The Designers' Notepad: Supporting and understanding cooperative design

Michael Twidale, Tom Rodden, Ian Sommerville
Department of Computing, Lancaster University, U.K.

ABSTRACT: We describe the development of a system to support cooperative software design. An iterative development approach has been used, based upon the observation of system use in authentic design sessions. This allows us to correct interface errors, and also to learn more about the nature of collaborative design. The observations of use and the resulting refinements of the system are described. In particular we note the variability in design activity both amongst designers and according to circumstances. We also note the way in which concepts mutate over time (often involving frequent and rapid revision) leading to an evolution of structure.

1. Introduction

Supporting the work of designers has been a major focus for the developers of cooperative systems and is illustrative of one of the fundamental problems of developing CSCW systems. While we recognise that most design involves collaboration, our understanding of its nature as a cooperative process is limited. Worse, our intuitions as to the support required may themselves be flawed (Grudin 88). Consequently, tool developers must discover the nature of the design process while simultaneously developing mechanisms to support and potentially improve it. We believe this interplay between the nature of an activity and the influence of the supporting tool to be a central feature of all CSCW systems development. Over the last two years we have been developing a system which provides support for system design. This paper describes the iterative approach to systems development based upon the observation of system use in realistic design sessions. The observations of use and the various features of the system which have emerged as

a result of these observations are also described. The development of co-operative systems is a problematic endeavour requiring a combination of skills often drawn from a range of different disciplines. By use of iterative development we can make use of the results of an ongoing ethnographic study of collaborative design (Button & King 92) as these results become available.

2. Supporting the work of designers

Design is an essential part of the systems development process and has consequently attracted considerable interest from the software development community. This has resulted in a range of different methods including JSD (Jackson 83) and OOD (Booch 91). These approaches prescribe a particular design model and require designers to adopt that perspective. The methods have been increasingly supported by the use of CASE tools (CASE 89). However, while CASE tools have provided techniques to support the enforcement of the approaches suggested by different methods, they have provided very little support for the creative process of design, as distinct from solution structuring, refinement and documentation.

The design process is often viewed by designers themselves as a creative and personal activity (Lawson 80). It involves the development and normalisation of concepts relating to the artefact being constructed. Designers tend to adopt flexible and personal notations to express these concepts. These take the form of diagrams, sketches and personal notes. These design notes form an integral part of the activity of design. However, they are viewed as personal resources and consequently are not intended to be interpreted by others.

The reality of modern design is that the work is shared both between different designers and across different phases of the development process. This communication and sharing of designs requires the adoption of some standard notation or formalism. To date these notations have been provided by different design methods. The methodological approach adopted by design methods and CASE tools have proven problematic, so that designers tend to design 'away from the tool' and use the tool to document designs after the event**.** Thus little tool support is actually provided for the creative design process.

A number of tensions exist in the design process which need to be addressed to allow support for design as a creative activity. Most notably two tensions which we intend to address are:

The private v communal nature of design
Although taking place within a cooperative setting many aspects of design are essentially a personal endeavour. The interplay between public and private design is a significant portion of design.

Freedom of expression v formalisation of shared understanding
Initial design concepts require a high degree of freedom in the notation used to express them. However, subsequent activities within the design and development process require a greater degree of formality to alleviate problems of misunderstanding across a community of designers and developers.

Our approach involves the development of techniques to directly address these tensions. We believe these tensions to be central not only to design but to many cooperative activities within real organisational settings. Consequently, the approach we have adopted to development is of interest to the CSCW community in general.

2.1 Previous approaches to design support

Within the domain of cooperative work, research in design support has generally followed one of two approaches. The first focuses on uncovering and recording the *rationale* used to arrive at different designs. The second concentrates on providing a *shared surface* for expressing designs and reflecting upon how these surfaces are used by designers. Our aim is to draw from the experiences of both these groups.
The intent of design rationale is documentation of the sequence of decisions made in realising a design. Some systems have adapted existing paper-based methods of externalising rationales. The most notable of these are gIBIS (Conklin 88) and rIBIS (Rein 91). Other representations have been proposed (Carroll & Kellogg 89, Fischer & Girgensohn 90, Lee 90 and MacLean et al. 89).

A number of problems have been noted with design rationale systems: they may have problems of acceptability (Yakemovik & Conklin 91) and difficulties with forcing users to decide at an early stage on the nature of each piece of information. This can be difficult to do, particularly where collaborators have to agree on the classification (Shipman & Marshall 92, Conklin & Bergman 88). Much of the motivation of current research is in the development of an appropriate theoretical base for the representation of design rationales and on the cognitive process of design. Little consideration has been given to the development of techniques and tools to support the creation of design rational within the early creative portion of the design process.

The research on *shared space* (for example Bly 90, Greenberg 91 and Ishii 91) adopts a contrasting approach to information capture. These systems provide a space upon which designers can collectively express and structure their ideas in a relatively unconstrained way. They have often focused on drawing as the primary means of expression and have studied how users interact with and through the shared surface. The Cognoter tool (Stefik et al. 87) is an example of this group. It provided facilities to allow ideas to be expressed, collected and commented upon by other users. However, later studies of its use revealed communication

breakdowns and suggested that these were a consequence of various assumptions in its design which did not relate to actual tool use (Tatar 91).

The NoteCards approach uses hypertext as a kind of shared surface. Although offering many advantages, classification problems similar to that for gIBIS have been observed (Shipman & Marshall 92, Monty 90). Users had difficulty chunking information into cards, naming cards and filing cards. Typed links were rarely used (and then, inconsistently). Link direction and link semantics also proved to be problematic. Many of these difficulties again spring from design assumptions which did not match actual systems usage.

3. Developing the DNP: an iterative approach

Our work builds upon the experiences of these previous approaches to design support and aims to tackle the difficulties that they unearthed. While we recognise the need to represent and record design information in a form which may ultimately be useful during later parts of the development process, we also acknowledge the problems of requiring designers to commit to a particular interpretation too early in the design process. Indeed, an overriding criterion for any system must be acceptability (Grudin 88). Therefore our main focus is on the development of an interface that users find easy to operate and which is appropriate for the early stages of design. We do not have a theory of design that we wish to impose or test on designers, but rather a desire to discover their requirements. This has led us to an approach to tool development based upon rapid prototyping.

We support the notion that idea representation tools such as Cognoter or more general design surfaces provide a medium of representation within the conversation surrounding the design process (Tatar et al. 91). Previous studies have examined the use of traditional whiteboards and shared drawing media in the design process (Suchman 88, Tang 89). However electronic drawing surfaces are different from these traditional media (Tatar et al. 91) and gaining a clear understanding of how these are exploited to express design concepts is a crucial part of our approach. Our central problem is that the medium of expression plays a central role in the representations. In fact, we would argue that the possibilities offered by electronic systems to represent designs and the nature of these representations are sufficiently intertwined that neither can be adequately addressed in isolation. Thus, not only is little known about the software design process, whether by one or many people, but the development of computer systems to support the process makes some features necessarily unknowable. This is because the computerised design tool is likely to change the nature of the design process as the word processor has changed the nature of writing (Haas 89). Therefore we have adopted an iterative approach to development based upon developing facilities in close cooperation with the designers using the system.

We started by providing an initial set of core facilities which allowed designs to be expressed. These facilities were then used by designers over a prolonged period to support a variety of real design tasks. This usage was observed and videotaped and the participants invited to comment on their experience with the system. The aim of the sessions was:-.

1. To isolate problems caused by features of the user interface.
2. To examine the use made of the system and to highlight additional functionality required by designers.
3. To assess the usefulness of existing functionality before adding to the system.
4. To provide information about the process of design itself.

The prototyping approach is shown diagrammatically in Figure 1.

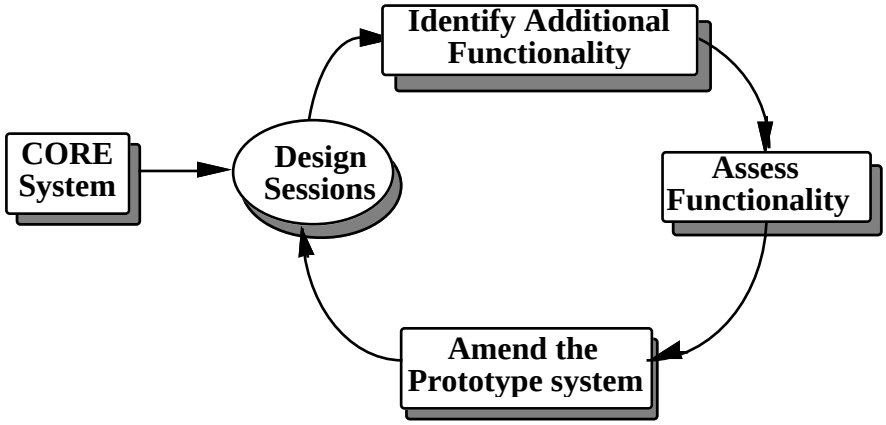

Figure 1. The development approach.

Our approach has links with participatory design (Bodker 91) and is similar to the approach proposed by (Tang 91). It also borrows from other domains including HCI (Hartson 91) and AI and Education (Twidale, in press). The latter domain has many similarities with CSCW in that both focus on the use made by people of computer systems and concern the support of human activities (learning and work respectively) that are imperfectly understood. In AI&ED it has been observed that a single flaw in an interface can have a substantial negative effect in learning outcomes, but also that a feature of the interface intended for one purpose can have additional beneficial purposes (Twidale 91). Thus features of the interface have the potential of swamping the effects of deeper, more interesting and sophisticated modules of a system.

An iterative approach to development can help to alleviate this problem. By testing basic versions of the system in simple circumstances, (such as with a single designer, two designers sitting at the same terminal or two designers with a terminal each sitting next to each other), we can eliminate the grosser interface errors. This can inform the design when it is extended to the necessarily more

complex cases (such as many designers separated in space and with limited bandwidth communication). For this approach to work, we need to focus more on the interface difficulties rather than the successes. We can be confident that the difficulties will scale up to the more complex cases if nothing were done about them, while the successes may not.

3.1 The Software Design Process

Rather than considering all aspects of design as an abstract process, we are primarily concerned with the software development process. This allows us to make particular assumptions about the design setting within which any development systems will be applied. In particular, the software design process is characterised by:

The development of structure

Software design is closely concerned with the transition from partially-defined, loosely structured ideas and concepts to fully defined and structured design descriptions. These descriptions form the basic plans for subsequent development. The majority of these design descriptions are specified diagrammatically using some form of network diagram. These diagrams show a number of software entities which are linked to describe corresponding relationships.

The work of groups

Any design of a significant size is the endeavour of a number of designers. A team of software designers will work together in a variety of different ways over a substantial time period to realise a completed design. The patterns of work involved and the different forms of cooperation which take place are only partially understood by the participants in the design process.

This view of the design process forms the context for our construction of the DNP. It is important to note that software design forms only the initial phase of the overall development process and some benefits of supporting design activities may only be realised much later in the software lifecycle. To reap these benefits though, the design tool must be *used* by software designers. Meeting the needs of designers by meshing with their work practice becomes an issue of paramount importance.

3.2 Design Sessions

When investigating a domain so poorly understood as collaborative software design it is vital to be in contact with designers as early as possible. A tool that designers find unusable or even just awkward to use will not be adopted.

Given that our development of a system to support software design is necessarily exploratory, it is inevitable that we will make mistakes. The aim of our approach is to increase the speed and ease of identifying and correcting these

mistakes. Considerable attention is given to mistakes relating to the interface. It is clear that the interface to any system has a substantial effect on its use. If care is not taken over the interface, problems of use may not be due to some feature of the support for CSCW that is counterproductive or missing but merely to an interface problem. It is possible to identify many of these issues when the system is being used by a single user, even though it is intended for collaborative work: difficulties that a single user has are more than likely to be also experienced with multiple users.

It is for this reason of simplicity that we chose to start by concentrating on a system for multiple simultaneous use of a single workstation by two or three people. A distributed version for two workstations has been developed and tested, but the bulk of the work has been on the single version. In the case of the distributed version we had the workstations side by side. We chose this in order to maximise the potential communication bandwidth between the participants. In this way any problems observed could not be ascribed to bandwidth limitations and so would scale up to true distributed usage if not remedied. The approach is intended to be complementary to ongoing research addressing the challenging technological and social implications of the more complex forms of collaboration such as synchronous working over long distances (Clark & Scrivener 92).

3.3 The Mini-DNP

Our starting point for providing support for designers was the development of a minimal system with a set of simple core facilities called the mini-DNP. This system was influenced by a previous system to support design by a single user (Sommerville et al. 1990) and the experiences gained from developing and using that system. The essence of mini-DNP is a means of creating entities and links between them (Figure 2).

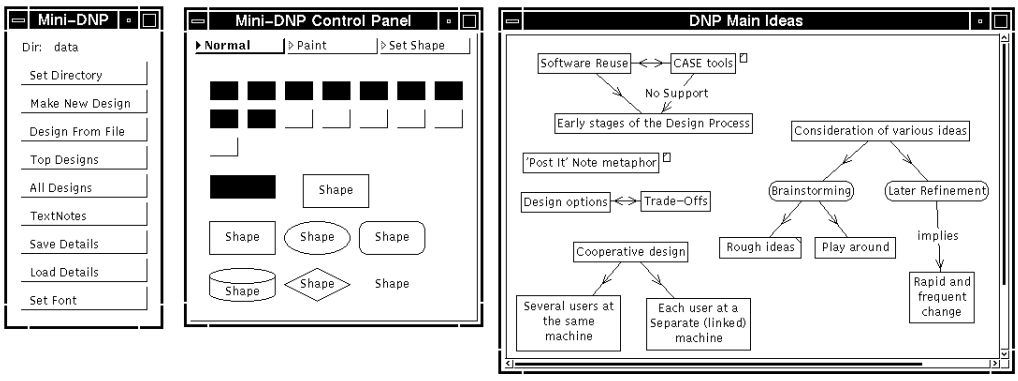

Figure 2. The minimal DNP interface.

The user creates an entity by typing in a design window. He or she may then move it with the mouse. Linking is done by selecting one entity and then clicking on the entity one wants to link to with the shift key down. This linking of entities

forms the diagrams ultimately needed to specify design descriptions. We also provided textnotes which are based on the Post-It Note metaphor (see figure 5) and allow designers to attach one or more notes to an entity. These can be used by design teams for more textual comments, ideas, opinions, code fragments, references etc. A variety of textnote types are provided and users may define their own (including form structures). An entity with textnotes has an icon attached (eg. the entity 'CASE tools' in figure 2) and the notes can be examined by clicking on the icon. Designs can be saved and loaded from a file and a paper report may be created containing a screendump of the design and a list of the entities and their textnotes. Each entity may itself be expanded to become a subdesign. A new window is opened and entities and links created in the normal way. Subdesigns may contain entities that are themselves subdesigns. A loose form of typing for entities and links is provided using colour, shape and labels. The user controls the degree to which s/he wants to use typing. The type of an entity or link can easily be changed at any time.

The initial system has certain characteristics which make it suitable for supporting the design process:

1. A fast, easy-to-use interface which supports the creation of directed graphs. These may be created at any number of levels with simple navigation from one level to another.
2. Untyped entities and relations. This is critically important in the early stages of an analysis where it is unrealistic to fit an entity into a type hierarchy.
3. Extensive annotation facilities which allow system entities and relations to be annotated with structured and unstructured text.
4. Post-creation type attribution. These facilities also allow the type of an entity to be easily changed.
5. Report generation facilities.

3.4 Assessment

We have used a variety of users to assess the system. Given that the aim is continual refinement, one can initially use members of the development team to test the system. At first sight it may seem a problem that they will be more motivated than most to use the system and consider it in a favourable light. Nevertheless, provided that they have genuine tasks (such as the ongoing design of the system itself) they are likely to discover some of the grosser interface errors. We can be confident that such errors will scale out: a feature that a developer finds hard to use will certainly be difficult for an outsider (and of course if the developers don't like using the tool on a regular basis it is unlikely anyone will). Later we tried the system on users not involved in its development in order to discover any hidden assumptions in the developer community about ease of use and needed features.

The system has been simultaneously subject to use and under development for approximately two years. During this time a substantial community of users have exploited the system. Table 1 summarises the variety of system use.

By designer:	Single designer
	Two designers at one workstation
	Three designers at one workstation
	Two designers at two adjacent workstations
By task:	Software design
	Information organisation and browsing
	Project management
	Lecture course design
	Designing papers (including this one)
	Designing talks
By user:	System developer
	Project members
	PhD students
	Sociology colleagues
	Visiting academics
	Undergraduates

Table 1. Nature of usage of the system in tests.

In all cases of use, whether by individuals or groups, we asked our volunteers to bring along a task that they had to undertake anyway. We believe it is important to develop the system on real-world tasks as these have features that are very hard to replicate in contrived tasks. These features include ambiguity, open-endedness, history and engagement of the user, who ideally will be focussing more on the problem than the tool. Our instructions to the user were to try and use the tool, asking where necessary how to achieve anything they want to do. Our assumption is that the tool will be usable up to some point when it becomes frustrating because it prevents the user doing something s/he wants. We can identify what s/he wants to do and then can assess what sort of functionality should be added to the system.

Given the desire to use authentic tasks we are somewhat at the mercy of the needs of our volunteers. Fortunately a tool intended to support the very early stages of software development is equally useful for many other creative design tasks such as project management, making ethnographic notes and the design of papers and reports (Sommerville et al. 93). Indeed the initial structure and elements of this paper were collaboratively developed using DNP.

Some of the system use has been in one-off design sessions, others have continued over a number of sessions and yet others (particularly those involving project management and the design and management of research activities) are ongoing over many months. The system is now accessible from an number of different locations on campus. In particular, two users have access to the DNP

from their office and use it on a day to day basis. These latter forms of use are beginning to reveal the nature of the requirements for management of the complexity of designs over time.

4. Design Support in the DNP

The intent of the design sessions was to discover the various patterns of use of the DNP (including difficulties) for different designers. Our observations of use informed the development of appropriate support facilities. This section briefly highlights the results of this process.

4.1 Variability of use

Our studies have confirmed the great variability in design activities both between users and by the same users over time and circumstance (see figures 3, 4, 5 & 6). For example, some designers use very terse entity names with textnotes to contain details, whereas others use phrase- or sentence-like names. Some designers use many links to indicate connectedness whilest others use two-dimensional proximity (Marshall & Rogers 92). Some use a great deal of colour and many different shapes, whilst others use black rectangles all the time. As examples of variability over time, we have observed cycles involving bouts of entity creation and rough positioning. These involve very rapid and intense activity where the minimum of options are employed. After such a bout there is a recovery period where the display is 'tidied up'. Links are created and the entities rearranged to convey additional meanings by their proximity to other entities as well as to reduce the clutter of areas of great activity.

This variability confirms the need for flexibility and ease of revision in the DNP. Much of this flexibility comes from our decision to avoid associating semantics with entities, links and subdesigns and to allow these to be used primarily as a means of expression by users. The meaning of these initial design graphs are left to the interpretation of different designers. The intense 'bursts' of activity surrounding the generation of entities caused us to focus on supporting the rapid entry of entities (just typing and hitting return between entities generates entries positioned in a list format). Once entered these entities can be tidied by altering their position, colour and shape. For example, contrast the design in figure 3 showing the early stages of the design with a later stage shown in figure 5.

4.2 The Evolution of structure

The elements of a design that are added to the Notepad in the form of entities are necessarily ambiguous; they consist of a label of a few words referring to a concept. However, not only is that label capable of misinterpretation by anyone

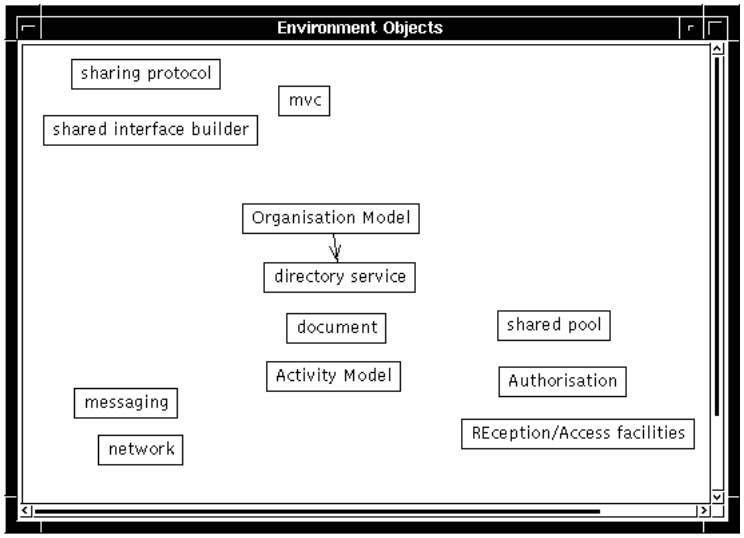

Figure 3. An initial design in the DNP.

other than its creator, but even for the creator it seems that deliberately ambiguous terms are chosen (for example, entities were often called "object" or "user"). This is an instance of postponing decisions about details in order to deal with overview concepts. Later on in the design process, the meaning of the initial concepts will be refined. This can involve qualifying the entity by editing its name (including completely changing the name), creating more entities for the constituent concepts in an ambiguous description, adding more information by attaching textnotes, creating a subdesign for the entity, or completely replacing the entity.

It would seem that designers need to have a certain degree of ambiguity during design so as to not get too held down by details. Gradually these ambiguities are addressed and refined. This has been observed in other situations (Marshall & Rogers 92) and reasons proposed for why users wish to avoid formalisation (Shipman & Marshall 92). In a similar manner, the meaning of a link can change over time. Initially its meaning may be 'these entities are in some way connected'. Eventually this is refined into a more precise meaning. The gradual evolution of precision is often associated with the usage of link typing; links with a similar meaning are now given the same colour or label.

The attributes of entities and links are used within the DNP as a mechanism to support the evolution of structure. The use of colour and shape as a means of typing both entities and links is directly supported by the DNP. Once created these different attributes of entities and links can be modified by users over time from the control panel shown in figure 2. This freedom allows appropriate structure to emerge within the design after the entities have been created and allows the type of entities to change after the creation of the entity. Figures 3 and 5 illustrate this migration. This is in contrast to previous systems which supported the definition of a set of types but required the designer to select the type for each entity as it is created.

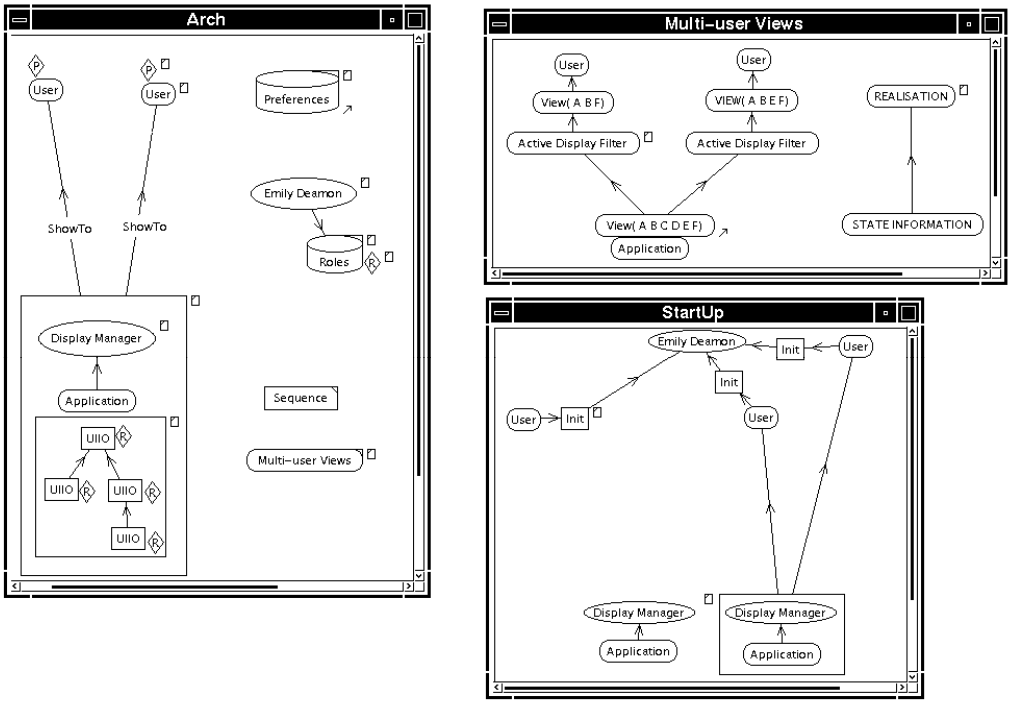

Figure 4. The arrangement, framing and grouping of entities.
The two designs on the right are subdesigns of the one on the left.
11 colours were used in this design

The relationship of entities within any non trivial design is extremely complex. In addition to links, space is often used to distinguish this relationship as appropriate entities are clustered (Marshall & Rogers 92). Users expressed a need to abstract from these groupings once they were made. Subdesigns were used to facilitate this. The user can frame entities to form a group. S/he may then 'push down' the group into a subdesign to be replaced by a single entity (figure 4).

4.3 Cooperative and Single use

The most visible difference between cooperative and single use is speed; single designers enter, refine and revise items far more quickly than do groups. This is mainly due to the need of group participants to justify and explain their actions to others. No action is completely unambiguous and the degree to which features can be left implicit by a single user are naturally far less than when ideas have to be shared. Not only is greater explanation and elaboration required, but actions such as revisions and additions have to be negotiated, leading to a debate about the appropriateness of the activity. By contrast, single designers create entities and links and rearrange them with at times almost bewildering speed. We take this as evidence for the need for the tool to support frequent and rapid revision of designs provided by the DNP.

The sharing of a keyboard and mouse on the whole did not appear to cause great difficulties, with users taking turns to enter text or rearrange entities. Control of the keyboard can involve either a position of power, temporarily controlling the collaboration, or it can involve a secretarial function, minuting the deliberations of the others. In the distributed version of the DNP, despite the close proximity of the machines involved, users exploited the shared view of the design to quickly partition the design activity and work independently upon different portions of the overall design. Substantial use was made of textnotes to annotate features of the other user's design (figure 5).

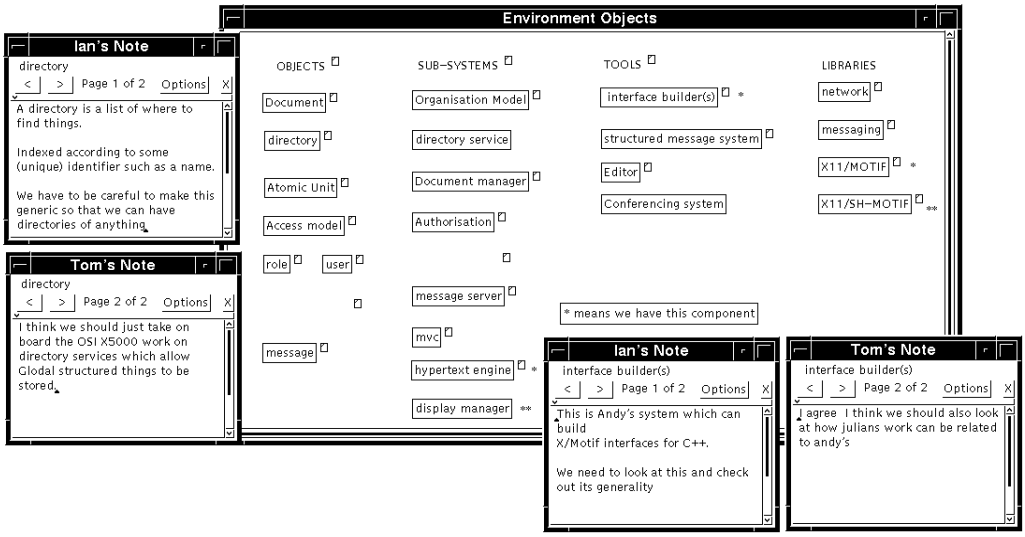

Figure 5. The use of textnotes to comment on a design.
7 colours were used in this design.

An interesting feature of cooperative use in both settings was how entities were moved and highlighted for emphasis, as part of a debate. The user currently controlling the mouse would select an entity s/he wished to discuss, causing it to be highlighted. S/he might move this entity slightly (from side to side or in small circles) as a means of emphasis. This activity of moving entities for emphasis can be contrasted with occasions when the move was more substantial, indicating some semantic feature, by positioning an entity closer to another entity. This could be permanent, or merely a continuous movement as a suggestion. Besides the use of the mouse, more direct interaction was observed, namely users pointing at the screen with their fingers. The cooperative situation also saw another use of the DNP's ability to support the evolution of meaning. Often participants in a group design activity would use an entity as an argument placeholder and use this accepted design entity to refer back to a previous debate.

As a result of our observation of groups, the facilities associated with the placement of textnotes were extended to allow users to easily exploit textnotes as a method of annotation. These extensions allow facilities for the definition of new note types which can be associated with different users and/or purposes. Figure 6 shows the tool to create a new pad of notes with a user-defined structure, and

another created notepad. Facilities are provided to filter and highlight the presence of notes from different designers.

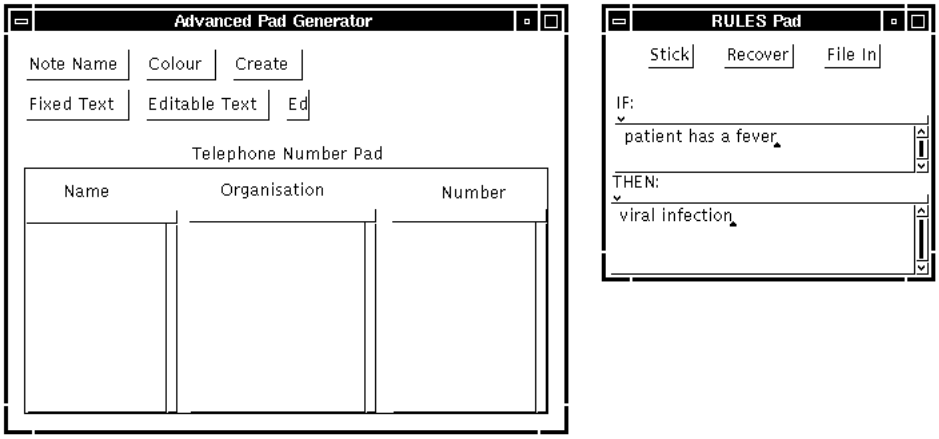

Figure 6. The generator for creating a structured note and a user defined notepad.

5. Conclusions

An iterative development approach is appropriate for CSCW systems development because it facilitates our need to learn more about the nature of cooperative activity while simultaneously developing the system. The testing of the early versions of the Designers' Notepad have emphasised the variability in design activity both amongst designers and according to circumstances. We also noted the way in which concepts mutate over time leading to an evolution of structure. This can lead to frequent and rapid revision. Ease of revision is important in encouraging the brainstorming of concepts that are necessarily incomplete. Systems to support the early stages of design need to support these features of the process. By means of an iterative development we are able to correct the grosser interface errors that have the potential to swamp the effect of deeper issues concerning design and cooperative work in general. The methodology also enables us to profit by the stream of results that an ethnographic study produces. In its current state the Designers' Notepad is as much a tool for acquiring information about design activity as it is a tool for supporting that activity.

6. Acknowledgements

Michael Twidale is a Science and Engineering Research Council Junior Research Fellow. The development of the Designers' Notepad was funded partialy by the Joint Council Initiative in Cognitive Science and COMIC (Esprit basic research project 6225). The authors would like to thank their colleagues in the Sociology

department; John Hughes and Val King for their support and advice throughout this project.

7. References

Bly, S.A. & S.L. Minneman, S.L. (1990). Commune: A shared drawing surface. *Proceedings of the Conference on Office Information Systems,* Boston, April 25-27. pp 184-192.

Bodker, S., & Gronbaek, K. (1991). Cooperative prototyping: Users and designers in mutual activity. *International Journal of Man Machine Studies,* 34(3) 453-478.

Booch, G. (1991). *Object Oriented Design with applications.* Menlo Park CA: Benjamin Cummings.

Button, G. & King, V. (1992). Hanging around is not the point: calling ethnography to account. Paper presented at the Workshop on Ethnography and CSCW system design, CSCW '92, Toronto.

Carroll, J.M., & Kellogg, W.A. (1989). Artifact as Theory-Nexus: Hermeneutics Meet Theory-Based Design. *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems.* pp 7-14.

CASE (1989). The CASE experience, *Byte,* April 1989, pp 206-246.

Conklin, J. (1988). gIBIS: A hypertext tool for exploratory policy discussion. *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '88),* Portland, Oregon. ACM Press, pp 140-152.

Fischer, G., & Girgensohn, A. (1990). End-User Modifiability in Design Environments. *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems.* pp 183-191.

Greenberg, S., and R. Bohnet, R. (1991). GroupSketch: A multi-user sketchpad for geographically-distributed small groups. *Proceedings of Graphics Interface '91,* Calgary, Alberta, June 5-7.

Grudin, J. (1988). Why CSCW applications fail: problems in the design and evaluation of organizational interfaces. In *Proceedings of the Conference on Computer Support Cooperative Work (CSCW '88),* (pp 85-93). Portland, Oregon: ACM Press.

Haas, C. (1989). How the writing medium shapes the writing process: effects of word processing on planning. *Research in the Teaching of English,* 23(2) 181-207.

Hartson, H.R., & E.C. Smith, E.C. (1991). Rapid Prototyping in Human-Computer Interface Development. *Interacting with Computers,* 3 (1) 51-91.

Ishii, H., & Arita, K. (1991). ClearFace: Translucent multiuser interface for TeamWorkstation, *Research report* NTT Human Interface Laboratories. January.

Jackson M.A. (1983). *System Development.* Prentice-Hall, New Jersey.

Lawson, B. (1980). *How designers think.* Chatham: W & J Mackay.

Lee, J. (1990). SIBYL: A tool for sharing knowledge in group decision making. *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90),* Los Angeles, California. ACM Press, pp 79-92.

Marshall, C.C. & Rogers, R.A. (1992). Two years before the mist: experiences with Aquanet. Proceedings *ECHT '92*. Milan.

MacLean, A., Young, R.M., & Moran, T.P. (1989). Design Rationale: The Argument Behind the Artifact. *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems.* Issues in Interface Design Methods. pp 247-252.

Monty, M. L. (1990). *Issues for supporting notetaking and note using in the computer environment*. Unpublished Dissertation, Department of Psychology, University of California, San Diego.

Rein, G.L., & Ellis, C.A. (1991). rIBIS: A real-time group hypertext system. *International Journal of Man Machine Studies* 34 (3) 349-368.

Shipman, F. M. & Marshall, C.C. (1992). Formality considered harmful: experiences, emerging themes and directions. Submitted to *InterCHI '93*.

Sommerville, I., Haddley, N., Mariani, J.A. & Thomson, R. (1990). The designer's notepad - a hypertext system tailored for design. In McAleese, R. & Green, C. (Eds.), *Hypertext: state of the art* (pp 260-266). Oxford: Intellect.

Sommerville, I., Rodden, T., Sawyer, P., Bentley, R. & Twidale, M. B. (1993). Integrating ethnography into the requirements engineering process. *Proceedings, 1st International Conference on Requirements Engineering*, San Diego, January 1993, IEEE Press.

Stefik, M., Bobrow, D. G., Foster, G., Lanning, S., & Tatar, D. (1987). WYSIWIS revised: Early experiences with multiuser interfaces. *ACM Transactions on Office Information Systems*, 5(2) 147-167.

Suchman, L. (1988). Representing practice in cognitive science. *Human Studies*, 11, 305-325.

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.

Tatar, D.G., Foster, G. & Bobrow, D.G. (1991). Design for conversation: Lessons from Cognoter. *International Journal of Man Machine Studies* 34 (2) 185-210.

Twidale, M. B. (1992). Student activity in an Intelligent Learning Environment, *Intelligent Tutoring Media*, 2(3/4) 113-127.

Twidale, M. B. (in press). Redressing the balance: the advantages of informal evaluation techniques for Intelligent Learning Environments. *Journal of Artificial Intelligence In Education.*

Yakemovic, K.C.B. & Conklin, E.J. (1990). Report on a development project use of an issue-based information system. *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW '90)*, Los Angeles, California. ACM Press.