [R: mmh][*indistinct passage, facing the board and drawing*]
	M	Well, are you just looking at this construction bit then are you?

	P	No, I'm actually saying those are the only things in the model there that are actually connected to
	R	That actually communicate, yeah
	M	This, is what monitors as well isn't it
	R	Sorry, what's in there
	M	The monitor
	P	Yeah, but we may introduce a connection to the monitor but at the moment the only thing that's actually connected to anything else is the one *indistinct*
	M	OK
15:01	P	'Cos we might, I don't know, we might say that the animator's looking after this, so this monitor is actually responsible for initiating the translation of the thing {M: OK} I don't know what can absolutely be said about this
	R	I mean something has to ... in the sort of area that we were before where some monitors can only make sense at the concrete layer
	P	{ Something has to decide that
	R	{ Something has to make that decision yeah, and the monitor might able to but might be able to
	P	Yeah, and the monitor might able to but might be able to
	R	And is there point in having some equivalent at the concrete layer of a monitor whose *indistinct* should be placed at the abstract or the kernel. Probably not.
15:02	P	Probably not no. Anyway those are the key things at the moment and there are probably important things to think about. So what we've got down there, in this layer. I mean this thing seems to be fairly separate anyway, the parser [R: yeah], so there's probably a parser in there, and they're all changing to red there; the ones at the top probably should have been red anyway, so we have a parser. We'll probably have same, the same problem with things going to wrong place - the cross patterns
	R	Yes, yes, yes. We're going to end up with far fewer things and your cross switch is actually part of it I think, necessarily because you have fewer things.
	P	Probably, yeah.
	R	What you've rid of then is these flows of control through, which you see here will not necessarily control, but you see something flowing through. It won't be quite so obvious.
	P	So do we have ... I think we'll probably have a monitor map of some sort, I don't know how you or is it just a functional thing or something else. Maybe we should leave it for now.
	R	Yeah. Map had connotations of it being instant space information [P: yeah], is that what you meant, as opposed to conversion rules but
15:03	P	That is probably what I meant, I wasn't thinking about the rules at the moment,
	R	{ Let's do something else
	P	{ Let's do something else, OK, the elements and the structural type things. Well, the elements anyway - we can do the values afterwards [R: mmh] - would need to be put somewhere, and I would have called that a map as well, actually some sort of element...
	R	Well it is 'cos its actually about instance data.
	P	OK well it was just a
	R	Lets argue about the names later, because map isn't, map's essentially a data object as opposed to a function or a *indistinct* and this is a two way conversion
	P	Two way conversion The other thing is the value, well I'm not quite sure because that sounds more functional to me, I think ought to be something where you don't need to know any specific information, you can just do it.
15:04	R	It's rule driven but it has no model data in it.
	P	Yeah, so
	R	Why not just call it a value transformer
	P	What as a function?
	R	Yeah
	P	But it's not
	R	Well, the parser's a function.
	P	Yeah, OK.
	R	I mean I think we have some, things can still be objects, transformation objects. Just because it hasn't got state doesn't mean it's not an object.
	P	Yeah possibly.... research issue. So that covers that, and that's everything apart from monitor, so we could say we had a monitor transformer as well.
15:05	R	Yeah. Let's just explore... I'm drawing a distinction here, the difference between the element map and the value transformer [P: mmh] is, if you like, the element map has state information, cos, if you drive it one way, you preserve information that enables you to drive it through the first way anyway so [P: mmh] The value transformer we assert, is not like that [P: mmh] The monitor thing, the monitor converter, are we just looking at monitor actions, and converting them into some other language, or are we talking about the population of monitor objects, and mapping them through in some way to the monitor objects in the other
	P	I think we have to map the monitor objects.
	R	So it's objects and
	P	objects and
	R	and language

	P	and language, yeah. I think that's the mapping object. I think we'll use the element map., in some sense, to do that
	R	Ah, wait a minute, I'm not talking about the elements referred to by the monitors.
	P	Right, OK , sorry.
	R	I'm talking about the monitors themselves.
15:06	P	Ah, right.
	R	Cos we've broken it down - the monitor we've broken down onto element mapping value conversion
	P	Ah, yes of course
	R	And monitor language.
	P	language... it may be just a language thing
	R	That's what I thought and when you said something about maps, I then thought it needed clarifying.
	P	So we can largely transform it.
	R	Maybe, it's just an action transformer, but you did say there was possibly a definition thing as well. OK.
	P	Yeah.
	R	OK monitor transformer.
	P	Well
	R	I mean, I don't like many of these names at the moment.
	P	I'll call it initial action transformer or something.
	R	But it's not. Part of the condition is to do with the elements and values [P: OK] and they ought to go to the element and value transformer
	M	Why don't you just call it a language transformer here or a parser or whatever?
	P	Why? Because it's got
15:07	R	Well, Mike
	P	Well we've got something
	R	We'll review this at length later.
	P	Now I'd say, fairly obviously in green there's got to be some sort of connection here [R: mm hm] I'd have thought, but it may well not be a connection to either of these two, because they may be only monitored by that or something that it talks to.
	R	The parser, it depends on what our notional view is [P: mm hm] but the parser is potentially just off the parsable view.
15:08	P	Yeah, that's true actually. So the parser *indistinct*
	R	Are we going to put directions on these things or are we not going to put them on at the moment?
	P	We're not going to put them on at the moment, until we've got
	R	Ok, yeah
	P	The animator, I'm going to leave it out at the moment. I think the value transformer is probably related in the same way, in some sense, at least for this view, I would have said
	R	Do the values come along as anything other than values of attributes?
	P	That's what I'm wondering about.
	R	In that case, they're subsidiary to elements.
	P	Yeah, so we might say that there was some link there.
	R	It's interesting, that, but
	P	That's the way I was thinking.
15:09	R	Seems tenable, yeah
	P	The animator
	R	Executing events
	P	Executing events. No we don't seem to have anything, any sort of object in there, that's dealing with that sort of thing.
	R	Right, because the element map was a sort of a role map, certainly the lower layers. We specifically said it wasn't a role map at this layer, so there wasn't any concept of roles in this layer. OK so this deals with the contents of the event rather than the event itself. We've got elements, values
	M	Events, could they not refer to things which are actually in the abstract user model, so they have to be converted using the map to something that's in the concrete user model, I would say.
15:10	P	It would use the map, but I'm not convinced that that would be the object you connect to
	R	From the concrete layer
	P	I'm wondering if there might be another object in there, which deals with that sort of thing.
	R	In which layer?
	P	In the mapping layer.
	R	That does what?
	M	An event controller [laughter]
	P	*indistinct*
	R	What do you mean - why do we need controllers now rather than event transformations which is the way we've been pushing ourselves, we've just had transformations in these layers.
	M	The user model would say from this layer, here's my group of elements, and it's this group of elements constitutes some kind of event.
	P	yeah, possibly

15:11	M	So it would have to make a call on this layer to be able to convert them, and this layer would have to pass them back up, rather than having to pass the results of that operation back down again.
	P	So if we thought about an operation just for the sake of concreteness at the moment, there would be some operation somewhere saying accept event, or event something, event this is it type thing [R: mm hm] so what I'm saying is would it make sense for the element map to have an event operation on it, that's the only question I'm asking.
	M	Wouldn't this idea you were talking about beforehand.... *indistinct* Couldn't you have a monitor within the animator which says, I want monitor operations on this element map, particular kinds of operations on this element map which might be connected with events.
15:12	P	I think that sounds a bit confusing and complicated actually. I thought what we were talking about before was we had an event coming in here, the abstract user model would take it, it would do things to its own state and then pass it on. So the sort of thing you've got going up is a *indistinct* value object in its own right, or possibly.... I thought that's what we were talking about.
	R	I mean that if we looked at the example we worked through yesterday or perhaps the day before, about two different approaches on the conversion of animation events, execution events and animation events, then [P: ah ha] the sort of thing we had was we started off with some statement about an execution event [P: mm hm] which referred to elements [P: yeah], we didn't want to talk about values particularly [P: no] but all we ought to be doing here is turning those elements into symbol sets.
	P	That's true actually.
15:13	R	And we ought to be also actually mapping values into a stable form.
	P	So we could possibly have it on the element map in that form, convert element or something like that.
	R	Yeah, OK
	P	Ok, I'm convinced. So the animator talks to that
	R	I don't somehow have this notion of this unconverted event coming into the animator and then the animator going down to this layer and saying convert it [P: no] as it comes through it gets converted.
	P	If we want to say who's calling who's services at this point, then that one's calling that service, that one's calling that service and that one's calling the service here.
	R	It's just taking a message here, *indistinct*
	P	That's the way they're going.
	R	Ok, fine.
	P	And presumably
	R	That's sort of more of a functional sort of thing, though
	P	There's something there, yeah
	R	Synchronous
	P	Synchronous communication.
15:14	P	The only thing that's missing - I mean all these things are included somewhere in that [R: mm hm] The only thing that's missing is how we talk to this one.
	R	OK. can I ask a) tokens, for want of a better word at the concrete layer [P: right] that's just going through the view, not the animator.
	P	That's interesting, actually.
	R	I think it is.
	P	It probably is. They're probably a sort of symbol, but a different sort of symbol. Probably a symbol which doesn't have any static attributes.
	R	It's possible. Yeah, well there may be a shape symbol which actually is transformed merely to attributes [P: mm hm] on the thing with which it's associated.
	P	(Yeah. The structure with that
15:15	R	(This morning we sort of said that state in the abstract model was merely a collection of dynamic attributes [P: yeah] but we didn't actually say whether the elements included in there [P: mm hm] were actually dynamic. [P: mm hm] In SE would a token be in the abstract model as an element or would it
	P	*indistinct*
	R	Or would it be an attribute on the flow
	P	(That's something we could put in
	R	(*indistinct* down to the customiser. It probably isn't crucial at this stage.
	P	So the only other thing is the monitor and the language transformer. I would actually be tempted to either do that, that's not enough, is it? That's not enough actually.
	R	Interesting.
	P	Would the monitor language transformer want to call the element back and say translate some elements please and then go to the value transformer and say translate some values please.... possibly it might be the way it works, that what it implies to me.
	R	If the concrete model layer were doing something like create monitor [P: yeah] we now need to do three conversions there, the elements, any value that's sitting with the conditions on the elements and there's the actions [P: yeah] *indistinct* so there is an operation at this layer - that's create monitor or do we merely have a protocol converter that can't actually deal with bigger things - doesn't know anything about create monitor [P: mm hm] but knows that what's come along with it are three bundles, bundles of three sorts rather, and it converts those but doesn't actually
15:17	P	I'd be tempted to say, to introduce a grey monitor that *indistinct* in the mapping layer. I might be wrong about that.

	R	I have this feeling that we're suddenly going to create all the operations [P: mm hm] in that layer even though some of them will be transparent, we won't be able to do anything about it. [P: mm hm] That layer will have complete functionality [P: mm hm] seemingly, although most of it will go straight through. It may not be a bad thing but... the sort of concept we've got is that if it's really at the level of protocol conversion , almost, it's turning ASCII into EPSIDIC or whatever.
	P	Yeah, so
	R	It doesn't actually have any understanding of the bigger things.
	P	So what you could almost have is a protocol converter object which takes all the messages in and calls on all these subsidiary things [R: yeah] or some subsidiary things, to do different bits.
15:18	R	It doesn't even need to know what sort of messages it can receive [P: yeah] It just needs to know that there'll be things in them which it should probably pass on unchanged and other things in them which say this is an element or this is a value or this is a language statement [P: yeah] and it does those.
	P	If I put that in, would we cross out all of the
	R	I'd actually say that was at a different level of abstraction.
	P	Ok
	R	Let's leave that for the moment, I'm not quite sure, I mean this idea may not work.
	P	Yes
	R	But what it does do, well if you add the functionality in, to the concrete and abstract, you don't actually need to add anything in at the mapping layer.
	P	Yeah
	R	As long as you're only talking about values, elements or whatever you don't need [P: yeah] I mean every time you add a message or an operation [P: mm hm] you don't need to add it in here, which seems to me quite a desirable state of affairs.
	P	*(indistinct*
	R	*(indistinct* conceptually it seems quite reasonable. Presumably if we did do that we'd have some sort of thing here with these things up here and other things in here that were subsidiary and used by that thing. [R: mm hm] that we could call thing [laugh] for want of a better word.
	R	Yeah, lets leave that for the moment, it gets in the way of the
15:20	P	Yeah, that's probably something like that with a query there, exactly where.... [R: yeah] That seems to be that mapping layer [R: mm hm] *indistinct* We have well... we probably want to split this *indistinct*
	R	Rules for what - all the various things
	P	I'm not quite sure how to represent it, but it would be nice to represent the customisable bits of these things in there so that we were clear where the customisation part *indistinct* well, we'd tell you about this bit
15:21	R	I think in some sense it's an implementation decision [P: mm hm] in that the pager, we're probably going to say, isn't conversion rules, it's a YACC'd item.
	P	It's actually a customised object itself which doesn't need any rules.
	R	Yeah, but one could have the notion of, if the way YACC worked was instead of compiling, you actually... it was a rule driven parser [P: yeah] then what you give it is a set of rules, and that actually seems to be a very interesting distinction at this level.
15:22	P	Right
	R	So the value transformer could be rule driven or it could have been generated C++ out of some rules. I think at this level we don't care. [P: mm hm] which also tends to suggest that when we're looking at the super user in debug mode [P: mm hm] when he deliberately wants some things to be driven in a rule driven interpreted sort of mode as opposed to compiler [P: yeah] for his iteration time cos he's actually changing the rules rather than changing [P: yeah] *indistinct* then what we're saying is *indistinct* shove in a rules interpreter.
	P	Right
	R	So at this level of diagrams we don't put them in. I think we actually then do an explanation or whatever, but... It seems we have a generic object here, but implemented in two ways.
15:23	P	So we have to explain that but we don't have to diagram it.
	R	We may well put in diagrams or whatever but I think it covers up what's going on between the layers.
	R	Well, if that's the case then that's probably it then....
		end tape 4, beginning tape 5
15:24	R	At this level you've got *indistinct* the event controller
	P	Yeah *indistinct*
	M	That has to be involved because that's what actually does the mapping, the mapping down to the kernel layer.
	P	Right
15:25	R	It invokes it, right, OK. and then you've got the abstract user model for this layer.
	M	mm hm
	R	I mean you've actually got the abstract user model as an object *indistinct* and whether that incorporates the graph grammar rule selector
	P	It could do, couldn't it?
	R	I think it does.
	P	Yeah, well *indistinct*
	R	Yeah

	R	What's happening with monitors here are they autonomous or is there a monitor monitor ? [laughter] Well, let's, OK
	M	lets
	R	I think we need to introduce user model specification here.
	P	mm hm yeah, it's in the view up there.
	R	mm hm it's not actually part of the abstract user model at this level, and it would be sensible to separate it out because different sorts of things happen to it.
	P	OK
15:26	R	That seems a reasonable set of things to start with.
	P	mm hm
	M	Do we need to draw the elements being mapped? I'm not sure that... that would go through the event controller which would then update the user model.
	R	Why? Why does it go through the event controller? I thought we decided to take construction events out, that what we were talking about under the event controller was execution event control?
	M	So this *indistinct* is operational upon these... the user model.
	R	I would have thought so
	P	Sound sensible. I mean the only reason it doesn't do so at the top is because it's not quite so well defined a concept up there.
15:27	M	*indistinct*
	R	In some sense the things coming into the user model have attributes, and those attributes have their value transferred.
	M	Right.
	R	So you only need to worry explicitly I think about value transformation, at least for those things.
	P	mm hm
	M	So we're going to like present them to the parser?
	R	Well, you didn't show model control in yours, Paul, explicitly, you just said model control went through like a dose of salts without any conversion on the way through.
	P	Without any conversion, yeah.
15:28	R	Model control comes potentially from the viewer and the animator.
	P	Yeah, it does, so do we need something in there to accept it?
	R	Well, we could show a line going through it I suppose. Let's come back to that, but let's do come back to it if we've got time. [P: yeah] We could just show it going through [P: yeah] which I suppose would fit in with this notion of a message gateway [P: yeah] and if there are things in, messages which are tags to do with certain types, then we can do protocol conversion *indistinct*
	P	Sounds reasonable.
	R	So it's a function we could put in. Going back up, as events, execution events, come into the abstract model, the event controller deals with them.
	M	Right.
	R	Then they go back up to the element map.
15:29	P	Yeah, that's the only place to go
	R	With values on them, but the element map *indistinct* value transformation.
	P	Yeah.
	R	Yeah, that looks
	P	sensible.
	M	Shall we talk about this layer, or try and model these abstract monitors?
	P	A comment at this point is that *indistinct* subsidiary to something else.
	R	(Well, *indistinct* well what I was about to say is going down the
	M	(*indistinct*
	P	*indistinct*
	R	But it implies that in your layer it has to be talking to the abstract user model. I really think that it's part of the abstract user model.
	P	(I think it's talking to the abstract user model, or at least part of it.
	R	(*indistinct*
	P	Well I think it depends on how you implement it. Certainly associated with it, in some way, we'll probably... It's in a sense a value transformer, you could consider that as part of the element map if you wanted to.
15:30	R	*indistinct* Yeah, you might want to see an object class in this implementation called GG rules. [laughs] They'll think we're in for the horses.
	P	[laughs] Change it's name to the builder or the galloping *indistinct*
	R	What... we could draw a set of objects in the next mapping layer, which are element map, value transformer...
	P	*indistinct*
	R	There's the role map, which we didn't separate out [P: yeah] and I don't... every time we use the element mapper we use the role part of it. I think [P: mm hm] OK *indistinct* the value transformer, yes value transformer, you missed out scheduler at the abstract user model and of course also at the concrete. [P: yeah] Let's get it at the abstract, so that we know that there's a role there.

	P	*indistinct* except it's connected to an element.
	R	Yeah, that's one way of looking at it.
	P	I don't think I'd actually put it in, but we can consider it anyway.
	R	OK I think you should write scheduler as an object in your abstract user model here.
	P	Yeah
15:32		*pause - M drawing*
	R	I'm just looking at the mapping there on this other board - we've got values going both ways, elements going both ways, and this monitor. So if we just stick in a monitor language transformer in this mapping layer. I don't know hat to do with it for the moment but put it in.
	P	Yeah.... looks more like a jam sandwich for the moment.
	R	It does, yeah.
	P	*indistinct*
	R	Ok so if you're going to draw the lines between the abstract and the mapping..
		pause
15:33	R	Construction first.
	M	There's values between *indistinct* here.
	P	I think they're very closely similar to the concrete layer in that respect.
	R	Yeah, yeah we could get symmetry across the layers, they're not quite... there's no equivalent of parsing here.
	P	I would argue for the role map *indistinct*
	R	I think as it's the roles of the elements I actually think it's part of the map.
	P	Yeah, I think so in the same way that values are.
15:34	R	Well, in different ways that values are.
	P	Well, yes, they're more like types.
	R	Well, I mean each transition.... repeat transition places have a role.
	P	Yeah
	R	And each set of roles is notation *indistinct* whereas values are just some sort of general value domain, nothing to do with models at all, particularly
	P	Yeah, yeah
	R	We'll bundle them later... who wants to take the minutes?
	R	When we schedule we have element roles and values [P: mm hm] The element map, roles goes two ways, up and down, the values only goes up.
	M	Any thoughts?
15:35	R	Stick with green. It is becoming spaghetti, but we can... You know what we're going to end up with, don't you..... precisely the set of boxes that we started off with..... we drew this in February. The values *indistinct* part of the element map. You do definitely lose flow of control through here [P: yeah] causality, which is more explicit in the
	P	Part of it's not naming these message types.
	R	These are control type things I think.
	M	Yeah *indistinct*
	R	Pass. So, user model spec is related actually to value transformer but not the element.
15:36	P	Mmm, that's interesting, isn't it?
	R	What I mean by the user model spec to evaluate you go through value transformation.
	P	Yeah, it's going to have a direct link, isn't it?
	R	Yeah.
	R	I mean it is a spec for an element but I think that's irrelevant.
	M	*indistinct*
	P	You need to know what element it is.
	R	Depends on where you detect which spec it is.
	P	Yeah
15:37	R	Interesting, that may well be rather implementation dependent.
	P	Yeah, well, we know that it's got to do that sometime.
	R	Yeah, two ways, yes
		pause
	P	It is the same set of boxes we had to start with
		laughter
	R	Well at least we understand them a lot better now.
		laughter
	R	We have done something different.
	P	We've got all these subsidiary things,
	P	We've changed the names. The animator used to be a visualiser and *indistinct*. We've done a lot.
15:38	R	OK Let's just do the kernel level and try and convert what's missing. HRTPN I would assert and transition spec, as being sufficient.
	P	Yeah

		pause - M drawing
15:39 15:40	R	HRTPN - generating - is accepting construction events, control commands and generating execution events - so it's two way on the element map cos the events that come back go through the element map as well. The role map's involved here as well, but I think it's better if *indistinct* are either made subsidiary to the element map or incorporated from it. Leave it separated and then it's involved in the HRTPN as well - is that true? Yes, because that tells you which transition spec, which parts of the transition spec, yeah. Transition specs, when they execute, can invoke, can require the evaluation of the user model spec so they drop down through the value transformer and back down through the value transformer and scheduling is captured in the flows we've already got, associations we've already got, between the Petri net and the
	M	*indistinct* internal scheduling
	R	Well, we at one point we ruled that out, though I had a feeling that if we don't want model specific *indistinct* at random or *indistinct* but deterministic, we could handle it down here, but I don't think, I can't see a great deal of point in putting that in that in at the moment because it could equally well be handled at the attributes model layer. I don't think having these things handled at the different layers necessarily implies any performance cost at the moment.
	P	mm hm
15:41	R	I'm not talking about processes, computer process or anything of that nature at the moment, it's still conceptual.
		pause
	P	I think the scheduler there needs to talk up to the element layer.
	R	Only if you put scheduler up in the concrete model, which we have said we will, yeah OK
	P	I'd put it in as subsidiary to the animator anyway cos the animator actually receives events, so
	M	yeah
	R	*indistinct*
	P	*indistinct*
15:42	R	I think we're grouping concerns here rather than necessarily coming up with a final set of objects and names. Two way here.
		pause
	M	Monitors
	R	Monitors, yes. We've dealt with the element value set of monitors. It is purely this set of language recognitions and actions that we're talking about.
15:43	P	We might say that it's not transformed, I don't know.
	R	*indistinct* If we can come up with some sort of relatively generalised or protocol conversion type approach to this, there's no reason why we can't implement new types on it [P: mm hm] for which there is a conversion [P: yeah] and one can be function conversion, probably possibly if we look at domains where protocol conversion is a more natural thing we can actually come up with a sensible structure to this.
	P	Well, the thing is, if you find, say you have a language definition and action, then at some stage you're going to parse it. Now whether you parse it at the concrete user model level and then what you pass down, if you pass anything down, it's always the same thing, it's just executed.
15:44	R	I would say that you have to parse it there [P: mm hm] cos until you've parsed it you can't work out what are the elements and what are the (values
	P	(What are the elements and what are the values. So if you've got to parse it there, there may not be any need, we don't know, to change it, but then there might be. [laughter]
	R	So I think what we're looking at is language key words [P: ah ha] and do we have different languages at the different levels [P: yeah] If we don't, there's a keyword that goes through or at least a symbolised form of it.
	P	Yeah, yeah. So it might just go through. Well the thing is it probably does, but we don't... we know sufficiently little about it for, you know
	P	It's something we can mention, maybe, rather than introduce an object.
	R	Well, we can just say at the moment that we've made an assumption, but we are aware it is an assumption and requires validation or whatever. [P: yeah] They won't even have thought about monitors in that sense.
15:45	P	*indistinct* languages
	R	But it's breakpoints generalised, among other things.
	P	Lot's of other things.
	R	But we won't tell them about all those. [laughter]
	P	I'm copying this down, but I want to put a role down there, it's role mapping. I want to join it up to the element map and cut down the number of *indistinct*
	R	Have you any objections to that, M?
	M	To what, the role map?
	R	Putting the role map in the element map.
	P	I don't know if I'll put it in, but it's talking to it at least. It may be in it.
	M	Yeah, it could be, yeah.
	R	And so it doesn't directly communicate [P: yeah] with the other layers [P: mm hm] ?
15:46		*pause, writing*

	R	Yes, I'm wondering actually about this role map, if there's a place for role in the kernel layer.
	P	Actually in there?
15:47	R	Yeah. Because in some sense it depends on how you implement it. If you have this general notion of there being a transition spec for each transition [P: mm hm] then you don't need to worry about it cos you've got a transition spec for each transition. What the implementation of that is going to look like, however, is a set of generalised transition specs, one per role [[P: mm hm] and transition context information [P: right] one per transition [P: yeah] and transition context information effectively tells you which role and which generic spec you're using, plus all the ancillary data which tell you about the circumstances of this transition, its domestic circumstances.
	P	Is that a customisable part of the kernel model?
	R	Yeah well, transition specs are customisable, but
	P	So isn't that a customisation of that.
15:48	R	Yeah but you use - yeah it is, but it could be separated out as generic. Transition specs and context specific instances - context specific stuff [P: mm hm] it could also be shown as an interpreter if you want to but I liked the way that appeared at this level of abstraction. It's when in the kernel model we've got a transition and we want to interpret it - evaluate its guard or action or time condition [P: mm hm] how we pick up the specification for it. [P: mm hm] Basically what I'm saying is it's associated with the transition ID at some point, associated with that object [P: mm hm] but in fact the context information is that you've got to go through the role to get to specification.
	P	Yes. But you don't ever use the role map, you just use the role.
	R	Well the role map is not a map, it's not a map with elements at different levels.
	P	Isn't it?
	R	I think it's merely telling us what the role is for a transition.
15:49	P	Yeah, right.
	R	But not in terms of elements in the abstract user model. That's the element map.
	P	Yeah. I thought it was a matter of saying, OK so this is the name of, say the event if you like.
	R	OK, alright, that seems
	P	I didn't think it was relating the two at the kernel layer, cos what you're talking about seems to be specifically at the kernel layer.
15:50 15:51	R	Well no, lets get concrete. I mean we can rub it off from here at some point. What I'm trying to say is that if we're along the lines of what we've got there, we've got transition 1. [P: mm hm] Up here we've got process 1. [P: mm hm] In the element map, we've got, if you like - we have some relationship between the *indistinct* and the thing... Somewhere, I'm not sure which of these layers it's in now, I thought it was here, we know that T1 is of type process start, is a role process start, and T2 is a role process end. [P: mm hm] I'm inventing things on the fly [P: mm hm] but I think what you're also saying in addition maybe we've got some knowledge that process start as a role here maps on to something else as a role, as an event type up there [P: yeah] [M: yeah] So when we said role map, I was thinking more of this, you were thinking more of the other. [P: mm hm] Right. So we've got something else somewhere which is proc. start maps on to process started. I know this is trivial, but in principle it may not be trivial.
	P	Yes, I thought that was the point of it actually.
	R	I can't actually think of a non-trivial example at the moment.
	P	Me neither, but
	R	Yeah, OK. So there's now what I'm saying is that in this mapping layer you need to make use of all three bits of information.
	P	Right.
15:52	R	You get transition - you get information about transition comes up from here whether it's scheduling request or whether it's firing and you have to... what element is that transition part of, connected to what's the execution role of that transition and how does that role map, [P: ah ha] what's the conversion of that role up to here? [P: mm hm] So in some sense you see all three of those things are in here. This is actually notation specific [P: mm hm] and it defines a set of types. And those types can be used down here [P: mm hm] and these things are model specific, instance specific. Anyway what I was saying down at this level was actually I need this bit of the information [P: mm hm] in some sense, certainly in implementation terms, in order to pick up on the same thing here. [P: right] I'm certainly not interested in that and I'm not interested in that.
	P	I would have assumed that in that case that what went up into the mapping layer there was actually the transition and its type.
15:53	R	OK, my problem with that is that really that type conceptually has got nothing to do with Petri nets whatsoever.
	P	Well, yeah they can be anything at all, it's just what we were saying at the kernel model layer is that we need to separate out different types of transitions because they have different formats of transition specifications.
	R	But only in implementation terms. Conceptually you don't.
	P	Right.
	R	This is
	M	But once it's been customised they have types though.
	P	Do transitions have types down there?
	R	They don't need to have types down there. What you need to be able to do

	P	Yeah, OK, yeah
	R	is to say that each transition has a specification.
	P	The other reason for giving them types is to make it more efficient so you're not repeating stuff all the time.
	R	That's right so you're not giving them all code.
	P	Yeah
	R	So
	P	I'm not sure where we go
15:54	R	Conceptually, it's the way we drew it originally, I think, but in implementation terms, and we've already got to implementation terms in the work IFAD are currently doing, where they talk about - even in the spec they all talk about transition context information, they're already, you know, constrained.
	P	Well, all they're doing is writing down the implementation that they've already got then. [laughter]
	R	So
	P	So yeah
15:55	R	So the top bit of black there is role conversion, role transformation [P: yeah] and is purely to do with values, if you like [P: yeah] notation concept mapping, so what we've used are transformer type things, or whereas the stuff lower down, obviously the T1 T2 is actually the element map [P: right] but this T1 is proc start, T2 is proc end [P: mm hm] is a map in some sense [P: mm hm] If we have some distinction between maps and transformations [P: mm hm] in that it's generated at construction time.
	P	I don't think, OK, it's an implementation consideration, but it's an implementation consideration that's already been decided, and does it really matter if we put it down the kernel level?
	R	Only that we've floated things like user model spec and things like that *indistinct* at the kernel layer.
	P	That's different because they are in a different notation and they actually make a lot more sense up at the abstract user model. I would have said this is floating somewhere around the border level between the kernel model and the mapping layer.
	R	Yeah
15:56	P	I mean, can we say that there are types Tspecs and there are specifications for those types, a set of specifications?
	R	Effectively we've got typed transitions.
	P	Yeah, effectively hat's what we've said, and what we want to know, going to the map in that case, would be the type of that transition, in which case we'd map that type of transition.
	R	In that case, we wouldn't explicitly put that anywhere down here, you'd hide that inside the HLTP in that case. [P: yeah] In IPTES it was actually out in data. [P: mm hm] There still weren't any types in the Petri net, which was sensible.
	P	I think that's right anyway. I mean there are no types in the abstract user model either, apart from the data *indistinct*
	R	It wasn't even an attribute of a transition.
	P	Yeah, right, yeah.
	R	It was an external piece of mapping.
15:57	P	I would have thought that we've got typed transitions in some sense that behave differently but have things that are what the user defines it does. It's not a very clean Petri net, but it's not a very clean Petri net that's being used anyway, is it?
15:58	R	No. Assuming you have the notion of it being a high level time Petri net and therefore transitions have specifications, [P: yeah] which have by definition predicates, times and actions *indistinct* then it's pure. You can pollute it if you assume that VDM is the language for values and those things, though I suspect that if you construct it carefully, you don't even need to build that in. [P: right] In IPTES we had a compile tome IF-DEF which was the wrong way of doing it cos we could move it, we just had to switch between VDM and just string values. Obviously we couldn't execute string values or anything like that, in effect, we tested the kernel without actually having transition specs. In effect, we just had string values on them. [P: mm hm] I think if you structure that properly with the right classes [P: mm hm] you can structure the kernel so you have this notion of value and the evaluation of things. [P: mm hm] You don't actually need to know what... You might end up having to recompile [P: mm hm] to bring other ones in, but you don't actually have to read code.
	P	Yeah, well I'm not sure. Let's see the two concepts first, I don't see that there's a great
	R	OK, let's leave that as an issue for the moment. Have to get myself another page, it's one more issue. [laughter] *indistinct* should be represented explicitly in the kernel layer...
15:59	R	Is this it, at this pass?
	P	Yeah, we haven't missed anything off the other board? Apart from the fact that we haven't considered this layer which came apart *indistinct*.
	R	Yeah well. Let's not worry about that for the moment. The other approach is to think about scenarios like construct, mark [P: yeah] control *indistinct* very explicitly *indistinct*... the next generation of errors either as responses to things from above or generated during execution events. To be done. Configuration's not there.
16.00	P	It isn't explicitly, no
	R	Test case isn't explicitly there. *indistinct* broke down into primitives of state type things.
	P	*indistinct*
		pause

	R	OK
16:01	P	I think that if we wanted a representation of an execution configuration at the abstract user model layer, then presumably some of these messages are going to be something to do with that [R: mm hm] *indistinct*. There's no good passing symbols down and expecting to assume that you've got...
		pause
16:02	R	But if things like execution configuration come down like... the concrete model you connect A to B. [P: mm hm] A and B are of type symbol. [P: mm hm] The mapping layer doesn't know anything about connect but it knows about symbols [P: mm hm] so it can convert symbol to element [P: mm hm] so what comes into the abstract user model is connect element A to element B. [P: mm hm] You might actually do some processing there so connect probably has to be known to the user modeller to connect control or something like that. [P: mm hm] It then sends connect A element to B element down to the kernel. [P: mm hm] Again this mapper knows nothing about connect but knows about elements [P: mm hm] so it sends down what comes out as connect place [P: mm hm] and there's a possibility of it coming out as a set of things.
	P	Yes, *indistinct*
	R	So we might need to think more carefully about that. [P: yeah] But it comes down connect place so-and-so to place something else and the kernel wants to know about connecting or showing or whatever it is.
	P	Sounds sensible.
	R	Sounds very nice. I'm sure in the harsh light of Monday morning...
	P	Yeah, but the notion of the set of all interfaces can probably be worked out at each level, it probably knows what each set of interfaces are so if it's kept in step, presumably we ought to know *indistinct*
	R	OK what I propose we do is firstly we make sure that we have this recorded. I will take the left whiteboard away and record that, unless you've got that already.
16:03	P	That one?
	R	That left hand one.
	P	I haven't got that.
	R	I'll record that. I'll deal with that. I think M's probably got most of it.
	M	Mostly.
16:04	R	OK. I'll record all that. What I'd actually like is for somebody not only to record that right hand one but draw it on some tool or even on a piece of paper that we can photocopy, but without any further refinement or thought so that it's a shared record. [P: yeah] I think that one's an important one because this on the route there we might have to come back to at some point. [P: mm hm] And then on Monday we have to make some definite plans for actually wrapping this up and getting it into a document to send out on Wednesday I think. Right. It's going to be a rush job but I have to have something. I have to be able to talk about this and make decisions with people at the meeting, which is unfortunate, because it would be nice to involve the team on that but I think we need to have something in writing to give them ample preparation time. And the sort of thing we can have, it can start with the five layer thing we've got here, perhaps six with the meta-model.
	P	We can attach descriptions to each of the objects and perhaps examples of the services they provide or various
	R	I think what would be nice would be if we draw each layer actually separately.
	P	Yeah, and then we ought to put connections.
16:05	R	Yes, maybe connections or possibly message sequence charts so we could show construction. [P: maybe] So we could show the view connecting with the parser, then the element map [P: yeah] then inherently there, the value transformer [P: mm hm] coming down to the abstract user model, user model spec [P: yeah], possibly the graph grammar rule selector, or possibly we're including that, coming down through the element map, with the role map and the value transformer coming down to the HRTPM [P: yeah] and generating transmission specs... which is missing
	P	Yes *indistinct* what's a transition spec?
	R	Well notionally, every time you create a Petri net transition, [P: mm hm] you create a transition spec for it. You... what you may be doing is creating transition context information [P: yeah] which goes with the generic spec, which is part of customisation [P: mm hm] that is sort of by the way. You've got to create something every time you create a Petri net transition
	P	Which is what, exactly?
16:06	R	Which is the specification of its behaviour.
	P	Right. And what's the difference between that and the user model spec.
	R	Because it's to do with the mapping of the notation down to the Petri net. Petri nets are such crude, primitive notations without transition specs [P: right] that you can't actually capture anything in them except the simplest finite state machine.
	P	We've got the transition context information [R: OK] What's the other bit. Is that
	R	It's the generic transition specs which are produced by the customiser
	P	Right.
	R	Either way it's something at this level. Either it's the transition spec if we're taking an abstract view, or it's the transition context information [P: yeah] that actually has to be generated by
	P	From the element map
	R	The element map's a bad term. It's the element transformer.
16:07	P	Yes, OK, yeah
	R	And it has a persistent state, which is the element map.

	P	Yes, OK, fine.
	R	And the element transformer transforms... it's driven by members which are graph grammar rules, so you trigger an operation with the associated data which has to go with that operation, it then generates sub-nets, joining them into the pre-exiting net, and part of that is the transitions.
	R	So
	P	*indistinct*
	R	It comes down here. I mean you could draw it here
	P	Yeah, but
	R	But I don't think it matters.... Yeah, I'd like to .... I don't actually like maps as object names [P: OK] So they're transformers also then, but they're transformers with state, and that state is the map.
16:08	P	Transformers rather than transformations? Not that there's any difference between the two terms *indistinct*
	R	Transformations are the result *indistinct*
	P	*indistinct*
	R	A parser is actually a transformer [P: yeah] but let's stick with parser because everybody knows what that is.
	P	What about a role map? That's a transformer as well, is it?
	M	A converter?
	R	Yeah, I think we're getting into semantics here in the worst sense of the word. [laughter]
	P	You think it's partly responsible for the element transform?
16:09	R	Well there may well be customisation generated stuff [P: yeah] which is a mapping between types - just a set of rules if you like [P: yeah] and then there's stuff which is mapping information which is part of its state, so there's a mixture of things going on there. I think its actually... the type mapping's one of the bits of one of the sorts of rules [P: yeah] that drive it along with other sorts of rules [P: mm hm], graph grammar rules.
	P	So really it's just a part of that transformation.
	R	I think so.
	P	Seems reasonable *indistinct*
	R	The sort of thing I propose is that we have five layers, cos we've got separation of concerns, textual separation of concerns [P: mm hm] We can then layer by layer populate with objects and discuss those objects identifying those which interact with other layers.
16:10	P	Yeah Not specifying the connections
	R	Not specifying the connections, we've got *indistinct* layer. We do that across the layers [P: mm hm] We could... then we need to show interaction. We could do with this, saying we're only showing here the objects which interact across the layers we're not showing other objects.[P: mm hm] Which may be sufficient in its own right, but it does miss out this notion of causal things, scenario things, so there's a need for sequence charts... which is exactly what we've got on the left hand side, bundled up together. We can separate them out and show them in that way, but it's effectively what we've done. We followed an event at one level through its causal
	P	We could have an example for each of those saying what
16:11	R	Yeah, should do construction... let's do that. We can discuss this notion of the limited nature of the mapping layers, transforming layers [P: mm hm] and see if that flies [P: mm hm] - let's just describe our ideas there. What we can then say is when IFAD and ??? ought to be describing their objects - their transforming objects on the whole [P: yeah] although they've got other things there. There are other rules associated with *indistinct* as well. They can describe it as being a thing driven by rules or whatever [P: mm hm] to their heart's content, but we won't actually show the rules explicitly in the object level diagram because what the rules are about is how we're constructing this thing, not how it interacts with
	P	Does that make the rules similar to the ones we've already talked about?
16:12	R	Good question. I mean, possibly yes. Dunno. Possibly yes. We should identify things as being rules driven, perhaps, and therefore subject to customisation. It's like if you were to draw object models of StP, we'd draw GEE and a set of rules files [P: mm hm] - depends what you're trying to do, why and for who you're drawing the model. Because you might have a DFE object and an SPE object and an STE object [P: yeah] and in some sense the fact that they happen to be based in the same core driven by rules is an implementation concern [P: yeah] Depends on what level you're talking.
	P	Yeah, I understand. Makes the diagrams simpler anyway. If necessary we could always show these rule driven things as part of the diagram - do it like that if we wanted to.

16:13 16:14	R	OK so I propose we get back together first thing - well not first thing, early on Monday [P: mm hm] to have the opportunity to say - wait a minute, when I woke up Sunday morning I felt an urge to write this down, it won't fly [laughter] but in the absence of that it's to allocate some responsibilities for documenting things, actually making proper documents [P: yeah] which will include agreement on notation of diagrams [P: yeah] and decisions on such things as are we showing rule files in this diagram or are we doing it some other way, if we're showing an object being customisable, how are we showing it being customisable and we can go on from there. Now I suspect I could be delegating most of that, I'm afraid. [P: mm hm] I mean you two are closely involved and are going to be doing a lot of it naturally anyway. There's still this D31 review re-review to do [P: mm hm] which I hope we could do in half a day and I'm meant to be presenting a T23 plan at the meeting, haven't even though about a T23 plan yet *indistinct* ... so I'll be involved, but I'm not sure at this stage how much of this first pass of writing and drawing I'm going to be doing [P: mm hm] I think we've got the effort, I think we can do it in the time - won't be polished.
	P	Presumably we just use the format of this document anyway to some extent.
	R	Yes, although we've got different sorts of diagrams in there. Right, so who wants to draw that right hand one out so that it gets a good - you know, we need a permanent record so if someone's got their hands on either X-fig or something equivalent or something on a PC or whatever - it's the piece of paper I'm interested in, not in anything we use in the document.
	M	OK I'll draw it.
	R	I'll copy that one down. I'm probably not going now cos I need to leave so I can catch a train.