UNIVERSITY
of
GLASGOW

Department of
Computing Science

Encouraging the Effective Use of Contextual Information in Design

Steven Clarke

A Thesis submitted for the degree of Doctor of Philosophy

to

The University of Glasgow

June 1997

Acknowledgements

Without Gilbert Cockton's excellent supervision and encouragement this thesis would not have been possible. He gave me the confidence to believe in my work and the insight to know when things weren't quite working out. It was hard work, but enjoyable.

Phil Gray also gave valuable input and was always a useful source of ideas. In particular, I have Phil to thank for helping me with the evaluations and with several reviews of the thesis. Chris Johnson also helped out, especially during several of my crises. Keith van Rijsbergen asked some interesting questions during my first and second year vivas, and these were worked on in the thesis.

I also have to thank Tony Salvador and Michael Mateas of Intel Corporation, without whom my thesis would probably have been at least two chapters shorter but would have neglected several crucial points. They gave me a fascinating project to work on at Intel and were great to work with. Thanks for making clear some muddled thoughts I didn't even realise I had.

I'd also like to thank Darryn Lavery, Fraser Hamilton and Kevin Kerr who have had to put up with my troubles and moans over the years. They all provided excellent comments on my ideas and encouragement. Thanks also to all the G091 residents and all the other research students in the department who've helped make working here so enjoyable.

Thanks to all my friends and family for putting up with me and even being interested in what I was doing. I know it can't have been easy.

Finally, there is one person who doesn't want to be mentioned but deserves to be. She's supported me spiritually and mentally (and yes, even financially at times I must admit!) and has done a fantastic job. I hope you think that it has all been worth it. Now I can get a real job and start paying you back!

Contents

Table of Figures

List of Tables

1. Chapter 1 Introduction

1.1 Introduction

There are many examples to be found in computer science literature of interactive systems that have been designed without taking adequate account of their context of use. Influential contextual factors such as environmental conditions, task duration and operator responsibility have been ignored in design projects involving nuclear power control rooms, missile detection systems and aircraft maintenance systems sometimes with disastrous consequences (Bignell and Fortune, 1984; Friedman and Kahn, 1994). With hindsight, it seems surprising that such factors were overlooked, yet as Seely–Brown and Duguid (1995) point out, it is easy to overlook contextual factors in the design of an artefact, since by their very nature, contextual factors are not a canonical part of the artefact being designed but are rather, 'peripheral' to the artefact. Hence, designers often neglect to take account of contextual factors due to their focus on the artefact itself.

Such a focus could perhaps explain one of the largest accidents to occur involving a nuclear power reactor. While focusing on the power plant control room itself, the designers may have neglected to consider the different operating conditions that could potentially exist within the control room. The accident at the Three Mile Island nuclear power station in 1979 could in part be attributed to the poor design of the layout of the control room and of the various control switches and displays (Bignell and Fortune, 1984). From reports of the accident, it would appear that little attention was paid to the different conditions that could exist in the control room (for example, in an emergency) and the effect that these conditions would have on the ability of the operators to utilise the appropriate switches and displays in the control room to adequately control the power station.

1.1.1 The Three Mile Island Accident

The Three Mile Island accident began when maintenance staff were renewing the resin that was used to treat the water that circulated the nuclear reactor. Some water escaped into an air circuit, causing a number of valves to shut off. These valves controlled the flow of water around the nuclear reactor, water which transferred heat energy from the reactor to the steam turbine and the electric power generator it drove. As a result, the heat energy could not be transferred from the reactor to the turbine. Hence the water began to rise in temperature and pressure. Safety mechanisms were in place for such an event, and they automatically came into action. A safety valve was opened which allowed the steam and water to flow out into a drainage tank and control rods were dropped into the reactor to slow down the production of heat energy.

Unfortunately, the safety valve failed to close when the temperature and pressure had been brought back to normal levels. Instead, it remained open and continued to release water. However, this equipment failure, the only equipment failure that occurred throughout the accident, was not sensed for two and a half hours due to the poor feedback provided about the state of the valve in the control room. Instead of showing the state of the valve, the indicator lamp in the control room indicated the state of the solenoid that controlled the valve. While the solenoid had functioned, the valve had not. The operators in the control room took the state of the lamp to indicate the state of the valve. Hence they believed that the valve had closed, when in fact it remained open.

Emergency feedwater pipes were used to provide the reactor with water in the case where the main feedwater pipe had to be shut down, which had happened in this case. However, valves on these pipes were closed, meaning that water could not reach the reactor. Two lights in the control room indicated this, but these lights were not noticed by any of the operators. One of the lights had been obscured by a maintenance tag. Hence, due to the poor feedback provided by the control panel (in displaying the state of the safety valve and the emergency feedwater valves), operators believed that water was flowing into the reactor when in actual fact, water was flowing out. This mistaken belief led to confusion amongst the operators when other automatic safety features were triggered. A further two emergency valves opened to allow water into the reactor. However, the operators believed that there was already enough water in the reactor and so acted to reduce the water level. At this point, confusion reigned. At least a hundred alarms were operating in the control room and the operators were frantically trying to understand what was happening and to react to the situation. The confusion lasted a number of hours, with operators taking the only action they felt appropriate, given the feedback from the control panel and the situation they were in. At one point, a meltdown of the reactor was likely but fortunately, this was avoided.

Reports into the nature of the accident described the poor feedback provided to operators in the control room and the poor layout of both indicators and control switches. For example, controls were placed unreasonable distances away from their associated indicators. Many important indicators could not be seen from the normal working position of the operators. Some controls were unnecessarily large while others were too small in relation to their importance. Some controls were operated by turning them in a clockwise manner. Others were controlled by turning them anti–clockwise. Many items in frequent use were out of reach. As operators stretched over the control panel to reach such items, the likelihood of them inadvertently switching some other control was increased. Colour was used haphazardly throughout the control panel. For example, the colour red was used to signify fourteen different states. Further examples of poor design in the layout and operation of the control panel are given in summaries of the accident reports (Bignell and Fortune, 1984). It is clear that the designers of the control room did not pay sufficent attention to the way that the operators worked, especially in the different conditions that could potentially exist in the room (e.g., an emergency). Had the designers paid attention to the context of use of the control room, it is unlikely that the control room would have been so poorly designed.

1.1.2 Further examples of a lack of attention to context

More recent examples of poor design resulting from a lack of consideration of the context of use of a system also exist. Friedman and Kahn (1994) cite the US General Accounting Office report (US General Accounting Office, 1992) which discusses the failure of the Patriot system to intercept Scud missile attacks. In the 1990 Gulf War, Patriot Missiles were deployed to intercept incoming Scud missiles and destroy them before they could reach their target. However, the designers of the software that predicted the Scud's flight path had used the system's internal clock. Due to the way that the Patriot system worked, the longer the system was left running, the less precise the calculations of the Scud flight path became. Hence Patriot missiles often failed to intercept the Scud missiles, since the Patriot system was left running for long periods of time (at least as long as army commanders felt that they were in danger of Scud missile attack). Other factors, as well as the imprecise calculations also contributed to the failure of the Patriot system. However, it is clear that in this example, lack of consideration of the context of use contributed to the loss of life in the Gulf War.

Friedman and Kahn (1994) also report the efforts of SAS, a Swedish airline, to introduce an expert system in the maintenance process of their planes. However, the airline failed to take into account the effect such a system would have on the roles and responsibilities of those who would use the expert system. The mechanics, who had previously been responsible for the maintenance and upkeep of the planes, now became the operators of the expert system. The responsibility for the maintenance of the planes was transferred to the computer system. Hence the mechanics were not as concerned about maintaining the planes, since any mistakes would be blamed on the computer and not on them. As a result, repair quality declined, rather than increased, as a result of introducing the computer system. Other examples of designs that ignored the context of use can be found in Landauer (1995), Kling (1996), Sachs (1995) and Orlikowski (1992).

However, even if designers are aware of the importance of using contextual information in design, success is not always guaranteed. Cockton *et al.* . (1996) relate the experiences of

Powrie and Siemieniuch (1990), an interim report from the CAR project. The CAR project was an investigation of applications for integrated broadband communications in the European automobile industry. Powrie and Siemieniuch focused on co–operative design using multi–media communications and collected large volumes of contextual data to investigate how the introduction of broadband communications technology would affect the tasks performed by automobile designers. There was clearly no problem in collecting the data. The problems arose in using the data systematically. Seven hundred user requirements were generated from the contextual data, two of which were

- interaction with each graphical application should be as consistent as possible;
- users should have access to facilities that let them express and explore their ideas formally and informally using graphics.

Such requirements could apply to many different domains. The contextual information that was collected appears to have had very little influence on these requirements. Powrie and Siemieniuch suggest that much of the contextual data they collected was extraneous, but from the two requirements listed above, it can be seen that some of the information was not extraneous, it just wasn't used. Contextual information could have been used to generate requirements that were more specific to the context (i.e., the automobile design industry). Instead, much of the contextual information was simply unused, and general, non context specific requirements were produced.

1.1.3 Why do these problems occur?

Cockton *et al.* (1996) suggest that the problem with using contextual information lies not in using contextual information to derive user requirements but in using it fully and effectively.

> "The problem can be reduced simply to understanding the relationships between human contexts and systems designs. To understand these relationships, we require descriptions of human contexts and systems designs, and ways of linking between these descriptions. We need to prepare these descriptions without prior commitment to any prescribed methodology for moving from context to design (or vice-versa). The problem thus changes from that addressed in the CAR and similar projects (i.e., transforming context data into requirements), to one of finding a more systematic way of underpinning design decisions with contextual knowledge." (Cockton *et al.*, 1996).

Furthermore, this thesis will argue that in order to understand the relationships between descriptions of context and design and to link between these descriptions, any framework for using contextual information in design must take into account the following five factors:

- **Linking between representations**. As identified by Cockton *et al.*, (1996) and Clarke (1996), making the relationships between descriptions of context and design explicit will allow accurate judgements about the use of contextual information in design to be made;

- **Scope**. When using contextual information in design, the model or framework must ensure that the correct range of contextual information is both identified and collected. When one model uses a different scope than another model, the design is biased differently towards a different set of contextual information;
- **Definition.** Different definitions of context place different biases on the way that that context is used (Cockton *et al.*, 1995). Designers should be aware of the definition used by any particular framework and the bias that that definition introduces. Ideally, a framework should allow multiple definitions of context;
- **Quality**. Designers must ensure that they are collecting quality data, that is, data that truly reflects the reality of the situation being investigated. Designers must be able to determine the quality of the data they are collecting;
- **Understanding.** Designers must understand the contextual data. This can be difficult if one part of the design team have collected the data, and another part of the team have to use the data. Consensus must be reached amongst the team about the significant factors that can be elicited from the data, with respect to the design project;

These five factors are demonstrated and justified in Chapter 2 of the thesis, which reviews the current literature describing methods for using contextual information in design. To put the factors, and the thesis, into context, the next section discusses related issues and defines the scope of the thesis.

1.2 Related Issues and Thesis Scope

Having stressed the importance of using contextual information in design, it should be made clear that paying attention to contextual information is by no means the only way to attempt to design a successful computer product. Three other approaches are described below.

1.2.1 Psychological models

There are numerous psychological models which claim to model users and their abilities. One such model is the GOMS (goals, operations, methods and selection rules) model (Card *et al.*, 1983). The GOMS model models the procedural or 'how–to–do–it' knowledge required by a user in order that they may succesfully achieve some predetermined goals. A typical GOMS model shows the goals that a user is aiming to achieve and the methods by which these goals can be achieved. The operations or steps required to carry out each method are described. Whenever a user has a choice between different methods to achieve the same goal, selection rules are described which will select the appropriate method. Operations and methods are described at varying levels of detail, depending on the purpose of the analysis. They can describe fairly high level operations such as reading a text or they can described lower level operations, such as retrieving an item from the user's Long Term Memory[1]. GOMS models can be used to analyse how effectively the user maps their knowledge of what they have to do to the actions or operations that are required in order for them to achieve their goals.

Other models exist that focus on different aspects of human behaviour. For example, TAG (Payne and Green, 1989) models the learnability of a system from the user's point of view. In general, it is claimed that psychological models can be used to predict typical errors that users will make and areas where users will have difficulties in using a system. Knowledge of these errors and difficulties may allow designers to modify a system to remove those aspects that make the errors and difficulties likely.

1.2.2 Formal models

Formal models are used to model computer systems and their users mathematically, so that proofs of correctness (of the computer system) can be made. These mathematical models can also be used to identify likely areas of difficulty and typical errors that may occur. Formal models can be used to identify any inconsistencies within the requirements of a system (see Dix *et al.*, 1993 p. 301 for an example). They can also be used to verify that a system satisfies its requirements. The preciseness with which these formal models can model systems makes them highly suitable for modelling safety critical systems such as nuclear power plants and emergency services control centres. In these cases, it is vital to know that the systems will behave exactly as they should. Mathematical models allow us to determine with a high degree of accuracy how well systems will perform.

One of the most widely used formal modelling notations is Z (see Spivey, 1988). Z is based on the use of mathematical sets and functions. Hence standard mathematical set manipulation techniques can be used to reason about and prove certain properties of a system defined using the Z notation (and similar notations based on the use of sets and functions). The simplest sets in Z correspond to types in programming languages like *reals, integers* or *natural numbers.* This allows designers using the Z notation to describe a system as it would be implemented but without any bias from the particular implementation platform interfering with the description. Other formal modelling techniques can be used to model different features of a system. For example, CSP (Hoare, 1978) can be used in dialogue design.

[1] Long Term Memory is generally agreed to be the area of human memory in which facts, dates, names, places and all kinds of information are stored and remembered over long periods of time (days, months or even years). See (Baddeley, 1990) for a full discussion of the make up of human memory.

1.2.3 Guidelines

There are a number of publicly available guidelines which set out to offer guidance as to the design of computer based products. For example, Apple's Human Interface Guidelines (Apple Computer, 1992) specify what menus should be available in a computer system and how windows should behave. Other guidelines offer similar advice but for different platforms. For example, the Microsoft Interface guidelines (Microsoft, 1992) offer guidance for designers of Windows based products. On a more general note, Smith and Mosier have produced one of the most well known set of guidelines (Smith and Mosier, 1986), which apply across all platforms and concern general issues such as '*adopting a consistent organisation for the location of various display features*'. Many other sets of guidelines exist, some more specific than others. In each case however, guidelines offer some general guidance that the designer is recommended to follow.

1.2.4 Problems with Psychological Models, Formal Models and Guidelines

Each of the three different approaches described above has a number of drawbacks. The validity of psychological models is a debatable point. With so many models available, how are we to know which one to believe? Mathematical models on the other hand are difficult to apply and understand. According to Preece *et al.*, (1994, p. 491) "Guidelines are limited in their use." Care must be taken to interpret guidelines appropriately and to apply them to the appropriate situation.

What they all have in common however is that they do not adequately model or take account of contextual information. Psychological models focus primarily on a single user and their cognitive abilities as they use an interactive system. Contextual factors such as who the user works with, where they work etc., are beyond the scope of psychological models of users. Since there are so many contextual factors that can influence the way that a user uses an interactive system and due to the nature of these factors (often imprecisely and partially specified), it can be difficult to thoroughly model them in a mathematical model. Attempting to model context in this way could swamp the model and get in the way of the design. Lastly, general guidelines such as "Know the user" and "Maintain consistency throughout the user interface" can guide the design of a system but they cannot be used to make specific judgements about a particular system due to their generality. Since contextual information is specific to particular systems, it is unlikely that guidelines would be useful in attempting to make effective use of context in design.

Hence, this thesis is concerned with deriving designs that are useful and that have been influenced by relevant contextual information. As described above, the thesis argues that in order to produce really useful designs, designers must understand fully the context in which the system produced from their design will operate. They must know which aspects of context influence the work that the system will support, so that the system is designed such that it takes these factors into account when it provides the work support. The system must work in tandem, rather than in opposition, to these influential contextual factors. A system that does not take into account the many contextual factors that affect users' work will either not be used or will conflict with the work to be performed.

1.3 Problem Statement

The tools and methods described briefly above all have their part to play in the design of an interactive computer system. However, as was pointed out, they do not focus on the context of use of a product. In order to know what product would be most useful to build, we need to understand about the context of use. To this end, we need different tools. Tools and methods are required that encourage designers to use contextual information efficiently in design. However, this thesis argues that current methods of contextual design can often hinder designers in their attempts to use contextual information efficiently.

There are a number of existing methods for eliciting information about the context of use of a proposed system and for using that information in design. For example, Customer Centred Design (Holtzblatt and Beyer, 1993) provides modelling tools for structuring the users' context plus methods for analysing this information for use in design. Other methods such as ethnographic engineering (Mateas *et al.*, 1996), Contextual Inquiry (Wixon and Raven, 1994) and a host of others attempt to capture the important aspects of users, their work and their environment and present this knowledge in such a way that it can be used succesfully in design. Many of the methods for collecting information about users and their context have their roots in the social sciences such as sociology and anthropology. They are well tested methods which have only been recently applied to interactive systems design.

However, this thesis argues that none of the methods available to designers describe successful ways to use contextual information efficiently in design. While the methods described above are strong on the collection of contextual information they each suffer from a number of weaknesses when it comes to using that information in design. Some methods assume that the people who use the information are the same people who collected it and will be able to apply the knowledge gained while collecting information in designing a product. Other methods suggest that a succession of models over the descriptions of context will lead designers to effective designs. In reality, those who collect data for a project will often not be the same people who attempt to use it in design.

However, even if the designers and maintainers of a product are the same people that collected information about the product, using successive models or any other technique that creates abstractions over the contextual data that are not explicitly tied to the contextual data itself bides the ways in which the data is used. It then becomes possible to suggest a design idea that has no grounding in the contextual data and it becomes impossible to say for certain how much influence the contextual data has had on the design. Since the way the data is used is hidden, it becomes impossible to assess how the data has been used, thus leading to difficulties in identifying inconsistent usage of contextual information.

Thus, the main problem facing designers using current contextual design methods is that it is impossible to say how efficiently contextual data has been used in design. Since relationships between context and design are not recorded, designers cannot accurately and easily determine how much contextual information has influenced the design nor exactly what the influence is.

1.4 Aim

This thesis has one main aim. It is –

> *To investigate the issues involved in creating and maintaining a set of explicit relationships between contextual information and design.*

The result of this investigation will be to discover whether or not explicitly describing the relationships between context and design will help designers make more efficient use of context in design. However, other issues that may have an effect on the efficiency with which contexual information is used in design also need to be investigated. Hence the thesis also aims to investigate other issues involved in making efficient use of contextual information in design.

As identified by Cockton *et al.*, (1996), using contextual information in design more efficiently than in current methods means using the information in such a way that makes clear the way the contextual information has influenced the design. By making clear the influence that context has over the design, assessments can be made with regard to the use of contextual information in terms of appropriateness, consistency and completeness. Thus, designers can be encouraged to make more efficient use of context in design.

1.5 Research Method

To achieve the aim of the thesis, Chapter 2 reviews the current literature on using contextual information in design to identify common issues and problems that designers encounter when attempting to use contextual information in design. The review identifies the notion of explicitly representing the relationships between contextual information and design as being of high importance. Chapter 3 picks up this theme and investigates the issues involved in maintaining a set of relationships between context and design. The results from this investigation form the basis of the design and development of a computer based support tool that is described at the end of Chapter 3. Chapter 4 describes three case studies that informally validate the tool and demonstrate how it could be of use to designers. Chapter 5 describes attempts to formally validate the first tool that was developed as part of this research. Such a formal validation adds extra evidence of the effectiveness of the tool to the evidence that was gathered in the informal validations described in the case studies. Chapter 8 draws the previous chapters together and presents the conclusions of the thesis. Here, the aim of the thesis is reviewed and the contribution of the thesis towards this aim is evaluated. The results of this evaluation form the basis for Chapter 9 which concludes the thesis by describing possible future work.

Appendices F & G describe an investigation of the way contextual information is used in a particular industrial setting and provide insight into issues that are not central to this thesis but are interesting nevertheless.

2. Chapter 2 Literature Review

2.1 Introduction

The claim that designers must make use of contextual information (in some way) in design has been taken more and more seriously within HCI over the last decade, as demonstrated by the growing number of publications that describe different methods or theories about using contextual information in design. The methods present an interesting picture of the different ways that context can be used in design. For example, both Whiteside *et al.* (1988) and Bevan and MacLeod (1994) advocate using contextual information to inform usability evaluations. Two related methods, Contextual Inquiry (Wixon and Raven, 1994) and Customer Centred Design (Holtzblatt and Beyer, 1993) describe ways in which contextual information can be used to learn about users' work and how that information can be used to develop successful computer based applications to support users in their work. Mateas *et al.*, (1996) describe their attempts to use contextual information to learn about a culture or group of people, in order to identify product ideas or concepts acceptable to that culture. Rosson and Carroll (1995) attempt to use contextual information to identify objects that should be implemented in an object oriented system.

This chapter reviews different methods of using contextual information to influence designs, as well as related methods and theories such as participatory design and activity theory. Throughout the review, the five factors described in Chapter 1 are used, when appropriate, to evaluate the methods described. The review also identifies the main issues involved in using contextual information in design, such as defining context and identifying relevant context. At the end of the review, the five factors will be reviewed and a number of conclusions about current levels of support for these factors will be discussed. These conclusions form part of the rationale for a four stage model of using contextual information in design, described in Chapter 3.

The literature review begins by examining a number of methods for using contextual information to inform the design process. Ways in which contextual information can be used to investigate different cultures, to identify markets and to identify design concepts to fit inside those markets are then described, followed by a description of activity theory and the ways in which it has been used in interactive systems design. Participatory design, a technique that aims to bring users and developers together in partnership throughout a design project, is then described. Scenarios and the ways in which they can be used are then reviewed. The chapter concludes with a description of some issues that are common to all methods that attempt to make use of contextual information in design.

2.2 Contextual Inquiry

The Contextual Inquiry method grew out of earlier work at Digital Equipment Corporation. Whiteside *et al.* (1988) had worked on the *Usability Engineering* technique which used contextual information to define usability goals for particular designs. Some of Whiteside's colleagues continued work on this method and investigated other ways in which contextual information could be used in design. This work resulted in the Contextual Inquiry method. Instead of collecting data about users and their work to derive contextual usability goals, Contextual Inquiry sets about collecting contextual data and representing it so that the critical issues and problems in users' work are apparent. By making these issues apparent, it is hoped that a design that addresses these issues can be more easily derived. Hence, contextual information is put into use early in the design process, and is used to represent and validate designs with respect to users' work. In contrast, the Usability Engineering approach, from which Contextual Inquiry is derived, used contextual information solely to identify and specify usability goals which were evaluated at the end of a design project.

One of the major contributions that Contextual Inquiry has to make to any work on using contextual information in design is its focus on collecting quality data. Quality data is data that accurately reflects the reality of the work situation under investigation. If designers use contextual information that does not accurately reflect reality, then the design produced will be unlikely to satisfy the needs of the users, since design decisions will have been based on inaccurate data. Wixon and Raven argue that to ensure high quality data, users should be interviewed about their work at their work place rather than in an unfamiliar environment such as a usability laboratory. The familiar environment allows interviewees to refer to and use artefacts of their work as cues for answering questions about their work and to demonstrate certain aspects of their work. The Contextual Inquiry method identifies collection of quality data as a prerequisite to using data effectively and suggests that any method that attempts to make effective use of contextual information in design should pay attention to the methods by which contextual data is collected.

In Contextual Inquiry, interviews are used to collect data about the users' work. Instead of simply relying on a set of questions to guide the interview, questions which could easily be discarded as a result of what happens in the workplace, the interviewers define a *focus* for the interview which states the concerns for the interview. Thus, interviewers shape their questions such that they fit the focus, both before and during the interview.

Wixon and Raven (1994) describe the Contextual Inquiry method. The authors begin by defining context. They believe that context is "The *interrelated conditions within which something occurs or exists.*" So the context of users' work is the work itself and the environment in which it is performed. The context of the users' work includes everything which involves the user when they are working. The authors suggest that this includes the factors listed in Table 1 and that all of these factors can, and should, influence the design of an artefact in some way.

User's work	User's work space
User's work intentions	Tools used
User's words	How people work together
Business goals	Organisational/cultural structure

Table 1. Aspects of context in Contextual Inquiry

There are a number of methods for collecting the above kinds of contextual information, all of which take place in the field. Before any data collection begins, though, a focus must be set so that the analysts are aware of the relevant data they are looking for. Wixon and Raven stress how important it is to *"Listen and probe from a clearly defined set of concerns"* (Wixon and Raven, 1994, p23). By defining the set of concerns, analysts can clearly identify what is and is not relevant during the collection of data. For example, if an analyst were interviewing designers about how they designed a particular product in order to gather information to improve the design process, the tools used by the designers, the areas where they work, who they work with, etc., would all be valid concerns and would be included in the focus. Analysts and designers should attempt to compile open–ended questions which the user will be able to answer. Answers to the questions should reveal aspects of the users' work. Focus creates a perspective on user's work. It reveals *and* conceals things about users' work. It directs questioning and creates an understanding of the work. Each designer will have a different focus, a different perspective, and the role of setting a focus before any analysis of the work takes place is to ensure that each analyst agrees on the focus. Wixon and Raven point out, however, that even with the same focus, different analysts will still see different things.

Once a focus is set, interviews and observation of users at work can then be carried out. This involves either sitting down with the user at their work or watching a video tape playback of them at their work and asking questions to elicit the concepts, words, methods, etc., that the user uses while they work. These methods generate a lot of data and the analysis of this data can be time consuming. It is important, however, to collect this data and to ensure that it is 'good' data. Since the analysis and design is based on the data collected, if the data is untrustworthy then the design will likely be flawed. Contextual interviews take place in the users' workplace, a familiar context. This allows interviewees to refer to artefacts of their work in order to answer questions about, and perhaps to demonstrate, certain aspects of their work. It may also put some interviewees more at ease than if they were being interviewed in an unfamiliar environment.

While the method focuses on the collection of contextual data, Wixon and Raven suggest a number of techniques for using the data in design. Contextual data can be structured appropriately so that it can be used effectively in design. Wixon and Raven suggest that affinity diagrams can be used to provide a structure for the contextual data. From the transcripts of all the interviews, the analysts write down ideas and concerns that were expressed during the interviews. These ideas and concerns are written on Post–It notes, which can then be stuck on a suitable surface (e.g., a wall). Similar ideas and concerns are grouped together. If more than four items are grouped together, it may be possible to divide the group into two smaller groups, held together by one higher level unifying idea. Thus, a hierarchical structure is created, with the highest level (most abstract) concern at the root of the structure and the low level (concrete) items at the leaves (as shown in Figure 1).

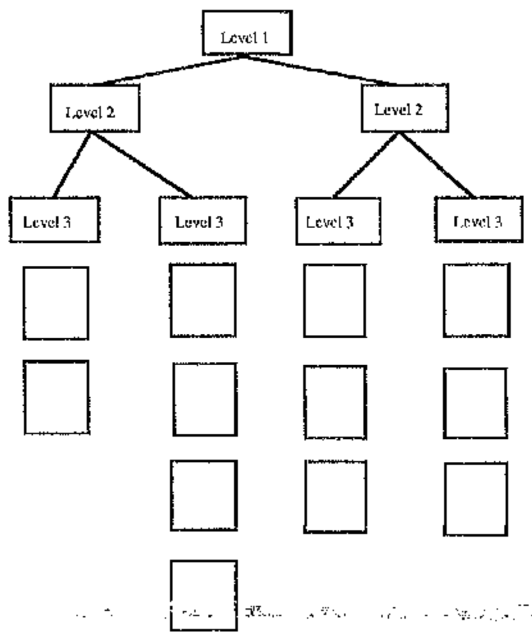


Figure 1. A typical structure for an affinity diagram

According to Wixon and Raven, structuring the information helps answer any questions or assumptions that were present in the focus, records core aspects of user work and tool use and can represent the initial design (although they do not say how or how well). Wixon and Raven stress that it is important to avoid predefined categories and that groupings should be allowed to emerge from the data. The groupings highlight the concepts and concerns that are present in the users' work. For example, when interviewing designers about how they design, answers to questions may be grouped in categories such as 'communication', 'identifying ideas' and 'demonstrating ideas'. These groupings suggest that these are the three main concerns that any system designed to support these designers should address.

Wixon and Raven suggest that the contextual data can be used to create work scenarios in order to test prototypes. The design can be refined in an iterative fashion, evaluating it at each stage with the users in their work context. Contextual data may also reveal work metaphors which can be represented in the interface. Work metaphors can be evaluated iteratively with users as well, perhaps in combination with the work scenarios. Designers should iterate a design from the concepts and concerns represented in the affinity diagram until a design which fits the context of use is produced.

2.2.1 How well does Contextual Inquiry support the five factors?

Scope

The contextual inquiry method uses a focus to determine the *scope* of contextual data that will be collected. Interviewers determine the scope of an interview beforehand and use the scope to guide them in the questions that they ask and in determining if new insights into users' work are worth investigating. Interviewers should only be concerned with collecting data that is relevant to the focus that was defined prior to the interview. Wixon and Raven suggest that after it has been collected, the data can be structured in such a way that the questions present in the focus can be answered. Hence, if the focus was concerned with editing documents, the data should be structured so that it answers any questions relating to editing documents. Hence, contextual inquiry supports the explicit definition of the scope of relevant contextual information, which may reduce irrelevant or extraneous contextual data that interviewers collect.

Links

The contextual inquiry method makes it difficult for designers to determine the *links or relationships* between representations of context and design. Such relationships exist, but they are implicit. The method employs two intermediate-representations between the contextual data and the design. The first representation is the affinity diagram or any other structure that answers the questions that were present in the focus. The second (and possibly further) representations are the scenarios, work metaphors etc., that are suggested by the data. Following these representations is the design itself.

Relationships between these representations are not recorded however. Hence, the gap between contextual data and the design is wider and less clear, making it more difficult to assess the influence that the contextual data has had on the design. This means that it is difficult to determine how well a design will fit its context. For example, Britten and Reyes (1994) describe their use of Contextual Inquiry to investigate expert usage of the Mosaic web browser. They spent roughly 30 hours collecting and analysing data regarding a typical expert's usage of Mosaic. They structured this data into an affinity diagram, as suggested by Wixon and Raven but then further refined the affinity diagram into what they called an

'Index of Understandings'. This is basically an indexed list of important points that became clear during the construction of the affinity diagram. The authors chose to represent the information in this way since the index provides a more secure form of storage as it is an electronic, textual document as opposed to the affinity diagram which is constructed using Post–It notes and is not as easy to use as the index. The index constructed by Britton and Reyes consists of 9 A4 pages of detailed information, an example of which is shown below.

D Social Catalyst Mechanisms

1 The subject stated that mailing lists and newsgroups are a better means than Mosaic for determining status of an interest area. Newsgroups allow many to many personal interactions, mailings lists and WWW allow one to many personal interactions, electronic mail is primarily used by the subject for one to one personal interactions.

2 He mentioned his dissatisfaction with the fact that newsgroups usually degenerate from being a forum for researchers to share information to a less academic forum in which "any jackass that can bray" will post a message.

3 He stated that a common mode of e-mail use is "quick back & forth" interaction with a person. He expressed an interest in obtaining an "intelligent agent" application program to filter his incoming mail. This was motivated by a misguided e-mail message he received from a student.

4 He mentioned his feeling that WWW is more "archival" than newsgroups.

E HTML Linearization Tool

1 The subject uses a tool, WWWRefs, which takes an HTML file and moves all the links to the end of the file. Additional information on the tool is available from http://info.cern.ch/hypertext/WWW/LineMode/Defaults/Distribution.html This causes the resulting page to be "more recognizably scholarly readable," that is, more akin to the style of text organization found in scholarly works. The resulting page is a text file without any "live" links. He has found this format to be better for e–mailing to people than the alternative of mailing HTML pages via Mosaic's mailing functions. Sometimes WWWRefs does not format the resulting page correctly. In the example that he demonstrated for the interviewers, WWWRefs incorrectly formatted some line breaks and tab stops.

2 After using WWWRefs, he "strips off" references using a separate tool. This operation is not clearly understood.

From the 9 page long index of understandings, Britton and Reyes produced 8 design changes that they recommended be made to Mosaic so that it could better support expert usage. Each design change is described in textual form and each is approximately a paragraph in length. A couple of the suggested changes are shown below.

1. "Webizer" functionality is needed: Mosaic can become a much more powerful WWW browser with the addition of a "webizer". A webizer would possesses the ability to scan the contents of a directory and produce an HTML (Hypertext Markup Language) summary file describing the contents of that directory

(such as text, image, sound, and subdirectories), in a more easy to browse form. For example, text files could have their first line displayed (cf. the MH scan command), graphics files could be represented by a thumbnail sketch, and sound files could play briefly when somehow selected (perhaps by positioning the cursor over the icon for the file).

2. Hotlist must be radically changed: The Mosaic Hotlist functionality must be changed if Mosaic is to be useful for helping the user manage the items on his or her Hotlist. The current Hotlist requires that the user visually, sequentially scan the list of items in order to locate the desired one. This can be time consuming if the Hotlist has many items. The Hotlist needs to become an HTML file, viewable and able to be acted upon from within the Mosaic Document View Window, able to be sorted in different ways, able to have the document URL's (Uniform Resource Locator's) title easily edited, but not allow the duplication of URLs.

Difficulties are encountered when attempts are made to relate the index of understandings to the design changes. Some of the difficulties are caused by the size of the index. It is difficult to keep track of all parts of the index that relate to any one particular design change since there could potentially be many related parts of the index. The main problem, however, is the implicit nature of the relationships between the index of understandings and the design changes. It is not clear which parts of the index are related to the design changes and which parts aren't. One has to attempt to re–enact the original design process and thoughts of the original designers in order to tease out the potential relationships. This is a difficult task and one which does not guarantee accurate results since the data represented in the index of understandings could have been used in many different ways.

Quality

The *quality* of the data that analysts collect using Contextual Inquiry cannot be guaranteed. The data is only verified at the time of the actual interviews when the interviewer has the opportunity to reinforce what the interviewee is saying or doing. After the interview, the interviewer has very little contact with the interviewee and the data that they have collected is subject to different, perhaps incorrect, interpretations, particularly when the interviewer attempts to explain the data to other members of the design team. From experience using these methods gained in the IT case study described in Chapter 4, it is clear that quality data cannot be guaranteed solely by two or three interviews of users. The process of working on the design after the initial interviews clearly identifies elements of contextual information for which data was either not collected at all or was incomplete. Collecting quality data cannot be separated from the process of using it.

Understanding

Wixon and Raven suggest that all interviewers participate in creating an affinity diagram which is based on all the data that each interviewer collects. This can be a useful exercise in terms of generating an *understanding* of the data since it makes the understanding explicit and shared amongst the whole team (i.e., everyone can contribute to and discuss the structure of the affinity diagram). However, through gaining experience generating affinity diagrams at a tutorial conducted by Wixon and Raven, it was clear that the creation of categories over the contextual data can lead to individuals within the design team reinterpreting their own data in light of the categories being created. One effect of this can be that individuals discard some of the data that they collected since they now doubt its validity if it doesn't easily fit into the affinity diagram.

Definition

The *definition* of context offered by Wixon and Raven– *"The interrelated conditions within which something occurs or exists"* suggests that the context of the object of analysis, for example, users' work, is the conditions within which that work is performed and the ways in which the work relates to other situations. It is a very 'situated' definition, focussing on conditions within the workplace, the tools users use to perform their work, the environment they work in, etc. It is vague and broad and does not offer guidance to designers in identifying what the relevant contextual factors for particular projects will be.

Summary

The table below (Table 2) summarises the above discussion. It rates the level of support provided by the Contextual Inquiry method for each of the five factors (1 = poor support, 5 = excellent support.).

	Scope	Definition	Quality	Understanding	Linking
Contextual Inquiry	4	3	2	3	1

Table 2. Level of support provided by Contextual Inquiry for the five factors

In terms of supporting designers in design, the Contextual Inquiry method offers little, although it is fair to say that it also promises little. It is primarily a data collection and structuring method. While Wixon and Raven claim that Contextual Inquiry allows developers to get at the important data quickly and to be able to design from this data, the effectiveness with which this can be done depends very much on the skills and experience of the designers, since Contextual Inquiry alone does not offer any support for this process. In response to these difficulties, Holtzblatt and Beyer developed the Customer Centred Design method. This is an extension of the Contextual Inquiry method, that provides some form of support for designers in using contextual information in design.

2.3 Customer Centred Design

Attempts have been made to extend the Contextual Inquiry method such that it supports the generation of designs better and does more than represent users' work. Karen Holtzblatt, who was involved in both the development of Contextual Inquiry and the Usability Engineering methods, left Digital and developed the Customer Centred Design method (Holtzblatt & Beyer, 1993), which is an extension of Contextual Inquiry.

Customer Centred Design uses the same techniques as Contextual Inquiry to collect data. Contextual interviews are performed with the users, in their workplace, by a team of designers. These interviews normally take half a day to perform with each analyst interviewing a different user. Afterwards the analysts meet to discuss the interviews and construct an affinity diagram, in the same way as suggested by Wixon and Raven, which captures the group's insight into the users' work. While Wixon and Raven now suggest that the contextual data can be used to suggest scenarios, prototypes etc., Holtzblatt and Beyer apply their *work modelling* tools to the data.

Holtzblatt and Beyer have developed a number of models which can be used to show the work of a single person or organisation. These models show the flow of information, the different roles, communication patterns, task steps and motivations and artefacts used to perform the work. Five different kinds of models are recommended by Holtzblatt and Beyer: *context model, physical model, flow model, sequence model.* and an *artefact model.*

A context model shows how "Organizational culture, policies, and procedures constrain and create expectations about how people work and what they produce. Context work models represent standards, procedures, policies, directives, expectations, deliverables and other constraints." (Holtzblatt and Beyer, 1993, p.95). The context model shows how the context constrains the various participants and highlights what can and cannot be changed in the current system. It provides a means with which to evaluate the consequences of any proposed changes to be made. An example context model is shown in Figure 2.

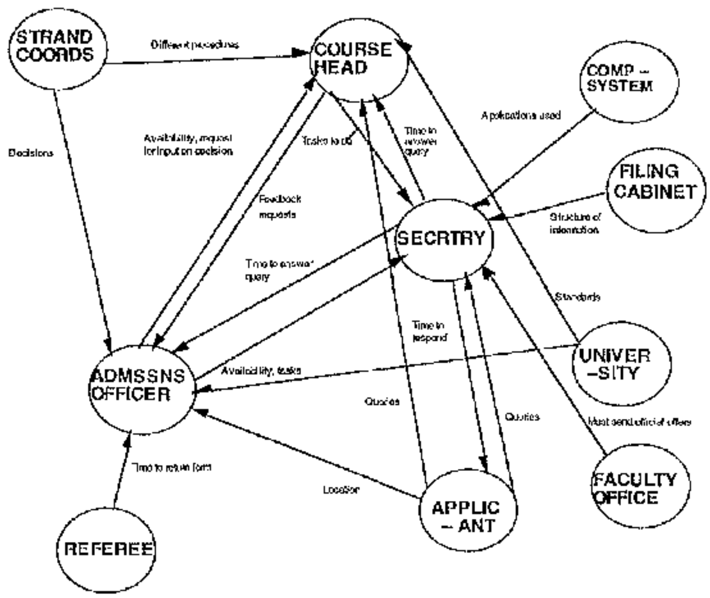


Figure 2. A context model

(taken from the IT case study described in Chapter 3.)

The model represents the context of the process by which applications for Glasgow's M.Sc in IT course are processed. It shows a number of different constraints, for example, that the admissions officer is constrained by the referee, in the time that referees take to return forms. One of the things that this example illustrates, is that the diagram cannot be detailed enough to fully explain the constraints and their effects. For example, the diagram does not describe *why* referees take a while to return their forms and it does not describe what the effect of this constraint is. It merely indicates that there is a constraint that must be addressed. The other models are similar in this respect.

The physical model "represents *the physical environment as it impacts the work*" (ibid., p.95). What we are concerned with when modelling the physical environment is not simply modelling the way things are, but rather, modelling the way things affect work. For example, the physical model can show that the printer which a secretary uses is in another office and that the secretary has to walk to that office to retrieve the printed output. The fact that this is shown in the physical model makes it an item of discussion. The secretary might be able to perform her job more quickly if she had a printer of their own. On the other hand, the fact that the secretary has to use a shared printer might mean that she comes into a considerable amount of contact with other colleagues and have more informal opportunities available to discuss work with others. The physical model does not show the workspace of the users to any scale. It simply shows the important physical characteristics (e.g. location) and how these characteristics affect the work.

The flow model models "the important roles people take on" (ibid., p.96). Roles are defined as "a set of responsibilities and associated tasks for the purpose of accomplishing a part of the work". Roles are generally not assigned to one person but to many. Also, people usually have a number of roles to perform. By modelling the roles, we are identifying the different types of actions and tasks that will be performed with the system. The flow model shows the communication between roles, what format it takes and what data is passed between different roles. Importantly, the flow model also shows the needs of people who, while not directly involved with the system, depend on the information or results in the system. By modelling these roles, a new system can be built that better supports them.

Sequence models *"represent the sequence in time of actions for specific important activities"*. They show the specific tasks users perform and are similar to task analysis in that they break a task down into a series of steps which are performed in sequence, with varying amounts of iteration and decision making. They define the work the system must support and show how it can be changed by combining, removing or adding steps.

Lastly, *artefact models* describe artefacts used by the users to perform their work. These models show the detail behind the artefacts, describing the structure, usage and intent of the artefact. They guide the design of new artefacts by revealing what is used and what is not used in performing work. By showing the structure of present artefacts, which can reveal conceptual structures used by the users, they indicate appropriate structures for new artefacts.

Once the work has been modelled by using the tools described above, work redesign can begin. As with the Contextual Inquiry method, affinity diagrams are created from the data collected from users. These diagrams help structure and organise the vast amounts of design ideas, key points, insights and questions that are generated as a result of both interviewing users and representing their work in the models described above. The authors recognise however that these diagrams are not sufficient to be able to represent users' work practices efficiently. In order to enable discussions about work as it will be performed, new models are used which model the common aspects of users' work across all of the models. These models omit individual details and only show what is important to all of the users. *Consolidated models* of each of the five different models described above reveal the underlying structure of the work and show what will be supported by the new system. Anything not represented will not be supported. When complete, the consolidated models serve as statements of how the users will work when the new system is complete.

By modelling the common aspects of users' work, designers are now in a better position to design a system that will work for the majority, if not all, of its users. Holtzblatt and Beyer recognise that there is little support available for designers to go from knowledge about the users to the design. In addressing this, they have introduced a new modelling language called *user environment design* which defines the environment the users will work in. From the consolidated models the designers can identify each activity that the users will engage in. To design support for these activities, the authors take inspiration from everyday life. Just as in real life when performing an activity, we group together everything we need for that activity and disregard those things we do not need, Holtzblatt and Beyer use the user environment design to do the same thing for each activity that the computer system is to support. The user environment design (UED) identifies each activity and the objects and functions that are required to perform that activity. Designers can then map the UED onto the user interface. Generally, in a windowing system, each activity is mapped to a particular window (similar to a house, where different activities are performed in different rooms e.g., cooking in the kitchen, eating in the dining room). Customer iteration is encouraged throughout this whole process so that no mistakes are made. Paper prototypes are used initially, then as the design progresses more work is applied to the interface, eventually building working prototypes that can evolve into the full system.

2.3.1 How well does Customer Centred Design support the five factors?

The Customer Centred Design approach is an attempt to closely *link* data about users and their work to the design of systems to support users in their work. In this respect it attempts much more than the Contextual Inquiry approach, which does little more than collect and structure data. It represents work in many different ways, allowing different perspectives on the work.

Many perspectives on users and their work are supported through the provision of at least eleven different types of models. These are the context, flow, sequence, physical and artefact models together with affinity diagrams and consolidated versions of each model. In addition, user environment designs are produced as well as paper prototypes of the system. While the goal of supporting the transition between data collection and design is a laudable one, experience suggests that mapping the transition with a multitude of models does not help in making the relationships between context and design clear. Indeed, the number of models and amount of data created can swamp a design team. In personal discussions with industrial designers, it was clear that many of them feel that the whole approach is "monolithic" (Czerwinski, 1996). Many designers find that they do not know how to handle and manage the large amounts of data. They find difficulties in maintaining the data and ensuring that all the data is used to its best advantage. It can be difficult to see exactly how the data has been used in design. This would suggest that any method that attempts to make effective use of contexual information in design should not overload designers with a large number of different representations that can be used to represent users' work and related designs. The number of representations of work and design should be kept to a minimum so that the relationships between the different representations are made clear.

Scope

One of the reasons for the large amounts of data generated by the method is that the method does not suggest, unlike Contextual Inquiry, that designers define a focus or *scope* on the contextual data. The method seems to ignore the issue of relevancy. Instead of defining a focus, Hotzblatt and Beyer suggest that data should be collected about everything, since designers "do not know what is important to ask about [in users' work]". The benefit of defining a focus, as suggested by Contextual Inquiry, is that it limits what data is collected, meaning that less irrelevant data should be collected (in theory at least). The lack of a focus could, in part, explain why some designers feel that the Customer Centred Design method is monolithic and difficult to manage.

Definition

However, a certain amount of scoping is introduced since only those elements of context that can be represented in the models are collected. Holtzblatt and Beyer *define* context as the "Organizational culture, policies, and procedures [that] constrain and create expectations about how people work and what they produce". These constraints are shown clearly in the models. However, as is the case with Contextual Inquiry, the *definition* employed by the Customer Centred Design approach is vague and broad. Designers may not be as aware of the effects that the definition has on the type of data that they collect as they would be had they explicitly defined a focus of their own, particular to their project.

Quality and Understanding

In terms of *quality* of data and *understanding* the data, similar concerns arise as in Contextual Inquiry. Both methods use the same techniques to collect and (initially) structure the contextual data, so as with Contextual Inquiry, good quality data cannot be guaranteed. Furthermore, while both the Contextual Inquiry and Customer Centred Design methods aim to generate a shared *understanding* of the contextual data, this depends on features of the affinity diagram, such as the range and amount of categories and the amount of participation by all members of the design team in creating the affinity diagram. However, because the Customer Centred Design method uses many more different types of models to structure the data, there are more opportunitues to introduce inaccurate descriptions of users' work.

Linking

Holtzblatt and Beyer have recognised the need for *linking* between different representations or models. They describe the benefits of attaching design ideas to affinity diagrams via Post–It notes – "Later, when the team picks up these ideas to develop, they will be directly tied to the customer data which sparked them" (Holtzblatt & Beyer, 1996). The trouble is that Post–It notes are difficult to maintain. They can easily fall off the part of the diagram they were originally attached to. They can swamp a diagram, if many Post–Its are attached to the same area in a diagram. An idea may be relevant to two different areas on the diagram, necessitating the creation of two Post–Its or more. If the idea is modified in any way, all the identical Post–Its must be identified and modified accordingly, a task which becomes all the more difficult, if other Post–Its are placed on top or if some of the Post–Its have fallen off the diagram. Britton and Reyes (1994) also suggest that an affinity diagram is too large and unwieldy a structure to be able to link to or refer to usefully in a design project.

However, this is not to say that the models themselves are not useful. Individually, each model supports the description of important aspects of context, such as constraints placed on users, tasks and roles performed. By their very nature, the models make the designers aware of certain aspects of context that should be investigated. For example, to draw a physical model, designers need to investigate the way that the physical layout of the users' environment impacts upon their ability to perform their job. However, the authors say that each of the models is useful to model nearly "every problem" (Holtzblatt & Beyer, 1996). Hence, they suggest that designers should create separate instantiations of every model for their particular project, so as to understand all of the users' work.

Summary

The table below (Table 3) summarises the above discussion. It rates the level of support provided by the Customer Centred Design method for each of the five factors (1 = poor support, 5 = excellent support.).

	Scope	Definition	Quality	Understanding	Linking
Customer Centred Design	2	3	2	3	3

Table 3. Level of support provided by Customer Centred Design for the five factors

As the above analysis has indicated, one difference between Contextual Inquiry and Customer Centred Design is how each method defines what data it should collect. Each method defines a different scope of contextual data. However, each uses the same data collection method. Contextual interviews, performed with a representative sample of the user population at their place of work, have several advantages over traditional interviews. For example, a contextual interview will collect a richer set of information since, as it takes place in the users' workplace, the interviewee has available a wide range of cues and artefacts with which they can illustrate their answers. There are a number of disadvantages however. Since the interview is primarily led by answers supplied by the interviewee and incidents that take place during the interview, it can be difficult for an interviewer to get answers to all the questions they had identified as relevant before the interview. The next section expands upon the advantages and disadvantages of using contextual interviews to elicit contextual data.

2.3.2 What effect does the data collection method have on the amount of support for the five factors?

Both Contextual Inquiry and Customer Centred Design use contextual interviews to collect data about users and their work. Contextual interviews are held with the users at their workplace, preferably while performing some work. Questions may or may not be predetermined, but in all cases, the interviews are led by the users and not the interviewers. As the users answer questions or perform actions, questions about these answers or actions arise. Therefore, the user performs naturally and does not have to jump from one aspect of their job to another, unrelated aspect, just to answer the interviewers questions. This 'partnership' between the interviewer and the interviewee (the user) recognises that the user is expert in their work and that the interviewer should look to them for the kind of information they require and should not rely totally on some other influencing factors, such as their previous experience with similar projects.

However, due to the necessary individuality of each interview, it can be difficult to get a representative set of data. Raven and Flanders (1996) in their description of Contextual Inquiry, suggest that interviewers visit three different sites and interview three different people at each site. This will ensure that common aspects of users' work will become apparent, according to the authors. However, in a commentary on the Raven and Flanders paper, Susan Kleimann questions whether or not this can be achieved so easily. When other demographics, such as age, gender and experience are taken into account difficulties are increased.

> "What was a fairly simple 3 x 3 matrix for nine interviews, now becomes much more complicated by expanding the number of interviews and increasing the cost and time. If the matrix is not expanded or if access is not available to three users from each representative market, then the study is counting on a single user from a particular type of market to provide relevant information. No matter how that the methodology provides a thick, detailed picture of one user's use, the methodology provides a very narrow range of user representativeness and limits to generalizability" (Kleimann, 1996)

Such difficulties were experienced in the IT case study described in chapter three. To collect initial contextual information for the case study, all of the different stakeholders involved in processing applications for the M.Sc in IT run by the Computing Science Department at the University of Glasgow were interviewed except for the applicants. Since applicants come from widely varying backgrounds (overseas students, mature students, students having just completed their first degree, students who graduated more than five years ago etc.) it was impossible to interview a representative cross sample of the applicants in the time that was available.

Kleimann also discusses the difficulties that are encountered in getting access to users. Some corporations will supply interviewers with those users that will present a good face of the corporation. Others will supply those users who volunteered to be interviewed. These kinds of users often turn out to be more capable, more articulate or have bigger axes to grind than more typical users. In each case, interviewers will not be collecting data regarding the typical problems that users face.

Kleimann goes on to argue against the reliance on a focus for determining what is and is not relevant. She picks up on the fact that the focus conceals aspects of context as well as revealing others. Kleimann says

> "one needs to be particularly careful of selecting information and data so that, while the hypothesis is supported, contrary information is not ignored. Much useful information can be found in the white spaces – that is, the parts that we don't focus on, the information that is not the focus of our research. Such a carefully planned focus on the analysis requires an equally conscious check on the information that is being ignored" (Kleimann, 1996).

Lastly, Kleimann discusses the changing situation of the workplace in which it may not always be possible to follow the method as described in its ideal form. For example, what should be done when project constraints are tightened and the two weeks that were available for user interviews is cut short to one week? Where should the priorities be set? Should the number of interviewees be cut by half or the number of iterations be cut? What stage should be cut such that it causes least harm? These are questions that Kleimann does not know the answer to, but she throws them up to highlight the fact that in this situation, which is common enough in commercial organisations, designers are not assisted by the method in determining priorities. One can argue that the focus helps prioritise in some way, that the issues identified in the focus should be addressed and answered, but how best to do this? And, in the case where no focus has been defined, such as in Customer Centred Design, what happens then? With no idea as to what is important, how can designers prioritise their time and make full advantage of the time they have?

Mark Simpson, in another commentary on the Contextual Inquiry method (Simpson, 1996), questions defining a focus for an interview in such a way that it is only concerned with those answers that the user is able to give. In agreement with Kleimann above, Simpson believes that the 'white–spaces' or inaccessible parts of the users' context are just as important as the obvious, accessible ones. Simpson also warns about being unclear about the subject matter of an interview. While the focus may be concerned with discovering the concerns with relation to system documentation for example, the actual subject matter of the interviews will be the users using the system. The way users work and use the documentation are all important to answer concerns about the documentation itself. However, the focus is not about the users' work *per se*. The distinction needs to be made clear to all members of the design team.

According to Simpson, one of the positive points about creating a focus is that it helps a team to reach consensus about the issues and problems surrounding for example, a particular piece of technology. Simpson claims

> "One of the real advantages of CI is that it can be an opportunity for members of a team to unite around a common focus and to jointly participate in getting data about and developing an understanding of that focus. The usual alternative is for the data—gatherer(s) to gather the data, and then throw it over the wall to those who must do something with it." (Simpson, 1996)

Salvador and Mateas (1996) confirm Simpson's claim. Salvador and Mateas were members of the Research Relations group at Intel. The group used contextual information to design new products. They experienced similar difficulties in relaying information that they collected via ethnographic techniques to other members of the group (Salvador and Mateas, 1996). Since the other members were not involved in the collection of that information, they found it difficult to understand and to grasp its significance. The next section describes the method by which the Research Relations group collected and used the contextual data.

2.4 Ethnographic Engineering

Ethnography is a branch of anthropology that attempts to describe individual human societies. The descriptions are built up over long periods of time in which the ethnographers immerse themselves in the particular society. By spending long periods of time investigating a society or culture, the ethnographers collect large amounts of detailed information about the society. Like Contextual Inquiry, ethnography attempts to collect quality data by collecting data 'at source' (i.e., from the context it relates to). However, its methods for collecting the data are radically different from Contextual Inquiry, which has ramifications for the way in which the data is or can be used. Hence it is worthwhile to review a particular industrial application of these techniques, to see if they add to our understanding of what it means to use contextual information effectively in design.

Identifying potentially useful and succesful tools for a particular society or culture was a main goal of the ethnographic engineering project at Intel Corporation. By investigating a particular culture, researchers at Intel hoped to identify possible areas in which computer based tools would be so useful to the group that they would be accept the tools. In contrast to the contextual interviews undertaken in the Customer Centred Design method, the Ethnographic Engineering project has employed traditional ethnographic techniques.

Mateas *et al.* (1996) describe their experiences in studying families with young children in the families' homes in order to identify possible lucrative markets for computer based products. The methods of analysis were very different to the contextual interviews described previously.

After identifying a number of families who were prepared to take part in the study, the authors met the families over dinner at the homes of the families. The authors took pizza dinners to each family and shared the dinner with the family. This gave both the authors and the families time to get to know each other and gave the authors an opportunity to further explain the purpose of their visit. After dinner, the authors took a tour of the families home, noting the physical layout of the house and the position of various artifacts such as electrical applicances, telephones, calendars, telephone books, etc. The tour also gave the authors the opportunity to learn more about the various activities that take place around the house, such as washing, eating, playing, etc., and when and where these activities took place.

After the tour, the authors spent time individually with members of the family to learn more about each persons typical day. Normally, one author would sit with the parents to discuss their typical day and another would sit with the children, typically in their bedroom. Weekly activities were discussed also, such as sports, music lessons, meetings etc. 'Fuzzy–felt' boards were used to represent a model of the home so that family members could 'walk–through' their typical day using the model.

2.4.1 Modelling

The authors note that ethnographic techniques yield an enormous amount of data and that it can be difficult to structure this data in such a way that it is understandable by others who were not involved in collecting the data. The data should be modelled to facilitate validation of potential computer based products and to let people with different backgrounds understand it.

Each family team (2 people) initially met separately to discuss their analysis of the particular family they had seen. Then the teams met together to discuss each family and to identify common themes and structure across the families. The resulting model (Figure 3) organised the data that had been collected by each team across the dimensions of space, time and social communication.

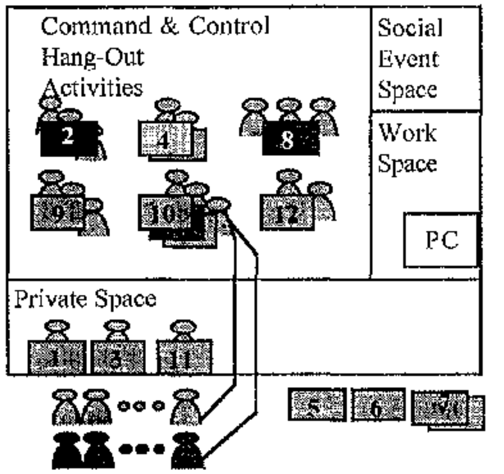


Figure 3. A model of behaviour of families with young children

(taken from Mateas *et al.* (1996))

The model shows that separate activities occur in separate spaces. Numbered boxes represent significant chunks of time in a typical day. Hence it can be seen that most of the activities take place, throughout the day, in the Command & Control centre. Also, most of these activities are social activities that involve interaction with other members of the family (the clusters of people around each numbered box). The only individual activities typically take place in the private space.

This model captures the essence of the ethnographic analyses that the authors undertook. In the true ethnographic spirit, it makes the familiar (the behaviour of a family with young children) seem strange (by modelling homes and activities in a non—intuitive way). The model makes people think about their typical day in a different way, in terms of spaces of activity, communication and time. Thus, important aspects of the typical day become clear, such as the fact that family activity is distributed throughout multiple spaces of varying significance. By focussing on what the model says about family life and what this implies for the design of computer based products, possible succesful products and ideas can be identified. For example, the typical PC can only be used in one space and forces its users to spend a significant chunk of their time to use it. It is also a very individual experience, with little affordance for interaction with immediate members of the family. According to the model presented above, this is incompatible with the typical family. The model suggests that families would make better use of a product that affords significant social interaction with immediate members of the family, does not take up large amounts of time and can be used in conjunction with the command and control activities and hang out activities that typically take up most of a family's day.

2.4.2 How well does ethnographic engineering support the five factors?

Scope

Ethnographic techniques define a *scope* over contextual information that concentrates on the point of view of those closely involved in the domain of investigation. They tend to neglect the point of view of other people or factors less closely involved in the domain. The very nature of ethnography forces the analysts to get as close as possible to the culture they are attempting to understand so as to learn as much as they can about that culture. It presents very much a view from the inside of that culture and can sometimes run the risk of paying too much attention to small details while ignoring larger details, simply because the view from inside a culture can hide the bigger picture. For example, the interactions that take place between different cultures may have a profound effect on individual cultures, but by concentrating on one culture and by becoming engrossed in that culture, these interactions may be overlooked. In the ethnographic engineering example above, some attention was paid to outside influences such as work, school and friends and relatives. Activities relating to these influences are shown outside the main activity space and connected by black lines.

Definition

Again, by nature, ethnographic techniques do not define what factors are relevant. The point about ethnography is that it is generally used to *discover* the relevant issues or factors in a particular domain. It does not attempt to define these factors.

Quality

Higher *quality* data, within the scope (implicitly) defined, can be collected than is the case with the Contextual Inquiry or Customer Centred Design methods, simply because the ethnographers have many more opportunities to validate their data. Ethnographers typically spend anything from a couple of weeks to a few years investigating a culture[2]. They can interact with the people they are observing and can use these opportunities to validate the data they are collecting. Ethnographers must take care however, that by interacting with people within the culture they are observing they do not alter that culture in any way. Ethnographers, and people using data provided by ethnographers, also need to be aware that while the data may be of high quality, it may not necessarily cover all important aspects of a culture due to the particular point of view from which the data was collected.

---

[2] Indeed, this is one of the major criticisms that is applied to ethnography, that it takes too long to be feasible in typical software engineering projects. See Hughes et. al (1995) for a discussion of these and other related issues.

Understanding

One of the major difficulties that may be encountered with ethnographic accounts of a culture or domain is in creating a common and shared understanding of the data. At Intel, there were many difficulties experienced in explaining the data collected during the families with young children ethnography that had been performed (and was described in the above paper). These difficulties were mainly due to the fact that the model presented a picture of what was a familiar domain (most of the design team who were trying to make sense of the data had families with young children) in an unfamiliar way. There was a certain amount of terminology that was unfamiliar (e.g., 'Command and control activities', 'Hang out space') that was crucial to understanding the model. Also, the model presented a lot of information graphically, of which some information was more salient than other information. For example, the model clearly showed that most families with young children use their personal computer in the 'work space', separate from the important part of the home where the command and control activities took place. Things that were less clear involved the temporal ordering of typical everyday activities (indicated by the numbered squares) and the different types of activities (the different coloured squares).

For an ethnographic analysis to be most successful, the culture studied should ideally be located in one easily accessible space. If parts of the culture are interspersed across large distances it becomes difficult for the ethnographers to get a good picture of the whole culture, since they cannot be involved in all aspects of the culture at the one time. Again, the ethnographic engineering project recognised this and so used models of the home to 'walk-through' a typical day in the life of family members rather than physically being at home, school, work etc alongside the family member for the whole day. These models also helped save time in the analysis, since to pick up on the most relevant features of a typical day only took the amount of time that it took to walk through the fuzzy felt model, instead of spending the whole day experiencing the day with the family member. Typical ethnographic analyses can take months to perform, which is time that most software engineering projects cannot afford to spend. There is a trade off between the amount of time spent collecting data and the quality of data collected and this has to be recognised and acted upon appropriately for each team.

Most importantly however, the ethnographic engineering project focused on how best to represent the results of their analyses to other interested groups such that they could fully comprehend it and grasp the implications present within the model. This requirement had to be met by the model since other groups within the organisation that the authors worked in were to be responsible for turning the ideas expressed within the model into viable products. This would be a difficult task if the model did not express those ideas clearly. The ethnographic engineering project states the clear need for a representation of a culture (or context) in such a way that it can be *understood* and used profitably by other groups and colleagues who were not involved in the collection of the data represented by the model. *Understanding* the data is difficult however for those not involved in the ethnographic analyses. Scharfstein (1989) claims that it is difficult for someone to fully appreciate a particular context unless that person has first hand experience of that context. Kyng (1995) makes a similar claim when he suggests that ethnographers or other facilitators cannot be expected to collect and describe contextual data to other interested groups in such a way that the same levels of understanding can be shared amongst all groups.

Summary

The table below (Table 4) summarises the above discussion. It rates the level of support provided by ethnography for each of the five factors (1 = poor support, 5 = excellent support. Areas left blank mean that there is no direct support for this factor.).

	Scope	Definition	Quality	Understanding	Linking
Ethnography			4	2	

Table 4. Level of support provided by ethnographic methods for the five factors

There are further approaches that attempt to model and understand a culture in order to design tools that will fit into that cuture. One such approach is the Activity Theory approach, now gaining prominence as a new paradigm in design (Nardi, 1996).

2.5 Activity Theory

Activity Theory is a Soviet psychological theory developed in the 1920s by Leont'ev (1974) and Vygotsky (1925/1982). Its main contribution to an understanding of using context in design is its emphasis on the consciousness of the user. It argues that there are both external and internal factors that are relevant to the design of tools to support users in their work. In contrast to the Contextual Inquiry and Customer Centred Design methods which focus on the users' work and external contextual factors, activity theory focuses to a large part on the users consciousness in describing their actions. It claims that to understand people and their behaviour, we must understand their consciousness and how it relates to their behaviour or particular activity that they are performing. In contrast to western psychology, which has tended to focus on the abilities of people to create mental representations of tasks and to use these representations to devise plans for performing activities, activity theory argues that consciousness is not solely located in a persons mind, but that aspects of a persons consciousness are found in their environment, in what they do and in how they do it. Thus, activity theory advocates that behaviour can only be properly understood in terms of the relevant, meaningful context in which that behaviour takes place. It cannot be understood in isolation, by concentrating on one aspect of the behaviour, such as the mental representations used.

Activity theory places a strong, but asymmetrical emphasis on the tools used by people while performing an activity. According to activity theory, tools mediate activity. The tools used by people shape the activity that they perform in some way, either by supporting some actions or opposing others. The tools reveal aspects of the activity and conceal others by their very nature. For example, a word processor with sophisticated formatting features may focus its users on formatting a document rather than editing its content. The way that a tool shapes an activity is built into the tool through its development history, the ways in which it has been used by other people in the past. Tools are not considered equal to their users in activity theory however. Tools do not have a consciousness. Instead, they are part of the users consciousness. Thus, people are considered more important than their tools. In activity theory, analysts do not consider people as 'nodes' in a system, along with machines and tools. Analysts do not reduce a system to a collection of tasks performed by people and a collection of tasks performed by machines or tools. Instead, analysts examine the activity that a person is attempting to perform, and describe the context in which that activity takes place, in order to understand that activity better. Their descriptions describe the activity and the constituent parts of that activity and how it relates to the tools that mediate the activity.

A detailed description of activity theory is now presented. It is taken mainly from Nardi (1996), an edited collection of articles concerning activity theory and its application to HCI.

2.5.1 What is Activity Theory?

In activity theory, human behaviour is analysed and described at the level of activities. Activity theorists define activity as a "form of doing directed to an object" (Kuutti, 1996). It is the process through which some change is brought about in an object. People may be engaged in a number of different activities at any point in time. Different activities are distinguished according to their objects. Objects are best described as the goals of an activity and can take any number of forms, as long as they can be shared and are accessible to everyone participating in the activity. For example, an object could be pasting up a newspaper, executing a plan or even developing an idea. As can be seen from the examples, the word object is used in the sense of objective. Hence objectives can easily be modified through an activity (i.e., an objective can be met by performing the activity).

Objects can be viewed like targets that provide those participating in the activity with something to aim for. Objects motivate and justify the activity. Objects are not static, however. They can change in non–trivial ways over the course of an activity. The most obvious change is that which occurs when an object is satisfied, for example when a newspaper has been pasted up successfully. They can also change in other, more complex ways. In the course of executing a plan, deficiencies in the plan could be discovered, necessitating the development of a new plan. Thus, the object of the activity is now changed to that of developing a new plan, rather than executing the plan. This example introduces another fundamental concept of activity theory, that of *history and development*.

If the activity of executing a plan is altered to developing a new plan, the conditions that caused the change in the object become a useful resource in the new activity. The conditions that identified the deficiencies in the plan should be used when developing a new plan, such that the new plan does not suffer from the same deficiencies. Therefore, the history of the object is an important resource that should be investigated and utilised by analysts. As Kuutii (1996) explains, "historical analysis of the development [of an activity] is often needed in order to understand the current situation.".

The last remaining concept to introduce is that of *mediating artefacts*. All activities require the use of some kind of artefact, be it a material object, such as a machine or something less tangible such as a set of procedures or laws. In any case, these artefacts mediate between the person performing the activity and the object of the activity. Thus, people are not directly related to the object of the activity but rather are indirectly related. People act through the artefacts they use to perform the activity. Hence the activity is shaped by the artefacts. They may make some aspects of the activity more salient than others. For example, a sophisticated word processor may encourage its users to format their documents rather than spend time editing the content of the documents. Kuutti (1996) cites Cole and Engeström (1991) who claim that objects are seen and manipulated "within the limitations set by the instrument". To understand these limitations, the history and development of the artefact must be understood, just as the history and development of the object of the activity should be understood. Kaptelinin (1996) suggests that the tool mediation perspective has serious consequences for HCI. He suggests that the field should not be named HCI, since it places emphasis on the interaction between humans and computers, which is secondary to the task of actually getting work done. Instead, Kaptelinin suggests that the field be named "computer mediated activity" or at the very least, that HCI should be concerned with the study of computer mediated activity.

The above discussion has presented the main fundamental concepts of activity theory. To complete the description of activity theory, the structure of an activity needs to be described. Activities are fairly complex structures which, like their objects, can change and evolve over time, depending on certain conditions.

Activities are long–term processes which require a number of shorter term processes to be carried out in order that the object of the activity can be met succesfully. These shorter term processes are called *actions*. Actions are goal directed processes. People are consciously aware of the actions they are performing and are aware of the short term goal that the action aims to achieve. Actions cannot be understood on their own, however. They can only be interpreted with respect to the whole activity. Thus the same action can take place in different activities and hence mean different things. Kuutti (1996) gives an example where "the action of reporting on the progress of a project will have a different connotation if it belongs to the activity of competing for promotion – even if the action and its other ingredients are exactly the same."

The way actions themselves are carried out is also important to understand. According to activity theory, actions are performed via a series of *operations*. Operations are routinized and performed unconsciously. In effect, operations are routines that are utilised by people in response to current conditions in which people are working. If the goal of an action remains the same but the conditions under which it is being performed change, then, activity theory claims, only the operational aspects of the action will change.

For example, consider the different activities, actions and operations that are involved in driving a van to collect building supplies. When building a house, a builder needs to drive a van between the building site and the builders' yard, to pick up supplies. Driving the van consitutes an action within the activity of building the house (other actions will involve laying the foundations, installing the plumbing etc). The operational aspects of driving the van include, changing the gears, applying the brakes, accelerating, etc. An experienced driver will not have to think about how to change gear. Changing gear is a routinised process for an experienced driver, which is triggered by certain conditions such as the revs of the engine being above a certain level or when decelerating to turn a corner. For a learner driver however, these routines may not yet have been operationalised. They must think consciously about how to change gear; depress the clutch, move the gear stick, release the clutch. After a while, changing gear will become routine, and the learner can add changing gears to their repertoire of operations. Once an action is operationalised however, it does not always remain an operation. Depending on the conditions that exist, an operation may become an action again. For example, if in the process of changing gears the driver hears a loud crunching noise and cannot get the vehicle into gear, they will tend to think consciously about the whole operation of changing gear, to try to work out what went wrong. They will think consciously again about depressing the clutch, moving the gear stick and releasing the clutch. Thus, depending on the conditions, actions can become operations and vice versa.

This concludes the description of activity theory. Only the most fundamental concepts have been described, those that were introduced in the book *Context and Consciousness* (Nardi, 1996). The theory has been developed further by other researchers, such as Cole and Engeström who further developed the social aspects of the theory by introducing the idea of a relationship between a person and their community, mediated via social rules. However, the description of activity theory given above is the most commonly agreed–upon definition. It is the one that has been used most in real projects, such as those described below. Some of these projects are now described, followed by a discussion of some of the issues raised by activity theory.

2.5.2 Using Technology to Reform Education

Bellamy (1996) describes two projects in which she was involved. The projects involved designing two pieces of software to be used in a classroom of 12 year old children. The first project involved designing software to educate students about paleontology. The second project designed software that was used in educating students about succesful communication and interpretation of ideas.

Bellamy saw the two projects as an opportunity to answer a number of questions relating to the ways that technology reforms education. How does technology change education? In what ways should it change education? How can we influence the change? She felt that the answers to these questions best lay in the application of activity theory to both projects, since activity theory places a large emphasis on history and development and on how activities evolve over time. She saw this emphasis on evolution as being directly relevant to answering questions relating to change.

Bellamy utilised Cole and Engeström's extension of activity theory to identify the ways in which technology should and can affect educational reform. Cole and Engeström added the notion of a community, its social rules and the division of labour to the fundamental concepts of activity theory described above. Cole and Engeström make the point that people do not typically perform activities on their own, and hence we must understand the broader community in which they belong to understand the activity. We must then understand the rules that the community deem appropriate, which determines what behaviour is appropriate. We must also understand the particular division of labour created by the community, since an activity may be accomplished by more than one person. We need to know how the accomplishment of that activity is shared across the community.

Utilising this framework of activity theory along with Vygotsky's particular ideas about development, Bellamy identified three principles for the design of educational technologies. These are, (taken from Bellamy, 1996)

- Authentic activities: Children should have access to, and participate in, similar cultural activities to those of adults and should be using age–appropriate tools and artifacts modeled on those used by adults;

- Construction: Children should be constructing artifacts and sharing them with their community;

- Collaboration: Educational environments should involve collaboration between experts and students and between individual learners and fellow learners.

Employing these principles, Bellamy designed and implemented two computer–based tools for the respective projects (paleontology and communications). Both pieces of software involved the children performing activities that were likely to be performed by experts in the domains involved. For example, the paleontology project involved students using a computer–based simulation to investigate rocks and fossils. Each involved construction of various artefacts particular to the domain (such as a newspaper article). The different artefacts were constructed via a collaborative process, with groups of students. Bellamy reports that the education process was fundamentally changed by the introduction of these pieces of technology. Students' learning was much more pro–active, with the teacher taking the role of facilitator, rather than instructor. The opportunity to construct artefacts was seized upon by students, resulting in them taking ownership of their work and becoming inventive and enthusiastic about the work. The software provided realistic examples of work in real life which enthused students. They felt they were doing something real, something that had a relation to real life. And finally, by encouraging collaboration, and the sharing of work, students were encouraged to ensure that they produced good work, so that they would not look foolish or say anything stupid. They wanted to ensure that they had something interesting to say and discuss with other students. The collaboration also increased students' understanding of the topics, since students felt free to discuss other work.

Bellamy notes that these positive changes may not be applicable across all areas of education. She states that, in accordance with Cole and Engeström's version of activity theory, which encourages consideration of the wider community, design should not just look at the immediate situation, but the whole picture. Thus, the school teachers, the school board, the head teacher etc. all take part in the educational process and should be considered when designing technology for change. On a practical level, technical support should be designed also, since the lack of such support is likely to have a detrimental effect on the take–up of the new technology. On the whole however, Bellamy presents a positive example of the way in which activity theory can influence the design of artifacts.

2.5.3 Applying Activity Theory to Video Analysis

Bødker (1996) describes how the concepts and constructs of activity theory were used to analyse video recordings of users using a computer system. In her analysis she particularly focuses on *breakdowns* and *focus shifts*, two specialised aspects of activity theory not covered in the general description given above.

Breakdowns are situations in which an artefact does not behave in the way that it is expected to. This can occur for a number of reasons. The user of the tool may be interrupted, the tool may behave differently than was anticipated causing different or inappropriate operations to be triggered and executed. Bødker particularly focuses on breakdowns that are somehow caused by the computer application.

Focus shifts are shifts of attention from one object to another. These shifts are more deliberate than breakdowns. For example, a teacher can shift focus from the subject they are teaching to technology designed to support that education. In this situation, the teachers focus is now on the technology and so what were initially operations (as the teacher was using the technology to teach) now become actions, since the teacher wants to explain how the technology works.

Bødker has characterised focus shifts depending on the particular characteristics of the computer application being studied (Bødker 1991). She distinguishes between

- The *physical aspects* – support for operations toward the computer application as a physical object.
- The *handling aspects* – support for operations toward the computer application.
- The *subject/object directed aspects* – the conditions for operations directed toward objects or subjects that we deal with "in" the artefact or through the artefact.

Using these constructs and others, Bødker analysed video tapes of users using a computer system for the Danish National Labour Inspection Service (NLIS). She was able to look at the video clips and describe the problems in terms of the various concepts offered by activity theory. Having categorised the problems, she was then able to design solutions. One of the problems concerned inserting page numbers in reports produced by staff at the NLIS. Bødker suggests that the handling aspects of the computer application prevented the users getting back to their real task, that of writing the report. Generalising from this example, Bødker claims that "it is possible to distinguish 'everyday fluent conduct" from more exotic breakdowns; handling aspects must be designed to support these in different ways" (Bødker, 1996). Bødker highlights the similarity with this analysis to that she performed on the method for inserting footnotes in Microsoft Word (Bødker, 1991). The handling aspects of Microsoft Word in this instance are poor, since, by forcing the user to specify a numbering scheme each time they insert a footnote, the program interferes with the users real task of writing the footnote. Thus, claims Bødker, users may never be able to operationalise the action of inserting a footnote. In this case, Bødker suggests that the dialogue be re–ordered in such a way that it does not interfere with creating a footnote.

2.5.4 How well does Activity Theory support the five factors?

Activity theory offers a rich and well defined set of concepts and constructs with which to describe people's work and the way that tools mediate between people and their work. In contrast, traditional ethnographic descriptions of work (and even to a certain extent, Contextual Inquiry) offer no standard concepts or constructs. Instead, an ethnographic description of work is an ad–hoc, detailed description of a particular situation that can be very useful when designing for that particular situation, but which cannot be generalised to apply to other situations without difficulty. The benefits of employing standard concepts and constructs to analyses of contexts and designs have been shown by both examples described above. Both Bellamy and Bødker gained useful insights into their respective projects through applying Activity Theory. However, through the concepts that activity theory employs, designers gain a particular point of view on a design or context. This point of view makes certain aspects of the context salient and other aspects less so. As Cockton *et al.* (1995) argue, these different points of view present a particular bias, encouraging designers to focus on certain aspects of context at the expense of others. Cockton *et al.* (1995) argue that designers should be aware of these differing biases and should have the flexibility to employ different approaches so that they can concentrate on all the relevant aspects of context, rather than those aspects that a particular method makes clear.

Scope

Since activity theory is a descriptive tool there is no need for it to define a *scope* over the contextual data that is to be collected. Activity theory is generally used *after* the data has been collected by some other method. It is this other method that will define a scope on the range of contextual information to be collected. It is true that, just as with Customer Centred Design, where the models that were used to model the contextual information also scoped the information (since only that data that could be modelled would be collected), the concepts and constructs employed by activity theory may influence the scope of contextual data that will be collected. However, this will only be true if it has been decided that activity theory will be used to analyse data before it has been collected, unlike Customer Centred Design in which it is known for certain that the various work, context, environment models will be used to model the data collected. Nardi (1996a) presents an example where activity theory was used to analyse some data that had been collected without any thought to activity theory. On the other hand Bødker (1991) describes a method that makes use of scenarios to collect data that will be analysed using activity theory. In particular, Bødker suggests that scenarios should be used to collect data about users' work because the tacit elements of work can be uncovered through scenarios. The focus on the tacit aspects of work comes from activity theory since one of its main constructs, operations, are actions that are performed unconsciously and are difficult for people to talk about.

Quality

Activity theory makes no claims about the *quality* of the data modelled, since other methods are used to collect the data. An analysis of poor quality contextual data using activity theory cannot improve the quality of that data, although it could perhaps identify ways in which the quality of the data could be improved by identifying aspects of the data that are missing and that do not fit into the activity theory constructs.

Understanding

Given good quality data however, the constructs can be useful in generating a shared *understanding*. Once a design team is familiar with the constructs of activity theory, there should be no major difficulties in applying and understanding the constructs across different analyses. Thus, an analysis of contextual data using activity theory has a better chance of meaning the same thing to a design group, if each member of the design team is familiar with the terminology used. Unlike ethnographic methods which introduce new terminology for every analysis, activity theory uses the same terminology across different analyses.

Linking

No support is offered for *linking* between different representations because activity theory only represents data concerning users' work. It does not represent designs and hence does not support explicit representation of the relationships between contextual information and design.

Activity theory is predominantly a descriptive tool, rather than a predictive one. It is difficult to see how generalisations could be made concerning the *types* of activities and the types of actions that would be required for each type of activity. If such generalisations were available, then it would perhaps be possible to use activity theory to determine what kinds of designs would fit in with particular activities. However, activity theory does not offer such constructs, perhaps because it may be impossible to define a range of types over all activities, since the factors that play a part in the formation of different activities are numerous (e.g. consciousness, environmental conditions, other people, history etc.).

Interestingly, Bellamy's use of activity theory to develop educational technology concurs with a top down approach. She used activity theory to identify three high level principles that educational software should satisfy. She then used these principles to guide the design of two different pieces of educational software. She does not describe the processes by which the details of the software were designed but reports that both teachers and students reacted positively to the software in terms of the three principles identified. This example shows that activity theory can be used successfully in a top down manner, to suggest a direction for a project, as well as to critique and describe projects.

Definition

In terms of *defining* relevant context, activity theory makes a convincing case for including information about the user's consciousness in an analysis of their work or context. Behavioural descriptions alone are not enough, since identical behaviour may be exhibited for completely different reasons. For example, a person may perform some procedure as an action, consciously aware of what they are doing. In contrast, they may perform the same procedure operationally, in which case they are not consciously aware of what they are doing, but instead are reacting to conditions in their environment. In traditional task analyses, both procedures may be viewed as one and the same. Activity theory suggests however that they are different, due to the different consciousness levels. According to activity theory, the particular level of consciousness of people while using tools to perform their work is an important consideration when designing new tools to support work. Designers should be aware of the different levels of consciousness and how best to support shifts between these levels, such that users can turn actions into operations, thereby increasing their skill levels.

In summary, activity theory offers a useful framework in which to analyse and describe users' work. It highlights the importance of considering higher level aspects of work, such as the users' overall object(ive) of activity and their differing levels of consciousness while performing the work. It stands in contrast to traditional task analytic approaches to studying user behaviour by saying that behavioural analyses are not sufficient to adequately describe users' work. Activity theory has been used successfully to derive specific guidelines or principles upon which designs can be judged, as in the project carried out by Bellamy. However, it would seem that activity theory has most to offer in its rich set of constructs and concepts, that designers can use to analyse and describe users' work. These descriptions can then be used appropriately in producing designs to support users in their work.

Summary

The table below (Table 5) summarises the above discussion. It rates the level of support provided by activity theory for each of the five factors (1 = poor support, 5 = excellent support. Areas left blank mean that there is no direct support for this factor.).

	Scope	Definition	Quality	Understanding	Linking
Activity Theory		3		3	

Table 5. Level of support provided by activity theory for the five factors

Since activity theory is a descriptive tool, designers are free to collect contextual data in whatever manner they choose. Bødker (1991) suggests that scenarios can be a useful means to collect data that is especially amenable to analysis through activity theory, since they can be used to collect and investigate the tacit aspects of users' work, those aspects of work that trigger different operations. However, scenarios have also been gaining prominence in HCI due to their multi–purpose nature. Scenarios have been used in HCI now for some time, to describe current work practices, to evaluate proposed designs, to envisage future work situations. They are an excellent communication tool, since their narrative structure is familiar to many people. The next section presents an overview of the state of the art of scenario–based design, as presented in the book 'Scenario Based Design' edited by John Carroll (1995).

2.6 Scenario–Based Design

Scenarios are multi–purpose, flexible tools that can be used at various different stages throughout a design project (Campbell, 1992; Hammond, 1991; Preece et. al, 1994). Before describing a number of different examples of the use of scenarios in design, some background material (Kuutti, 1995) is presented which describes two different types of scenarios and the justifications for each type of scenario. Each type of scenario can be used to describe contextual information at two different levels. Kuutti argues that both types of scenario, and therefore two different levels of descriptions of context, are necessary in making effective use of contextual information in design.

A popular approach to representing and communicating information about users' work is to use scenarios. A clear consensus on the definition of scenarios has still not been reached (Kuutti, 1995) but at least two attributes are commonly agreed upon. First, in terms of interactive systems design, a scenario describes a process or sequence of acts related to using an interactive system. And secondly, it describes these acts from the viewpoint of the user using the system.

A scenario can be viewed as a narrative description of a user using a system and can thus be used to represent users' work and to communicate these representations to other members of the design team. Scenarios can be created and communicated effectively due to people's familiarity with narrative descriptions. When communicating via scenarios, no formalisms need be defined or explained in order for the scenario to be understood. As long as the scenario is written clearly, in a language understood by all members of the design team, it can be used as a means of communication.

However, there are differing opinions about the scope of a scenario (Kuutti, 1995). To some, a scenario should describe exactly what a system does.

> "More specifically, a scenario is a description of one or more end–to–end transactions involving the required system and its environment" (Potts *et al.*, 1994)

To others, a scenario encompasses a much broader view of the context in which the user exists.

> "An important feature of a scenario is that it depicts activities in a full context, describing the social settings, resources, and goals of users. It is not narrowly focused task description, but the 'big picture' of how some particular kind of work gets done" (Nardi, 1992)

Kuutti reconciles the difference in opinion regarding the scope of a scenario by arguing that the definitions given above, define *rich scenarios* and *narrow scenarios*. Kuutti claims that there is a place for both kinds of scenarios. Rich scenarios, by representing the broader view of users' work, represent the why and what of users' work. They represent the reasons for users' work and describe what that work is. Narrow scenarios on the other hand represent how users' work is performed. They describe in detail what the users have to do to perform their work.

Kuutti's article describes the justifications for both rich and narrow scenarios in design. There now follows some examples of both rich and narrow scenarios in design. Kyng (1995) presents a method in which scenarios are essential tools for communicating between users and designers. Rosson and Carroll (1995) describe their method, in which the authors attempt to bridge the gap between specification of a design and its implementation by specifying *envisionment* scenarios (similar to rich scenarios) then elaborating these to *task* scenarios (similar to narrow scenarios). As the scenarios are elaborated, objects are identified that can be implemented in an object oriented system. The resulting objects are linked to the originating scenarios, which gives access to the reasoning behind the objects.

2.6.1 Creating Contexts For Design

Kyng describes a participatory design method that uses scenarios to ease communication between users and designers. Participatory design is a method of design that grew out of the trade union movement in Scandinavia. The unions were concerned that the design and introduction of computer technology in the workplace in Scandinavia would result in the loss of some jobs and the de–skilling of the workforce. To address this, they encouraged the involvement of users in the design of workplace computer technology. Instead of designers asking users what kind of system they wanted, and then going away for six months to develop the system, designers work in collaboration with users continuously. Together, users and designers produce initial specifications and prototypes of the system. Together, they evaluate them and suggest improvements. The process is usually an iterative one, at the end of which the designers deliver a system that the users have spent a great deal of time critiquing and improving. The end result is hopefully that users will accept the technology, for two reasons. The first is that it will do what the users want it to do, since they have been involved in its design right from the start. And secondly, because they have been involved in its design, users have some 'ownership' of the system, and are motivated to use it and make it work.[3]

Kyng's method follows four basic stages (see Figure 4). In the first stage, the initial study aims to educate the designers about the users through observations, interviews, work demonstrations and study of work materials. This also provides the opportunity for users to get to know the designers in an informal setting. They learn what the designers are doing and how the project will proceed. As a result of the initial study, a summary of the organisation and work performed within the organisation is produced. This summary is a narrative description, very much in the form of the rich scenarios that Kuutti describes. It gives some of the reasoning behind why some aspects of work are performed the way they are (e.g., documents are photocopied and hand delivered because there are no networking facilities for the existing computer set–up).

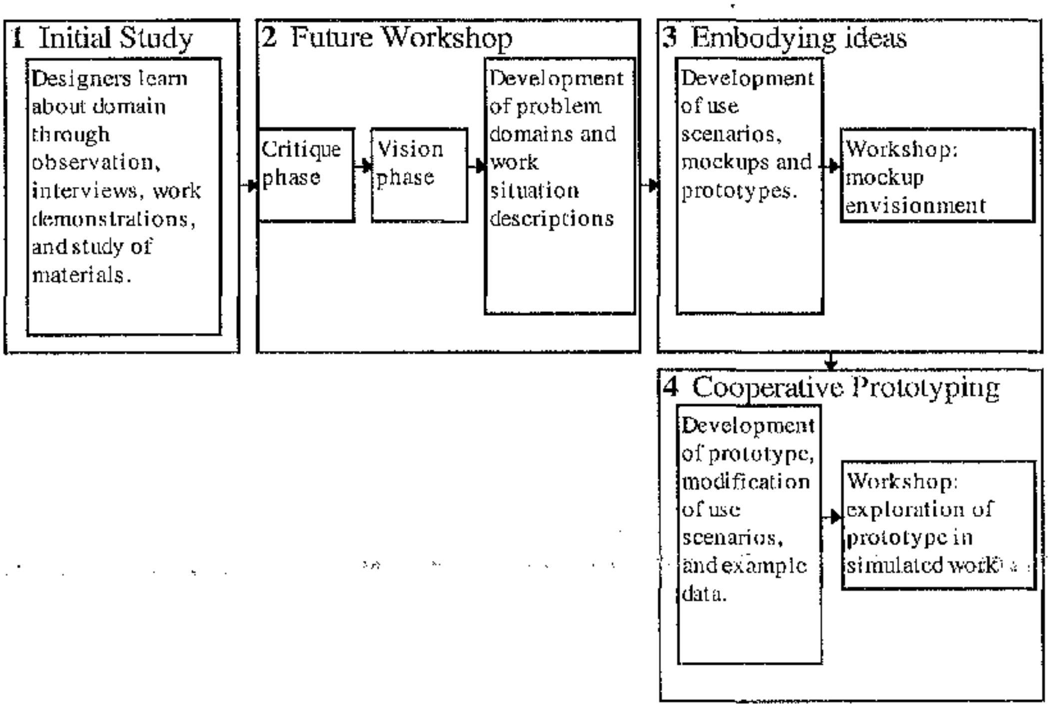


Figure 4 Kyng's four stage model of co–operative design.

The second stage involves creating problem domain and work situation descriptions. The descriptions are created by both designers and users. With respect to the users, the descriptions describe situations in their work that are relevant to them. For example, they may describe important parts of the work or they may describe parts of the work that

[3] Participatory design will be described in more detail further in this chapter.

constitute a bottleneck or introduce problems in other areas of the work. With respect to the designers, the descriptions describe situations in the users' work that are relevant to the designers, in that they describe situations in the work that could possibly be supported by technology. Separate descriptions are created for each relevant aspect of the work. Each description is a scenario, similar to the narrow scenarios described by Kuutti. Each description, however, can be tied back to the initial study description, giving access to the overall context in which the work situation exists, and hence some of the reasoning for this particular aspect of work being considered relevant.

The problem domain and work situation descriptions are used as input to the third stage, in which use scenarios, mockups and prototypes of the future system, designed to support the work described in the second stage, are produced. Use scenarios are developed in tandem with the mockups and prototypes and are intended to show how the future technology improves upon the current situation. In contrast to the second stage, where the descriptions described the current situation, the third stage focusses on the future situation and aims to discover if the proposed designs improve upon the current situation. However, while scenarios describe certain qualities of computer support, for example that users can inspect links connecting documents to see other relevant documents, they should not be viewed as describing requirements for the system. Rather, as Kyng points out, they "represent hypotheses to be evaluated through workshops". The use scenarios also do not describe the proposed computer technology in such detail as individual keystrokes required, or the physical appearance of the system. These descriptions are left to the mockups and prototypes. Thus the scenarios help explain the rationale for the prototypes and the prototypes themselves describe how the technology will work. The benefit of this approach is that the prototypes can be modified, in accordance with the evaluations by users. Depending on the materials used to build the prototypes, modifications can be far reaching and relatively easy to make (e.g., when paper and pen prototypes are created) or they can be at the surface level and slightly harder to change (when working prototypes of the system are used). At this stage, users experience the prototypes for the first time, and therefore have to spend some time learning about the limitations and possibilities inherent in the prototypes. In the fourth stage, they can use this knowledge gained to co–operatively prototype with the designers.

In the fourth stage, similar artefacts are used as in the third stage. The difference between this stage and the third stage is in the amount of participation that users can have in the process. Here, having learned about the prototype in the third stage, they can suggest modifications to the prototypes, mockups and use scenarios. It is essential at this stage that both users and designers collaborate. The users can provide knowledge about the demands that their work will place on the technology. And by using the technology, they can become aware of the technical issues involved in the design of the system. The designers input to the process is essential also, since, while they do not have expertise about the users' work, they do have expertise in design and can use that to suggest ideas, limitations etc. in the development of the scenarios, mockups and prototypes.

There are two further types of scenarios that are used by the designers both during and after the four stages described above. These are *exploration* and *explanation* scenarios. Exploration scenarios are used by designers to explore alternative methods of support for users' work. They are more detailed than use scenarios, which do not include any details about how a particular piece of technology will work, since they are intended to be used to discuss whether the particular technology supports the work. They are also more abstract than use scenarios as they don't depend on explicit references to specific situations or even specific workplaces. However, while exploration scenarios may simply appear to be detailed, but abstract, descriptions of task sequences, the benefit of writing them as scenarios and not as linear task sequences, is that it is still relatively straightforward to link them to the work situation that inspired them. Exploration scenarios are intended to provoke discussions amongst developers about the technology and its abilities to support the particular situation described in the scenario.

Finally, explanation scenarios are used to record the rationale behind the design of certain features in the computer technology. Thus hypotheses are preserved and may be used in the development of further use scenarios and workshops.

Kyng's method uses scenarios throughout the process. One of their main advantages according to Kyng is that users do not have to be taught a design notation or structure in order to understand the results of each stage. Narrative descriptions are a familiar device to both users and designers and hence scenarios are a tool which can be used to communicate effectively between both sets of people. The scenarios also encourage the creation of concrete situations in which to ground discussions. Except for the last two scenarios described (explanation and exploration scenarios), which are only used by designers, each scenario is aimed at a particular work situation. Therefore, for the users at least, the situations described in the scenarios are familiar and they can relate to them, making it more likely that they will contribute to the discussions about the scenario. However, Kyng suggests that this reliance on the concrete situation implies that only real users can be involved in the process, since only they know and can relate to the details described in the scenarios. Facilitators, such as managers, marketing people or ethnographers cannot be used because they do not have the necessary insight to enable them to make valuable contributions to the discussions. Likewise, developers need to be involved from the start of the project through to its completion in order to make the best contributions to the group. Without this amount of involvement, they cannot be expected to have gained a good understanding of the users' work and how the technology will impact upon it.

Kyng has described a participatory design method that makes use of scenarios throughout. Both rich and narrow type scenarios were used throughout the method, but the method tended to blur the distinction between the two, not separating discussions about what the tool should support with how it should support it. The next example of using scenarios in design makes that distinction by using two separate tools for each type of scenario. It also describes the use of the scenarios in a participatory design setting.

2.6.2 Narrowing the Specification–Implementation Gap

Rosson and Carroll (1995) describe the Scenario Browser, a system implemented in Smalltalk/V which integrates task scenarios with software implementations of those scenarios (see Figure 5 and 6). A task scenario describes a task that a user may typically perform, with or without a computer. It describes the actions a user has to take to perform the task. The authors contrast task scenarios with traditional functional specifications of systems which list all the functions that a system should support but does not describe the ways in which these functions will be used. A task scenario describes the ways in which individual functions are composed and used by users to enable them to complete their tasks. Thus, a collection of task scenarios will describe all of the individual functions that a system should support, but each will be described in a particular scenario which gives the designers useful contextual information about each function.

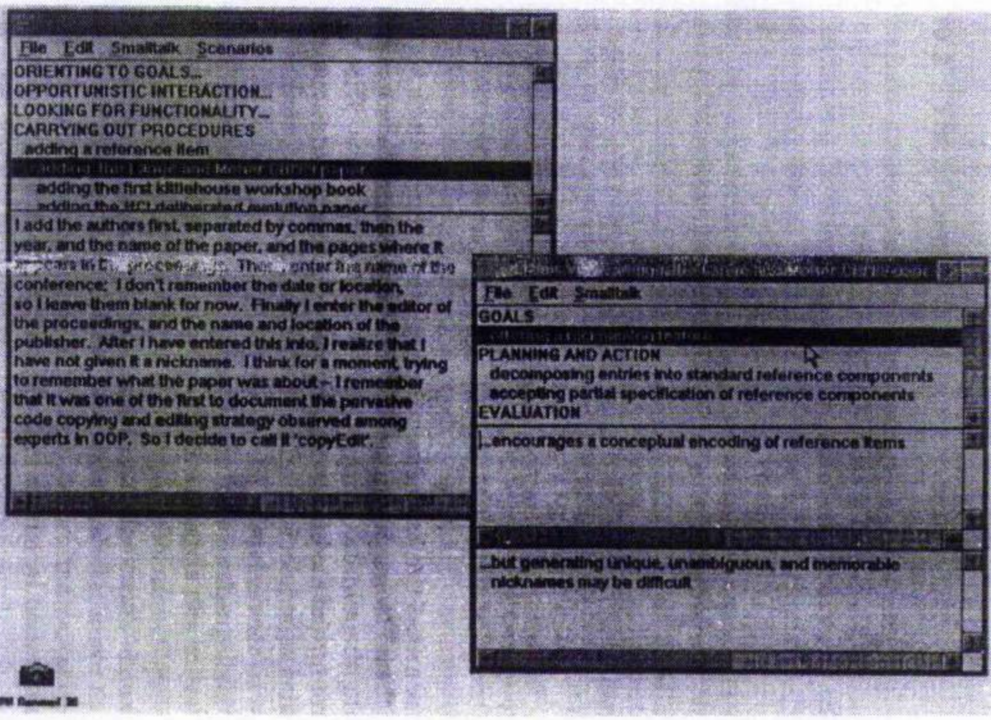


Figure 5. Scenario Browser and Claims View (Rosson and Carroll, 1995)

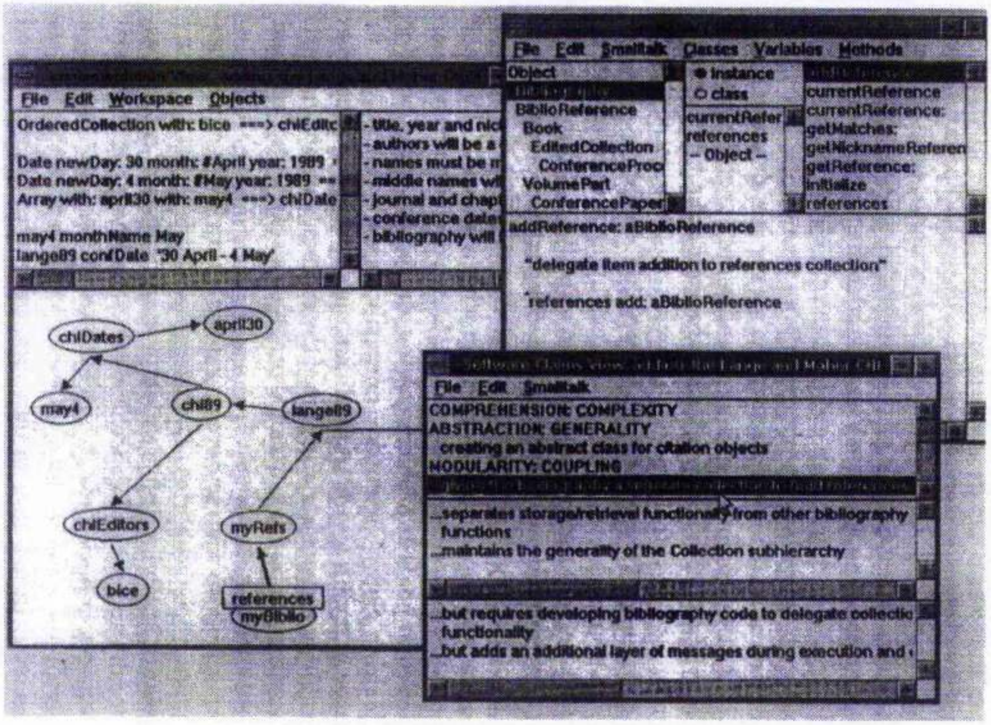


Figure 6 Implementation and Claims views (Rosson and Carroll, 1995)

At the start of a project, the authors suggest that it is useful to create a set of *envisionment scenarios*. Envisionment scenarios are like task scenarios, but they do not describe in detail how the user accesses functions and operations. Instead, they focus on what the user is aiming to achieve (e.g., inserting a reference into a bibliographic database) rather than on how (e.g., choosing the add reference menu option). In this way, designers can try to ensure that they are designing the correct functionality. Thus the early focus is on usefulness, rather than usability. Each envisionment scenario should describe one of the main tasks that users aim to achieve. Information about these tasks can be gained through methods which analyse current work practices. Once a set of scenarios have been developed that cover all tasks that users will typically want to perform (ascertained in conjunction with the users), each can be elaborated with details about how the users will perform the activities.

The scenarios evolve throughout the project, changing in reaction to certain design decisions and changes throughout the project. Some scenarios will undoubtedly be related, perhaps via reference to the same objects, or via the fact that one scenario served as the source to another. As changes are made, it is important to ensure that appropriate changes are propagated throughout the set of scenarios. The authors have attempted to achieve this via their use of *claims*.

Claims can be made for different aspects of a scenario. Each claim will have both positive and negative consequences. For example, when entering a reference into a bibliographic database, one of the features of the database system might be that it offers a nick–naming feature for each entry. This feature would be described in a scenario that describes a user adding a reference to the database. Usability claims can be made about the feature and tied to the scenario. The authors provide the following usability claims for the nickname feature (Table 6)

offering a reference nicknaming feature encourages a conceptual encoding of reference items
but generating unique, unambiguous, and memorable nicknames may be difficult

Table 6. An example claim from an 'add a reference' scenario

Claims can help identify further scenarios. The consequences described in a claim can become source material for a new scenario. Claims can also suggest links between scenarios, if the same claims are made for a particular feature across each scenario for example. Fundamentally, claims provide the reasoning behind a scenario and any objects within that scenario. By inspecting the claim, users can get an idea of the reasons for particular features and the issues, both positive and negative, that were considered when designing the feature. Thus, claims analysis ensures a deeper understanding of the objects within scenarios and their relationships. In the Scenario Browser system, links can be created between scenarios via claims. The system aids maintenance of the set of scenarios by, for example, identifying all those scenarios that are affected by a claim so that they can be updated if the claim is updated.

As scenarios and claims are created and evolve, the authors encourage identification of system objects within the scenarios that can be implemented as objects in an object oriented programming language, within their Scenario Browser system. So, in the bibliographic system described briefly above, designers may identify objects such as paper, journal, proceedings, authors etc. These should all be implemented as objects in any implementation of the scenarios. The Scenario Browser system supports the creation of objects from scenarios by allowing instances of these objects to be created in Smalltalk. Actual instances of objects are created, rather than abstract classes, since the authors believe that it is preferable to develop abstract problems using concrete, testable solutions and ideas. As scenarios are implemented, and objects created, claims for the implementation can be made in much the same way that claims were made for design features. Thus, particular features of the implementation can be described and reasoned about in the same way that features of the design are described and reasoned about through the claims analysis. These software claims can impact the scenario implementations (again, in the same way that the design claims impact the design scenarios). Lastly, any impact felt in the scenario implementation may be reflected in the scenario itself and vice versa.

In summary, the Scenario Browser allows information about users' tasks, in the form of user scenarios, to be linked to the implementation of those tasks, in the form of implementation scenarios. Software and usability claims provide the glue that sticks everything together as well as providing the rationale behind the decisions made. Interactions between task and implementation scenarios (and the corresponding sets of claims that go with each scenario) introduces new design ideas and new tasks which should be represented in their appropriate scenarios. Thus, the Scenario Browser presents a way in which scenarios can be used pro-actively in design.

2.6.3 Design Space Analysis and QOC

Claims analysis is very similar to design space analysis. Preece *et al.*. (1994) describe the notion of a design space and design space analysis thus:

> "Design can be viewed as an exploration of a space of alternatives. That is, there are a host of alternative designs that fulfill the system's specification and the process of design involves identifying the one, or ones, that satisfy the system's constraints and goals as closely as possible." (Preece *et al.*, 1994, p. 528)

Claims analysis could be used to explore the space of alternative designs, since a claim for a design feature may suggest a better design feature that suffers fewer negative consequences than the original feature. Simply exploring the space of alternative design decisions with paper and pen (and a great deal of thought) can be ineffective since there is a danger that designers will not consider as much of the design space as possible by neglecting to consider alternative design ideas and ways to judge them. The claims analysis technique of Rosson and Carroll is one attempt to be more systematic. Another is through the use of the QOC notation, developed by MacLean *et al.*, (1991). The authors claim that this notation can be used to explore the design space efficiently and comprehensively. QOC is used to specify the design space of an interactive artefact so that all the stakeholders of that design can gain a better understanding of it. It helps designers since it forces them to discuss explicitly the advantages and disadvantages of various design decisions by creating a QOC diagram such as the one below (Figure 7 – taken from MacLean *et al.*, 1991; p. 209).

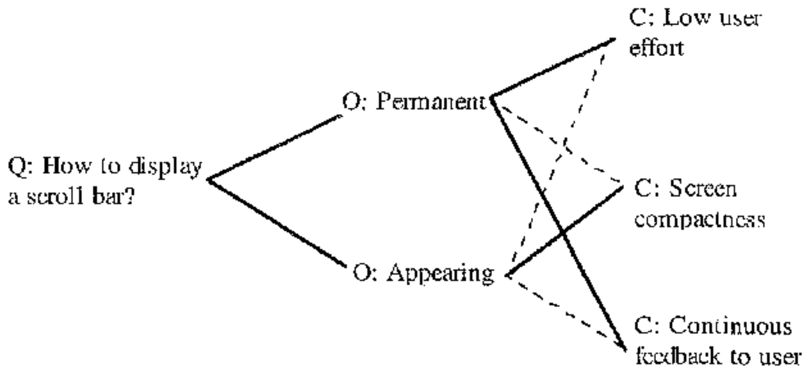


Figure 7. An example QOC diagram (taken from Maclean *et al.*, 1991)

This diagram shows the questions, options and criteria considered by some designers when deciding how a scroll bar should be displayed in a window. The designers have decided on two options, either the scroll bar is displayed permanently or it appears whenever the user moves the cursor over a certain area on the screen. The designers have also identified three different criteria to help them choose between the alternatives; low user effort, screen compactness and continuous feedback to user. In the diagram, solid lines represent positive relationships between the option and the criteria while dashed lines represent negative relationships. So displaying the scroll bar permanently is good for user effort and feedback but not for ease of screen compactness. Having identified the options and the relationships between the options, and the criteria used to judge them, the designers then settle on one of the options. It is not a simple case of subtracting the number of dashed lines from the number of solid lines for each option and then settling for the option with the higher score. Certain criteria may be judged more important than others and so will weigh more heavily when it comes to making the actual decision.

The QOC notation is therefore useful when deciding on alternative design decisions. It makes decisions explicit and helps record the reasons for making those decisions which is useful for evaluation purposes. It also helps the designers consider more alternatives since the notation actively encourages them to do so.

QOC, and, in general, exploration of the design space, has been identified as a useful practice (see Bellotti, 1993; Buckingham Shum, 1995; Conklin and Begeman, 1988; Lee, 1990; MacLean and McKerlie, 1995; Moran and Carroll, 1995). It makes designers aware of other alternative designs and forces them to be explicit about the reasons for decisions. Design space analysis can therefore help designers make more efficient use of contextual information, since it provides another way in which this information can be used. It can be used to suggest design alternatives and to judge between them. Thus, designers can use contextual information to suggest better designs *and* to increase the understanding of a design (through using the context to identify criteria with which to judge individual design ideas or options). Judging individual design ideas, through criteria based upon contextual information, helps to identify how well a particular design idea will fit its context. As in the method described by Rosson and Carroll (in which claims are linked to scenarios), QOC diagrams can be linked to scenarios and other design specification notations. MacLean and McKerlie (1995) describe how they linked QOC specifications to scenarios. The QOC representations were used to coordinate and complement the concrete scenarios. The QOCs were used to make the distinctions between and within scenarios clear and to provide justification for each scenario. Johnson (1996) describes a technique for linking QOC and formal specifications of a system together, a technique which again uses the QOC notation to justify and explain a complementary design representation, in this case, a formal specification.

Other techniques exist which use scenarios in a similar way to Rosson and Carroll's method. Interested readers should consult Robertson (1995), Jacobson (1995) or Wirfs–Brock (1995) for more information on how scenarios can be used to identify objects to be implemented in an object oriented system. Crucially however, each differs from Rosson and Carroll (1995) in that none provide means to link between scenarios and/or objects via explicit rationales.

2.6.4 How well do scenarios support the five factors?

Scenarios are undoubtedly a useful and efficient means of communication between users and designers and between designers themselves about particular aspects of a design or users' work. The narrative form of scenarios means that they are familiar to everyone who has ever told a story to other people or heard a story being told. People do not have to learn new formalisms or new structures in order to create a scenario. Scenarios are informal structures that can be created and modified by anyone and, hopefully[4], understood by all those involved. As the above discussion has shown, a further benefit of scenarios is their versatility. They can be used to investigate a current situation, to envisage a future situation or to evaluate a proposed situation. They can also be used in projects lasting days to projects lasting years.

[4] Of course, the interpretive quality of a scenario depends on the reader understanding what the author of the scenario is trying to say or describe.

Quality

However, there are a number of problems with scenarios. The first concerns the *quality* of the data modelled. As with activity theory, scenarios can be used to model data that has already been collected by some other method. In this case the quality of the data is determined by the method that was used to collect the data. However, scenarios can also be used to collect data (Bødker, 1991). In this case, difficulties arise since there is no way of knowing what kinds of scenarios to create, such that all the important aspects of users' work will be uncovered. Some brief guidelines exist in these cases. For example, Carroll (1992) suggests that a dozen scenarios are generally enough to cover all aspects of users' work, but at what level should the scenarios be written? If they are written at keystroke level, or if they are written at the level of user goals, will a dozen scenarios cover all the important aspects?

Kyng attempts to ensure quality data by using a variety of methods to learn about users and their work right at the project outset. He uses interviews, observations and a typically ethnographic approach to data collection. He is further advantaged in his method by the fact that it entails constant communication and interaction between users and designers. So, if the designers were to create a scenario that did not genuinely reflect users' work, users would inform the designers to suggest ways in which the scenario could be modified. However, as Nardi (1995) points out, this kind of access to users is a very ideal situation, and one that would be difficult to replicate in other circumstances.

> "The reality in most countries and companies is that of a complete lack of the infrastructure and philosophy necessary to support Scandinavian style cooperative design. The time and money involved across designers, managers and users is far beyond what can be expended in ordinary industrial projects." (Nardi, 1995)

Scope

The *scope* of contextual data that can be represented using scenarios is particular to the way that the scenarios are used. Rosson and Carroll (1995) focus exclusively on using scenarios to identify objects to be implemented in an object oriented design. Hence the scope includes task specific objects and relationships between objects but may focus less on higher level issues such as the environment in which the design will be placed.

Definition

Similarly, *definitions* of context are particular to the methods that use the scenarios. If a scenario is being used to identify relevant objects that should be implemented in a computer based tool, the definition of relevant context will likely focus on objects and their interactions in the work place. Scenarios alone rarely define the relevant contextual information for a project.

Understanding

One benefit ascribed to scenarios is their familiar narrative structure (Erickson, 1995). Scenarios, it is claimed, are constructed in a style and language that is familiar to most people, making it easier to understand what the scenario is describing. However, Kyng claims that the scenarios, work situation descriptions, initial studies etc., cannot be used to help explain the work or the design to others not involved personally in the analysis of users' work and the creation of these artefacts. "Without such personal access to the reality behind the design artefacts it is not possible to go beyond the understanding already achieved in the artefacts." (Kyng, 1995). Hence, Kyng appears to claim that scenarios and scenario type representations as used in co-operative design projects, are not capable of explaining the context of users' work to others.

Linking

In suggesting a means by which scenarios can be used to better explain the context of users' work to others, Kyng (1995) identifies the need to *link* the scenario to the original data that was the source for the scenario. Linking in this way allows some form of evaluation to be made, and allows extra contextual information to be added to the whole description of users' work. Without this extra information, it is doubtful whether scenarios alone could be used to describe a complete work setting or design. For example, there are certain relevant aspects of users' work that may not fit easily into a scenario such as, information about the users' age or physical capabilities. Information about work or users that is common across all scenarios may be left out, since each scenario addresses details of a particular situation and the aspects of that situation that make it relevant and worthy of investigation.

Care must also be taken when creating scenarios, not to write them in a biased or disrespectful way. For example, it may not seem appropriate to describe the users' cognitive abilities while creating a scenario that describes them performing a task. Without taking care, it would be easy to include this kind of information in a scenario in a way that biases the design inappropriately. Care must be taken not to judge users through a scenario and not to prejudice certain design decisions.

Due to the specificity of scenarios, the *quality* and *scope* of data modelled by scenarios may suffer when systems are described using scenarios that are very general in nature or have very many differerent uses. Nardi discusses the problems of using scenarios to describe multi purpose software, such as a word processor or spreadsheet. In both cases, there are many different ways in which both pieces of software can be put to use. In particular, Nardi poses the problem of devising a set of scenarios that describe fully the many ways that users can make use of Microsoft Word. Because there are so many ways that general purpose software can be used, attempting to analyse and list them all in the form of scenarios would be a demanding and error prone process.

> "A short, crisp depiction (or even a set of them) does not always substitute for a full blown task analysis, or an ethnographic description of the complexities of the workplace, or a carefully researched list of user requirements, or an exhaustive set of specifications, except perhaps in the case where designers are working so closely with the prospective users that gaps in the scenarios are made up in everyday interaction." (Nardi, 1995)

Finally, even though attempts have been made to use scenarios throughout development, none (Carroll, 1995) describe ways of using scenarios other than to identify objects that can be implemented in an object oriented system. The *links* from the scenarios to the objects are an excellent way to tie the implementation of a system to the data that inspired and justifies it, but none of the methods described demonstrate how scenarios could be tied to lower level implementation issues such as command names, command sequences, task support, screen and icon design. Presumably such links would offer the same benefits to maintenance of the higher level aspects of a system as they do to the lower level aspects, namely that the links help to maintain changes in the system, identifying all those aspects of a system that are affected by a change. Focussing on the design and how it relates to the scenarios also helps identify any omissions in scenarios. Presumably, supporting linking between higher level aspects of a design would encourage identification of other omissions in scenarios, as well as suggesting completely new scenarios.

In summary, scenarios are an excellent communication tool due to their familiarity and ease of comprehension and creation, but they cannot be used to describe a full design or to depict all the details of users' work on their own. Attention must be paid to Kyng's point that people not involved in analysis of work or the creation of the scenarios lack the background knowledge and insight to fully understand the scenarios. In effect, scenarios, like ethnographic descriptions, are ad–hoc analyses of users' work. They must be backed up with other data that can be used to fill in the gaps and to justify what is said in a scenario. They can be put to good use in representing the rationale behind a particular object oriented design, and have the potential to be useful in linking between higher level aspects of interactive systems and their rationale also.

Summary

The table below (Table 7) summarises the above discussion. It rates the level of support provided by scenario based design for each of the five factors (1 = poor support, 5 = excellent support. Areas left blank mean that there is no direct support for this factor.).

	Scope	Definition	Quality	Understanding	Linking
Scenario Based Design	3		3	3	2

Table 7. Level of support provided by scenario based design for the five factors

The paper by Kyng introduced participatory design techniques. The next section now presents an overview of participatory design. For a detailed description of particular participatory design experiences, readers are directed towards Greenbaum and Kyng (1992), Bjerknes *et al.*, (1987), Schuler and Namioka (1993), Floyd *et al.* (1989) and Ehn (1989).

2.7 Participatory Design

Participatory design originated from Scandinavia in tandem with union concerns about the deskilling of workers and the loss of jobs through computerisation of the work place. It aims to bring users and developers together to work jointly on the design of a system. The main idea behind bringing the two groups together is to empower the users to make valuable contributions to the design of a system that will impact their lives and the way that they work. Participatory design recognises that users are experts on what they do and that designers can never be, without becoming workers themselves. Therefore, the users must play an important role in design.

Participatory design is relevant to an investigation of the requirements for a method that makes effective use of contextual information in design, since it suggests that the important factor in how effectively contextual information is used is the richness of the contact that designers have with users, not the different tools or representations of context or design employed by a method. Two methods, Contextual Inquiry and ethnography, have been described which both include significant interaction with prospective users. However, this interaction served simply to collect information that the designers then used separately to develop designs. In a participatory design project, interaction between designers and users continues throughout the whole design project. Both users and designers are involved in collecting relevant contextual information and using that information in design. Participatory design projects do not place a large emphasis on tools or representations, since designers have ready access to users to ask them questions about their work, and users have ready access to the designers to ask them about design problems. While tools and various representation techniques are used, there is theoretically less need to communicate solely via different tools and representations since users and designers can easily talk to one another. Hence an investigation of participatory design projects investigates how effectively contextual information can be used in design if that information is primarily represented and communicated via the users themselves and not through some other representational medium.

Clement and Van den Besselar (1993) have produced a summary of a number of participatory design projects that have taken place since the 1970s. Their summary provides a detailed description of the important issues that people should be aware of when considering setting up a participatory design exercise.

The first issue concerns the setting of the project and the motivation of those involved. Many of the projects that Clement and Van den Besselar reviewed suffered from lack of motivation from participants. Interestingly, in many of the Scandinavian projects the unions were unwilling to fully participate in the projects, even though the project aims were the same as that of the unions, namely to improve the quality of life of the workers. The authors cite one of the main researchers in one of the Scandinavian projects as reporting that there were "differences in political perspectives" with the central unions. Other issues relating to the setting included available resources, such as time and money. In many projects, the designers (who were often also researchers) provided much of the money to enable the projects to happen. In one project, the researchers provided money to hire temporary workers to do the work that the workers involved in the participatory design project would have had to do. Therefore, motivation, time and money as well as some other resources such as availability of space and prototyping tools, must all be available for participatory design to proceed smoothly.

If the proper resources are available for a participatory design project to proceed, the next issue involves ensuring that the correct participation takes place. Inevitably, the bulk of the project involves designing and improving upon previous designs through iteratively evaluating and modifying designs. Therefore, the designers are likely to take a lead role in specifying what is achievable with current technology and educating users about what is and is not possible. Given this situation, it is difficult to ensure that designers and users do not regress into the stereotypical roles of user as passive receiver of designs and designer as active generator of designs. Some researchers found it difficult to give equal responsibility for the design to the users, feeling that they (the researchers) knew best. This was reflected in some of the comments of users who said that they recognised that the researchers were experts in design and that they were not, therefore how could the users question any of the design decisions or suggestions that the designers made? All of the projects used a variety of methods to try to ensure equal participation in the process, but the success of the process was often down to "a strong political focus on participation, communication and learning" (Clement and Van den Besselar, 1993).

The last issue relates to what happens when the project is completed and the participants split up. Many projects reported that the direct involvement of users meant that they were able and willing to take the initiative with respect to using new technology after the completion of the project. Users gained competence in evaluating and using technology. "Users moved from their traditional passive roles into analyzing, designing and evaluating roles" (Clement and Van den Besselar, 1993). However, this does not always lead to an increased acceptance of new technology by users. Rather, it makes any inadequacies in the technology more apparent to users, since they are now more skilled in evaluating technology.

Most disappointingly however, active user involvement in the maintenance and continual improvement of a system ceased after the project was completed and the researchers or designers moved on to other projects. Only when the researchers kept in touch with the users did active involvement of the users in the maintenance of the system continue. Indeed, in one case the whole system that was developed by users and developers was scrapped soon after project completion (although extenuating circumstances were partly to blame for this).

In another project, traditional management roles and development procedures were reinstated almost immediately after the completion of the project. Given the close involvement that designers and users have during a participative design project, it is perhaps not surprising that such situations exist. In traditional software engineering, one of the golden rules when creating a software team is that if one person on the team is indispensable, they should be replaced as soon as possible. In other words, no software team should depend on one person or group of people for a large part of its success, since if anything should happen to that person or group, then the project is not likely to succeed. This rule would seem to explain the disappointing results after completion of a participatory design project. Since users and designers work so closely with one another and recognise that each depends on the other (designers depend on users for their expert knowledge about users' work, users depend on designers for their design expertise) it is very likely that the project will only succeed while both parties have access to one another.

This has severe implications for making effective use of contextual information in design. The context of use of a product is dynamic and constantly evolving. In order to remain useful and usable, a product must be updated and modified as the context in which it is used is changed. Using contextual information effectively involves using that contextual information in maintaining a product as well as in designing it. By making the influence of context over the design unclear or only known to the designers, it is unlikely that changes in the context of use will be effectively used to maintain the product. Hence some form of constantly available representational medium that demonstrates the influence that contextual information has had over a product or system is required.

2.7.1 How well does participatory design support the five factors?

Much of the process of understanding the data, defining relevant data, ensuring high quality data all emerges as a result of the partnership between the users and the designers. This will be expanded upon later, but there are some other issues relating to participatory design that should also be examined.

Clement and Van den Besselar's article gives a good summary of the main issues involved in participatory design projects. The main thing to be learned is that participatory design is very difficult to do well. The users must belong to an organisation that firmly believes in the principles embodied by participatory design and must give full support to user participation. Resources such as prototyping tools, adequate space, money, time, flexibility and co-operative management must exist, otherwise the team will not have the necessary equipment to enable them to participate fully in the project. To a certain extent, the availability of resources influences the *scope* of contextual information that will influence the design. If there are no resources available for members of the group to discuss and utilise certain aspects of contextual information then these aspects will have little or no influence over the final design.

As Nardi (1995) points out many companies and institutions do not have the kind of resources that enable successful participatory design projects to happen. This is unlikely to change, although more companies are beginning to recognise the importance of generating some feedback from users about the computer systems they are to use. However, it is doubtful whether most companies would allow groups of employees the time to involve themselves in a participatory design project.

One of the major problems with participatory design projects though, concerns the continued maintenance of the system. This is especially relevant to using contextual information in design since making effective use of contextual information involves making use of context to maintain a system as well as to design it. This is difficult in participatory design projects, however, unless the context and design descriptions have been fully recorded together with the relationships between the context and the design. Without this information, successful maintenance of the product depends on the availability of both the original designers and the users. This is difficult to achieve though. Once the project is complete, half the team (the designers) that were involved in the project are typically no longer accessible to provide answers to questions about the maintenance of the system. While users have learnt about design through participating in the project, it is doubtful that they will have developed adequate skills to be able to take charge of the maintenance. This problem is further compounded by the fact that users and developers will rarely work together at all times in a participatory design project. While developers are developing software, they do not have the time to educate the users in software development. Therefore as implementation decisions are made during development, it is likely that users will be left out of these decisions. However, even if users do have the skills necessary to maintain the system, it is doubtful whether they will have access to all the knowledge and rationale behind a system to maintain it consistently. Many participatory design projects generate *ad hoc* descriptions of users' work in the form of scenarios, ethnographic descriptions, interview transcripts. There is no structure to the information that went into inspiring the design. Often, much of the rationale for a design may simply be stored in the heads of those involved in the design. Since members of a participatory design team interact closely each day it is easy for them to ask questions about the design as it proceeds. Through this process, that information stays fresh, but once the project is complete and users have to get on with their work and designers move on to other projects, participants start to forget details of the design. Maintenance of the system then becomes difficult because the rationale behind parts of the design become unclear. Hence, to enable effective maintenance of a system, maintainers require access to the context and design descriptions and the relationships between both descriptions.

Definition, scope, understanding and quality

By its nature, participatory design does not suggest ways in which to *define*, *scope* or create a shared *understanding* of context. Rather, these factors develop and evolve as the PD group works together and the designers learn about the users' work. The idea behind participatory design is that the users are the experts in defining context and its scope, and that by working together with users, designers will be able to understand relevant aspects of the users' work. Due to the close interaction between users and designers, the quality of the data collected is likely to be high, since users will always have the opportunity to correct any misunderstandings or mistakes, and designers will always have the opportunity to ask questions.

Participatory design encourages consideration of all aspects of the work, such as how it fits into the organisation as a whole, why it must be carried out and why it must be carried out in the way it is. Therefore, even though participatory design is more a philosophy of design rather than a method and hence does not offer any concrete guidelines, steps to follow or definitions to use, the principles embodied by participatory design imply that designers and users must look beyond the immediate context of their work to find the relevant context that impacts their work.

Summary

The table below (Table 8) summarises the above discussion. It rates the level of support provided by participatory design method for each of the five factors (1 = poor support, 5 = excellent support. Areas left blank mean that there is no direct support for this factor.).

	Scope	Definition	Quality	Understanding	Linking
Participatory Design	3	3	4	4	

Table 8. Level of support provided by participatory design for the five factors

2.8 Related Issues and Techniques

Before concluding and summarising this literature review it is worthwhile considering other related issues and techniques that exist in the software engineering literature. This section will briefly describe Soft Systems Methodology, a method for reasoning about and describing problems systematically. A description of SSADM (Structured Systems Analysis and Design Method), a method of software development widely used in industry, will then follow.

2.8.1 Soft Systems Methodology

All of the methods described in this chapter suggest that when designing a computer system, designers spend time collecting contextual information related to the computer system. Most then go on to say that designers should analyse the data and from this analysis, current problems and difficulties that users are experiencing will become apparent. The new, improved computer system, it is claimed, almost flows logically from this analysis.

However, this method of design is not always applicable. For example, designers at Intel knew little about any of the problems or difficulties that the users they were studying were experiencing. Therefore it was almost impossible to identify relevant contextual information to collect. In order to learn about users and the problems they were experiencing, participants in the ethnographic engineering project used ethnographic data collection methods to collect detailed information about the users' culture. Participants were then able to use the ethnographic data to learn about users and to work out what the main problems were, before even attempting to collect more detailed information about how to tackle some of the problems.

Unfortunately, as was described earlier, difficulties were experienced in communicating the results of the ethnographic analyses to other participants in the project such that they could use the data in the design of new products. The graphical models used to describe the data were difficult to understand, partly because the terminology and constructs used within the models were proprietary and unfamiliar to most people. A methodology that represents problem situations using standard and well-defined terminology and constructs may have helped the ethnographic engineering team to better describe and communicate their analyses. Soft Systems Methodology (Checkland and Scholes, 1990) is one such method that offers analysts a particular way to think about and represent problems such that they can be tackled effectively.

What is Soft Systems Methodology?

Crucial to Soft Systems Methodology (SSM) is the system concept. A system is a collection of entities that interact to exhibit emergent behaviours or properties and that has the ability to control its behaviour in order for self-preservation. An emergent behaviour or property is one that is exhibited purely through the interaction of separate entities and is not exhibited by any single entity. For example, a bicycle is composed of wheels, handlebar, brakes etc., but only exhibits the property of being able to transport someone when it is ridden and the rider uses the handlebars, wheels, brakes etc., in tandem with each other. The bicycle is capable of 'self-preservation' in that the rider (who is part of the system) can observe their surroundings and take appropriate action so as not to collide dangerously with any other objects.

SSM utilises the notion of systems in the way that it suggests we think about problems. It suggests that we reason about a problem by modelling possible solutions to the problem as systems and comparing these systems with the problem situation. Through this comparison, we are able to see the difference between the models we have produced and the real world problem situation. We can use these differences to further improve our understanding of the problem, improve the models we use to understand the model and eventually solve the problem. This method is known as systems thinking and is the foundation upon which SSM is based. However, SSM adds a number of different modelling techniques with which we can create and describe systems or models (holons in SSM terminology) to analyse the problem situation. There are two main types of analysis; cultural analysis and logic based analysis. Each different analysis uses different models. The next section describes the logic based models followed by cultural models.

Logic based models

In the logic-based analyses, we are seeking models that transform the problem situation into a more favourable state. Such transformations must be implementable given the current situation. In SSM, a transformation is defined via what is known as a 'Root definition' in which the major stakeholders are identified, the transformation process is described, environmental constraints are listed and, crucially, the point of view which makes the transformation described meaningful is described. This last point is crucial according to SSM since there can be many points of view on one problem and some transformation may make sense from one point of view but not from another. Each of the elements described above is captured neatly via the CATWOE mnemonic, shown below.

C	'customers'	the victims or beneficiaries of 'T'
A	'actors'	those who would do T
T	'transformation	the conversion of input to output

	process'	
W	'Weltanschauung'	the world view which makes the T meaningful in context
O	'owner(s)'	those who could stop T
E	'environmental constraints'	elements outside the system which it takes as given

Given a definition of a transformation in the form above, the next step is to model the transformation process, to show that it is achievable and to show how it is to be achieved. Since the transformation is a system, it should be made up of a number of sub-entities that interact to exhibit the behaviour necessary to perform the transformation. The sub-entities should also be capable of monitoring the system's activities such that it can adapt as necessary. A simple diagram showing the main steps involved in the transformation is often sufficient. One or more steps will involve monitoring the process and taking appropriate control action. The remaining steps will involve the work that is necessary to perform the actual transformation.

There could potentially be many applicable models which will perform the same transformation. In order to choose between these relevant models, SSM judges each in terms of its efficacy (does it work?), efficiency (the amount of output divided by amount of resources used) and effectiveness (is T meeting the longer term aim?).

Finally, the model is compared with the problem situation. There are many ways this can be achieved. Checkland (1981) suggests four different ways;

- Informal discussion;
- Formal questioning;
- Scenarios, based on 'operating' the models;
- Trying to model the real world in the same structure as the models

Cultural models

While the logic based models describe how a model or transformation will take place, cultural models describe the context within which the transformation will exist. Cultural models in SSM represent roles, social systems and political systems. Each of these is captured in what is known as a 'rich picture'. In essence, a rich picture is an informal diagram that represents the various relationships and value judgements that exist between roles, social systems and political systems. The representations used are up to the person responsible for drawing the rich picture but some form of semantics has been developed for rich pictures (e.g., a crossed swords symbol represents conflict). However, these need not be strictly adhered to.

Rich pictures will show various roles. Of prime interest to SSM are clients (who caused the initial study to take place), problem solvers (whoever wishes to do something about the situation in question) and problem owner (whoever is responsible for overseeing the problem solution). Modeling roles in this manner is fairly straightforward according to Checkland and Scholes (1990) but is a useful practice since it helps to define boundaries and responsibilities.

Social systems model the way that roles, norms and values interact. Roles are social positions recognized as significant by the stakeholders, norms are the behaviours expected of people fulfilling those roles. Values are the measures by which performance within those roles is judged good or bad. It is important to realise that the interaction between roles, norms and values is dynamic and that each will have an impact on the other.

Finally, political systems model the way that power is expressed in a situation. SSM suggests that a number of questions be asked to learn about the political system. For example, what are the commodities through which power is expressed in this situation? How are these commodities obtained, used, protected, preserved, passed on, relinquished? Through what mechanisms? The commodities through which power is expressed can take various forms, for example, formal authority, personal charisma, important information, membership of various committees etc.

By modeling the roles, social systems and political systems, preferably in a rich picture, it should be possible to see how the culture (or context) of a system will impact that system.

Summary

In essence, SSM is a method for thinking about problems systematically. It encourages consideration of the whole picture, from multiple points of view, so that a good understanding of the problem can be gained. This is important, since one of the major difficulties in solving problem situations, be they in business, computer systems design, education etc., is understanding and defining the problem.

SSM therefore provides designers with a potentially useful way to think about and model a design problem, such that the context of the problem situation is taken account of. However, the amount of relevant context that is modelled by the method still remains up to the designers. While the method suggests that designers take into account the different points of view on a problem, the social and political systems etc., identification of all the relevant information is still left to the designer.

Furthermore, while SSM could potentially be a useful tool for thinking about and representing a problem situation, it is less useful for using this knowledge to build computer support. Support does not exist for linking between the different models produced by SSM and a system design, nor are there any suggestions as to how a design can be derived from the models. Checkland and Scholes (1990) suggest that the method can be used to identify how best to represent data to users such that they can interpret it effectively but do not report on any large scale use of the method in system design.

2.8.2 SSADM

SSADM (Downs *et al.* 1991; Eva 1994; Weaver 1993) is a very large and detailed method of software design that was developed in response to the UK government's request for a standard software development method to be developed and utilised on all government software engineering contracts. The hope was that the use of a standardised method across all projects would reduce maintenance costs, increase mobility of staff between projects, promote open tendering and allow reuse of designs for different hardware platforms amongst other benefits.

In 1981, Learmonth and Burchett Management Systems released the first version of SSADM. It has since been through a number of revisions and in 1990, the fourth version of SSADM was announced. The main characteristics of SSADM are as follows.

- Multiple viewpoints of the system. Multiple viewpoints are provided by the use of a number of different modelling notations. A system can be modelled functionally, logically and chronologically in SSADM. SSADM makes good use of these multiple viewpoints to check for internal consistency of the system;
- SSADM takes a cookbook approach to system modelling. SSADM describes detailed techniques and recipes for each stage of the method. It uses checklists to make sure that all the necessary steps have been completed;
- SSADM can link to other standard techniques such as PRINCE (Projects in Controlled Environments) for project management and CRAMM (CCTA Risk Analysis and Management Method) for initial risk analysis.

SSADM consists of five modules, each of which consists of one or two separate stages. Each stage consists of a number of steps within which there are a number of tasks which are carried out using appropriate techniques. Figure 8 provides an overview of SSADM.

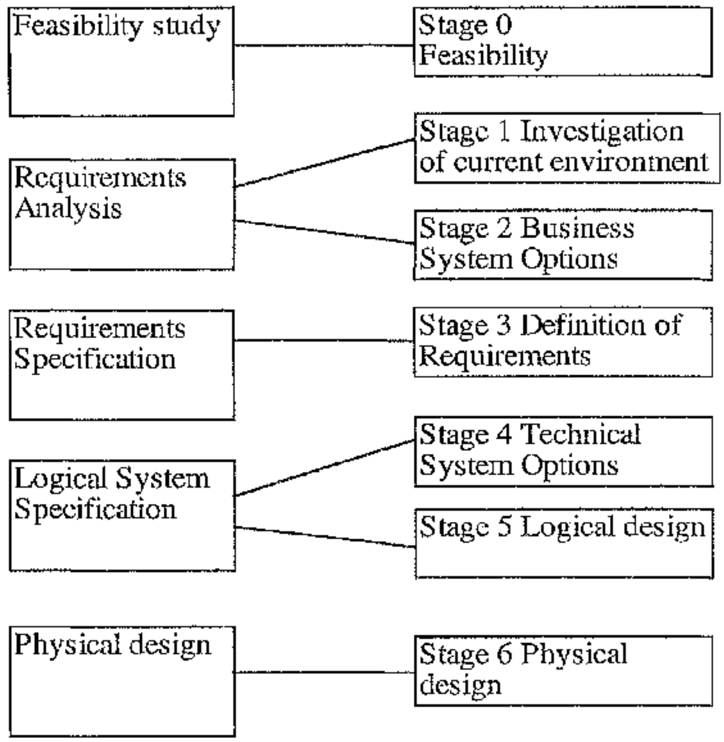


Figure 8. An overview of the SSADM method

There are three different techniques that are widely used throughout SSADM. These are

- Data Flow Modelling. This technique models the flow of data around the system using Data Flow Diagrams (DFD);
- Logical Data Modelling. This produces a structure that describes the logical structure of a system and how parts of a system relate to each other. It produces a Logical Data Structure (LDS) which is very similar to the Entity Relationship diagrams produced in Database Design;
- Entity Event Modelling. An Entity Life History (ELH) is created for each significant entity described in the Logical Data Structure. The ELH defines an entity in terms of the events that instantiate the entity, the events that affect that entity and the events that cause the termination of that entity.

Each stage described above in Figure 8 makes use of some or all of these techniques. Initially, a feasibility study is performed in which a problem definition is prepared and high level LDS and DFDs are used to describe the current system under investigation. These models are then used to define the requirements of a proposed system and subsequently refined into a logical system specification and a physical design.

There are a number of benefits to the SSADM method.

- The three different modelling notations are used to ensure consistency throughout the specification. For example, a Data Flow Model of an existing system can be checked against a Logical Data Structure. The Data Flow Model should describe separate processes to create, update and modify each entity identified in the Logical Data Structure. If it doesn't, the reasons for the missing processes should be well documented;
- The different stages are closely related in terms of their expected inputs and outputs. Each stage makes use of the outputs from the previous stage(s) and transforms them in some way. The 'recipes' defined for each stage describe how these transformations are applied;
- The first module of the method defines the scope of the system in terms of what will and will not be investigated and modelled.

However, there are also a number of major drawbacks, in terms of the arguments presented in this thesis.

- SSADM does not focus on contextual information nor on the ways that people will use a system produced using SSADM. It does not make use of contextual information to alter or modify any of the models produced. Each of the models described above centres on the system being developed by the method. They model the data that flows around the system, the life history of the data and the relationships between the data. Very little attention is paid to the ways in which people interact or use the data other than to define the scope of the system, to define the functions and relationships between the data that should be supported by the system;
- The cookbook approach does not give designers much freedom to adapt the method according to their particular project.

These drawbacks also apply to other traditional software engineering techniques and methods since the three techniques described above are widely used in other software engineering methods. Sommerville (1992) describes other software engineering methods which make use of the techniques described above or similar techniques. Gruhn and Wolf (1995) describe how they used a business process orientation to develop a new application for the support of all business processes of housing construction and administration. They used three different modelling techniques.

- Data models. These were used to describe the structure of objects and their relationships. This is a similar modelling technique to the Logical Data Modelling of SSADM;

- Activity models. These are used to define activities to be a process. These are similar to the Data Flow Diagrams used in SSADM except that they are based on high level petri nets (Reisig, 1986);
- Organisation models. These are used to define which organisation entities are involved in a process. They are related to the Entity Life Histories produced in SSADM since the organisational models are used to define access rights for starting or executing processes.

Having produced these models, Gruhn and Wolf (1995) then describe how they integrated each model, defining the different types used in the activity models from the data model and defining which organisational entities are responsible for which activities. Similarly to SSADM, the method described by Gruhn and Wolf has a systems oriented view. While the authors attempted to take account of the organisation in which the system would be used they only went as far as defining which organisations have responsibility for particular activities. No account was taken of contextual information and how this might affect how the activities are executed.

Hence standard software engineering methods, that make extensive use of the techniques described above, do not make effective use of contextual information due to the systems–centred view of the techniques used within the methods.

Parker *et al.* (1996) echoed this view at a User Centred Requirements Engineering Workshop (Johnson and Jones, 1996). At the workshop, participants discussed techniques for generating requirements for computer systems that focus on the users and not on the system. Many different approaches were considered at the workshop (e.g., modelling users tasks to develop requirements, modelling users knowledge about a system, modelling semiotic aspects of users (i.e., languages and symbols used by users), formally specifying user tasks and specifying temporal aspects of user interaction amongst others). However, none of these approaches focused primarily on the context of use of a system and how this information could be used to generate user centred requirements.

Diaper (1996) reported how the TAKD method developed by Johnson *et al.* (1984) could be modified to provide input to the SSADM process. Diaper reports how TAKD could be used to automatically generate Entity Life History's and other models. The process by which these models are automatically generated was not described, but presumably it would involve extracting the essential knowledge for the construction of an Entity Life History from the appropriate TAKD structures. Hence, all those aspects of a TAKD structure which are not applicable to the Entity Life History (be they contextual or not) are simply ignored.

Therefore, the same criticisms still apply, that little attention is paid to contextual information, even though HCI techniques have now been applied to generating some of the necessary structures for the SSADM process.

2.9 Summary

This concludes the review of current methods for making use of contextual information in design. Contextual information can be used to inform and influence the design of a product. There are many ways in which this can be achieved and this review has described a number of such methods. *Contextual Inquiry* employs interviews and observations to elicit contextual data and uses simple structures (e.g., affinity diagrams) to describe the data in such a way that concerns and issues are represented and can be tackled by designers. *Customer Centred Design* uses much the same approach but uses more complex structures to describe the data. *Ethnographic methods* generate ad–hoc but detailed descriptions of context which can be used to influence designs. Scenarios are another typically ad–hoc descriptive method, but in contrast to most ethnographic descriptions, are more easily understood and can be created and applied to all stages of design. *Activity theory* supports greater structure in descriptions of context, allowing systematic comparisons of different contexts to be made. However, it is really a descriptive tool and at present, offers little in the way of support for design. Finally, *participatory design* is a philosophy of design rather than a method. It suggests that a rich collaboration between users and designers will result in a useful and usable product being designed, but this review has argued that such collaboration leads to difficulties in maintaining and managing such projects and that context and design descriptions must be recorded together with recordings of the relationships between context and design in order that a product can successfully adapt to changes in its context of use.

The review has offered descriptions of each technique, method or philosophy and has evaluated each with respect to the five factors described in Chapter 1. The next section brings these evaluations together and evaluates each factor with respect to their contribution to making effective use of contextual information in design. It discusses the issues that are relevant to any method, technique or philosophy that attempts to support these five factors and make effective use of contextual information in design. The issues therefore have direct implications for this thesis. These implications will be addressed in Chapter 6 where they are used to inform the development of a framework for using context in design.

2.9.1 Linking between representations

Throughout the review a number of different arguments were made for making the relationships between context and design explicit. Explicit representation of the relationships between context and design can aid assessments of the usage of contextual information in design (see the case studies in Chapter 4) and can aid maintenance of a system by indicating how different parts of the system depend on each other. Some of the methods reviewed also support some form of linking between representations.

Rosson and Carroll (1995) use claims analysis to make links between different scenarios. The links can serve a number of purposes. They provide access to the rationale behind the design of the system. A scenario and the claims made for that scenario may inspire the design of a new feature. In order to explain the feature and to describe the reasons for its existence, a link can be created between the feature and the scenario, via the appropriate claim. Then, if designers want to understand why the design feature exists, they can follow the link through the claim to the scenario that inspired the feature. The claim gives a detailed reason for the feature and the scenario allows the designer to see the context in which the feature should exist. Thus designers can make sense of a design by traversing the various links between it and the scenarios that inspired the various features of the design.

Other uses of links are to provide evidence of a scenario's pervasiveness throughout a design. For example, a claim may be made about a particular scenario. The designer may realise that the claim is in fact relevant to a number of scenarios and decide to link each of these scenarios to the claim such that its pervasiveness is illustrated. Links can show the relationships between different scenarios. Rosson and Carroll (1995) suggest that a scenario is unlikely to describe all the consequences of a particular design feature. It may require more than one scenario to describe the different effects and uses that a particular feature may have. In this case, it is useful to be able to link all related scenarios together, so that designers can easily see the full consequences of a given feature. The authors claim that combining claims analysis with linking between scenarios "allows the designer to record explicitly not only the causal relations in each scenario, but also the ways in which these causal relations interact across scenarios."

Kyng (1995) also makes use of links in his method, although to a lesser extent than Rosson and Carroll. By linking use scenarios to the originating work situation description(s) that inspired them, designers can ground the scenarios in real situations and fill in any missing details that may become apparent when the scenarios are used by real users. Thus Kyng suggests using links to back up scenarios and to provide any missing detail.

Both papers, especially the Rosson and Carroll paper present a good case for explicitly linking between related representations of work and designs. Further support for linking between representations can be found in Kaiya *et al.* (1995) and van Aalst *et al.* (1995). Links can take different forms, for example they can be mediated by claims analysis or other design rationale notations (e.g. QOC, Maclean *et al.* 1991) or they can be direct links. The links can be used for a number of purposes as described above. Mediated links, as opposed to direct links, provide extra explanatory power, at the cost of added complexity over the representations involved. This extra explanatory power could be vital for maintainers of a system. Once a system is installed and delivered it must be maintained, so that it can adapt to the inevitable changes in its working environment. To ensure that only appropriate changes will be made to a system, the maintainers of the system need to know the reasons for design decisions, so that they know whether or not the proposed changes are in conflict with the original rationale behind the design, or indeed, if the original rationale still holds. In this way, explicit links between design and context representations, mediated via design rationale representations, can aid maintenance of a system.

Another use of explicit links, one that is not suggested by either Rosson and Carroll (1995) or Kyng (1995), is in identifying aspects of the design or context that are affected by a change. If designers make a change to either the design or the context representations, they can use explicit links to identify other aspects of the design or context that may be affected by that change. For example, if a particular aspect of context is used to influence three different design features, then, if that aspect of context changes in any way, links created between the context and design representations can identify the elements of the design that may need to change as a result of the change in the context. At the very least, the parts of design influenced by the changed piece of context need to be investigated to ensure that they are still valid under this new element of context. At the other extreme, the design itself may need to be changed or even discarded. Just as the change in context sparked investigation of potential change throughout the rest of the representations used, so too can the corresponding change in a design spark off further potential for change. Thus, designers can ensure that changes are propagated throughout a design appropriately. It is clear that one small change can create an exponential growth in the number of subsequent changes required to keep the design and context representations consistent with one another. Thus, another use of explicit links would be to evaluate the cost of making a change. If the semantics of a link are clear then designers will be helped in determining an appropriate action to take as a result of a change to the source or destination of a link. If it is clear that the cost of accommodating a particular change is high, it may not be worth making the change.

Other kinds of evaluations are possible through links between context and design representations. Designers can identify potentially inconsistent usage of contextual information by identifying those pieces of context that have been used more than once in influencing the design. The same piece of context should be used consistently throughout the design. By inspecting all the potential sources and destinations of links, designers can judge how fully the contextual information has been used in forming the design. Designers can inspect the ways in which context has been used in the design, since they have direct access to this information, and hence can use their judgement to determine the appropriateness and validity of the link, with respect to the project the designers are working on. Further judgements can be made about the relevancy of contextual information, since it should be possible to identify elements of context that have not been used to influence the design. Having identified these elements, designers can decide whether they should exert an influence over the design (i.e., is relevant) or if it should be ignored (i.e., is not relevant).

Thus, links between context and design representations make different kinds of judgements possible. These judgements can provide a way to help manage the information collected about a system's intended context. For example, designers can use the links (or the lack of links) to identify areas in both the design and context where further investigation is required. If some part of the context has been used sparingly throughout the design, it should be investigated further to see if it could have a greater influence over the design. In contrast, if some other part of the context has been used heavily throughout the design, then it is likely that further investigation into how this piece of context could be used further in design is not required. By making these judgements, designers can get a better handle on the large amounts of data that are typically collected as a result of ethnographic or contextual interview methods. In an iterative design process, designers can use the links to identify areas of further investigation during each iteration, instead of collecting large amounts of data in one go which may go unused or used without verification.

2.9.2 Scope

Each method defines a different scope of relevant contextual information either implicitly or explicitly. They differ in two ways. First of all, the methods differ in the ways in which contextual data is collected. Different methods of data collection result in different sets of data being collected. Secondly, the methods differ in the way that they use contextual data. To use contextual data differently, different kinds of contextual data must be collected. Not all data is relevant to all the ways in which contextual data can be used. These two differences in scope are elaborated upon in the following sections.

Collecting contextual information

There are three different factors which contribute to the different scope defined (implicitly or explicitly) by each method in terms of the data collected by that method. These are

- focus;
- elicitation methods;
- post–processing.

Each theory has a different *focus*. For example, Contextual Inquiry suggests that analysts create a focus which defines the concepts and concerns that the analysis is concerned with and which identifies aspects of users' work that users are likely to be able to articulate or answer questions about. This has benefits in that it clearly defines what the analysts should be investigating, but it has a number of drawbacks. The major drawback is the concentration on those aspects of work that users can articulate. Activity theory stresses the importance of investigating the tacit aspects of users' work, the very aspects that users cannot articulate. It states that it is essential to understand the users consciousness as they perform their work, to know what they are thinking and just as importantly, what they are not thinking about. People can perform two identical actions for completely different reasons. These different reasons may have a large impact on the design of any system to support people at their work. Therefore it is vitally important to know what the user is thinking and how their working environment affects the work they do.

A further drawback in terms of using a focus is the ease with which the focus can be defined. In different situations it may not be so easy to define a focus, particularly when the analysts know little about the domain they are analysing. The third drawback however is potentially more damaging. By focusing on a subset of the potentially relevant context, analysts run the risk of ignoring or not even seeing, some aspects of context which, although not defined in the focus, turn out to be relevant during the analysis. This would mean that important data would be at risk of not being recorded with implications for the design. Lastly, as Simpson (1996) points out, analysts may have to keep on top of a number of foci. The main focus is the one that defines what they want to learn as a result of the study. There is another focus, however, which is on the actual study itself. Simpson suggests that if a team wants to learn about existing documentation, it will use this as its main focus; while performing the actual study and observing users using the documentation, they will focus on the use of the documentation.

Each of the other methods have implicit foci. Bødker's application of activity theory tends to focus on learning, designing dialogues such that novices and other users can learn effectively how to use the system and experience fewer breakdowns. Activity theory itself focuses on the individual and how they work, although others have attempted to extend activity theory by including notions of the community and how individuals relate to the community via social rules. Customer Centred Design tends to focus on the constraints that the context places on the users, while Rosson and Carroll's method has a development focus.

The *elicitation methods* used also limit the scope of the data that will be collected. The contextual interviews used by Contextual Inquiry and Customer Centred Design only collect the data that users can describe or show. Interviews will not be able to discover the tacit elements of work which activity theory highlights as essential to generating a good understanding of users' work. These elements are difficult, if not impossible, for users to talk about. Furthermore, interviews are not a reliable means to collect data about activities which last considerable lengths of time. Even if in these cases, as Wixon and Raven (1994) suggest, analysts ask users to describe recently completed activities, many aspects of the work will not be described by users as they will have either forgotten about them or do not consider them important enough to discuss. Users views on an activity differ once it has been completed from views while the activity is being performed. Users may be able to look back over an activity and understand why they had to perform some action but at the time, the reasons for performing the action may not have been clear and may have caused a considerable amount of confusion. The source of the confusion will be difficult to ascertain if users are trying to describe the event at a later date when they have the benefit of hindsight and now understand the confusion (see Carroll *et al.*, 1994 for a description of a system that captures design history and rationale).

Bødker uses prototypes to elicit data about the context, and while this approach has a better chance of eliciting the tacit aspects of users' work, it can suffer from encouraging users to concentrate on low level aspects of the interface such as button placement and command names rather than higher level issues such as the work they are trying to do. Also, prototypes can only represent a certain subset of the users' work (namely, those aspects of work which use a computer or some other physical tool). The other aspects of work could be ignored by relying on the prototype to elicit data.

Using scenarios alone to elicit data about users and their work is difficult, since before a scenario can be created some initial knowledge about the work must be available. Furthermore, it is doubtful whether or not a set of scenarios could be created that fully describe all relevant aspects of users' work. There are likely to be gaps and omissions in the scenarios that require supplementary information and data to back them up. The scenarios also create some kind of focus, since only those aspects of context which can be described in or discovered through a scenario will be elicited.

Further problems are encountered in the way that each method *processes* the data. None of the methods use the data in its 'raw' form preferring instead to use 'surrogates' such as the affinity diagram and the various models used in the Customer Centred Design method. These surrogates are structures that organise some of the data, and as such are good at highlighting some aspects of the data but not others. For example, the models produced using Customer Centred Design focus on relationships between entities (and in particular, constraining relationships) but filter out other aspects of the data, such as attributes of entities which can be important. The surrogates used by each method will tend to promote some aspects of the data while hiding others. For example, in the IT case study described in Chapter 3, it was difficult to represent the number of interruptions that the IT secretary typically experiences while carrying out her job in the context diagram, shown in Figure 2, section 2.3.

Using contextual data

Each method uses contextual data for different purposes. Bødker (1996, 1991) uses contextual data to make formative evaluations of the interface, looking for areas where breakdowns will occur and investigating how these can be avoided through making the interface easier to learn. Wixon and Raven (1994) suggest that the affinity diagram can be used in many different ways, such as to create work metaphors, scenarios of work or to create new questions for further contextual analyses. Rosson and Carroll (1995) use the elaborated scenarios to determine the objects that should be implemented in the application, while Holtzblatt and Beyer (1993) use the models to define the scope of the computer system and to make some attempts at defining how it should behave and appear to the user.

A method that attempts to identify the software objects to be implemented in an object oriented system will scope context differently to a method that attempts to use contextual information to perform an evaluation of a system. In Rosson and Carroll's use of contextual information, they focused on the entities that users manipulated when maintaining a database of bibliographic information. They focused on the ways that users made use of bibliographic information in order to identify software objects that would have to be implemented in a bibliographic database management system that would support users in their tasks. Bødker reported how the word processor used at the Danish National Labour Inspection Service interrupted the user's task of creating a report when they attempted to insert a page number into the report. The software did not allow users to insert a page number easily. In this example, Bødker uses the contextual information to suggest how the dialog aspects of the software are less than optimal and could be improved. In Bødker's method, she focusses on contextual data that allows her to make an evaluation of the software in terms of how well it supports users' tasks, i.e., the dialog aspects of the software and how it relates to users tasks whereas Rosson and Carroll focus on the contextual information that allows them to identify potential software objects i.e., detailed descriptions of users tasks.

In summary, each method scopes contextual data in a particular way. If a method is used by itself, analysts will collect a particular view of contextual data that is suggested by the method they use. It is vital that analysts are aware of the individual views of context offered by each method, so that analysts can either choose the method that will most closely match the view of context they require for their project or so that analysts can combine different methods in a useful way, using one method to collect particular types of data and another method to collect other types of data.

2.9.3 Definition

Each of the approaches either explicitly or implicitly attempts to define context. The Contextual Inquiry method states that context is *"The interrelated conditions within which something occurs or exists"*. This suggests that the context of the object of analysis, for example, users' work, is the conditions within which that work is performed and the ways in which the work relates to other situations. However, this definition can in no way identify all those relevant aspects of context for a particular project. It is too broad and vague (and Wixon and Raven probably wouldn't suggest that it be used to identify relevant aspects of context). What it does is highlight the fact that context is something beyond the object of interest itself. It makes designers aware that they must investigate peripheral aspects of users' work if they want to understand the context of that work (c.f. Seely–Brown and Duguid, 1995).

The definition does not identify individual elements of context for individual projects. It points designers in the direction that Wixon and Raven believe they should look for relevant aspects of context for their project. Other methods offer similar directions or foci of attention. Customer Centred Design suggests designers pay attention to those aspects of users' work that constrain the way the work is performed, be that through organisational constraints, physical constraints or other means. Activity theory defines different levels of activity and consciousness and implicitly defines context using these constructs. Ethnographic methods of analysis offer no direction to avoid the bias inherent in methods that suggest particular directions to investigate potentially relevant context. Participatory design is similar in this respect, suggesting that designers can never fully understand users' context and that only users can define these aspects of their work for use in a design project. It would seem that the best that can be offered by any method is a guide, or direction to what 'context' is. Concerns about potential bias from pre–conceived notions of context are well founded (Holtzblatt and Beyer, 1993; Wixon and Raven, 1994), but in practice, it is more than likely that designers will have some pre–conceived notion of what the relevant context may be for their project and it is better to make this explicit, so that consensus can be reached amongst all the designers, rather than ignoring it and allowing each individual's own ideas to interfere with the analysis (Cockton *et al.*, 1995).

Understanding context would seem to be like learning to program. You can't do it until you try. No matter how much instruction someone gets programming, they can never learn how to solve a programming problem without trying. They may know how to write a while loop, how to declare an array, how to call a procedure, but this all seems very abstract until they have to use this knowledge in writing a program. The same is true for understanding context. All the definitions described so far can point designers in a particular direction, but it is only when working in a real project that designers can start to fully understand and use context. Just as a programmer gains new knowledge and expertise by programming, a designer gains an understanding of context by applying a definition of context to a particular project.

So, *producing a detailed definition of context is not a prime goal of this research.* This thesis argues that in order to understand context, designers must 'get their hands dirty' and try to understand context in terms of a particular project. Definitions like those given above are useful but not crucial to making successful use of context in design. Ethnographic methods have been used successfully in understanding context and using this understanding in design, without using a definition of context. However, ethnographic techniques typically require skilled ethnographers and some of the definitions given above can be so vague that designers struggle in their attempts to use contextual information. To make some of the definitions given by other methods more concrete, this research has produced a 'map' of context in the form of a checklist similar to the checklist used in the MUSiC project (Maissel *et al.*, 1993). This map of context lists elements of context that have influenced design projects in the past and was compiled through investigating descriptions of design projects that used contextual information in some way and through personal experiences and contact with other designers. The checklist is shown in Appendix A. This checklist is merely provided as an aid to designers and is not central to the arguments presented in this thesis. It simply provides another view of context that may be useful. Its use is described in more detail in the next chapter.

2.9.4 Quality of Contextual Data

One of the major issues related to using contextual information in design is the quality of contextual data that is used by designers. If the data does not reflect reality, then designers will most likely design a system that does not meet the needs of its users. Designers should strive to collect and use quality data. Hence, the quality of data collected and used by designers needs to be verified and backed up.

Kyng (1995) reflects these concerns by suggesting that the scenarios used in his method are linked back to the original work situation descriptions that inspired them. That way, the scenarios can be verified to reflect reality and any omissions in the scenarios can be identified. Gaps in contextual data lessen the quality of the data collected. If the data collected mirrors only a part of users' work, missing other parts out, then at the best, a system designed using this data will only support parts of users' work. At worst, it will not support users' work at all, since the context omitted may have a large impact on the ability of the system designed to support users' work. The scenario based method of Rosson and Carroll may fail to provide quality data, since as Nardi (1995) described, it may be impossible to generate a large enough set of scenarios such that all aspects of relevant context are described. Activity theory uses other methods to collect data such as ethnography or prototypes (Bødker, 1991) and is used to describe the data and so has little effect on the quality of the data collected. Ethnographic data depends on a large part for its quality on the availability of skilled ethnographers to the design team. Ethnographers collect enormous amounts of detailed data, some of which is described in ad–hoc reports, but a lot of which is kept in the head of the ethnographers and only described when designers explicitly ask for it. This is due to the amounts of data collected and the lack of standard notations and models to enable descriptions of data to be written down.

Ethnographers have no way of predicting what will be the most important elements to describe. Furthermore, if they were to attempt to describe every piece of data, the resulting description would be so long that it is unlikely that it would be read by any of the designers. In participatory design, a similar situation holds. Designers rely on continual access to the users to ensure that they collect quality data. They also need to make sure that the users they are speaking to are a representative sample of the population of users, something that may be difficult to achieve in a diverse user population. Kleimann (1996) made this point with respect to the Contextual Inquiry method that claims that to ensure quality data, designers should interview three different people at three different sites. How can designers be sure that they are collecting all the relevant data, when users can differ on a number of dimensions such as age, gender, experience etc.?

2.9.5 Understanding the Data

A crucial point about using contextual data in design is that designers should understand the data they have collected or been presented with. If the data is of high quality and covers a wide scope but is incomprehensible to designers then it will not be used successfully in design. One of the main benefits of scenarios is that they are easily understandable to everyone on the design team (as long as they are written in a language understood by all members of the team and are not ambiguous). Although Kyng (1995) claims that they cannot be fully appreciated by people who were not involved in the creation of the scenario, they are nevertheless a good communication tool.

By providing a common vocabulary, activity theory aims to increase understanding of data by avoiding ad–hoc descriptions and terms. Furthermore, it allows systematic comparisons between different descriptions of context to be made, due to the common structure it provides descriptions.

Ethnographic accounts, on the other hand, are difficult to understand by people not directly involved in the collection of the data, since the data is so detailed and descriptions of the data are ad–hoc. Ethnographic descriptions do not use pre–defined terms in their descriptions which can increase the burden of understanding for other people on the design team.

Designers must be able to generate a good understanding of the contextual data in order to make the best use of it that they can. Data that cannot be understood will not be used, or worse, will be used inappropriately.

2.9.6 Chapter Summary

This chapter has described a number of different methods for making use of contextual information in design. Contextual information can be used throughout the whole design life cycle of a project, from initial 'requirements' gathering all the way through to evaluation. Each method has been compared to the five factors that this thesis argues must be supported if contextual information is to be used effectively in design. Table 9 summarises each method's level of support for each factor (1 = poor support, 5 = excellent support. Areas left blank mean that there is no direct support for this factor.).

	Scope	Definition	Quality	Understanding	Linking
Contextual Inquiry	4	3	2	3	1
Customer Centred Design	2	3	2	3	3
Ethnography			4	2	
Activity Theory		3		3	
Scenario Based Design	3		3	3	2
Participatory Design	3	3	4	4	

Table 9. A summary of the level of support for each of the five factors

To be used effectively, good quality contextual information must be collected, information that reflects the context in which the system being designed will operate. The information must be understood by designers and the understanding must be shared across the whole design team so that consensus is reached amongst the team and priorities can be set and agreed upon. Links between context and design representations support judgements about the way the context influences the design (and vice–versa) allowing designers to evaluate their progress and to adjust their plans accordingly. Links can also highlight the rationale for different design decisions. Maintenance is also supported by the links, especially if the rationales for the design are recorded, since maintainers can inspect the rationales for a particular design feature as well as examine the contextual information that suggested the feature.

3. Chapter 3 The IT Case Study

3.1 Introduction

The previous chapter presented a review of current methods for making effective use of contextual information in design. Five factors were identified which the review argued must be properly supported in any method that claims to make effective use of contextual information in design. The first such factor, linking between representations of context and design is central to this thesis. This thesis argues that if the relationships between context and design are known and explicitly recorded, assessments of the usage of contextual information can be made, which can in turn increase the effectiveness with which contextual information is used. For example, assessments regarding the extent to which contextual information has influenced the design can be made by enumerating the design decisions and elements that have been influenced by the context. Such design decisions and elements are indicated by links to them from aspects of context.

A design project which attempts to record the relationships between context and design, will probably produce many links between context and design due to the many different ways that context can be used to influence a design (Moran, 1994; Whiteside *et al.*, 1988). Given this, how might the representation and maintenance of these links be best supported? Are tools required, and if so, what features must they support? This chapter describes a case study that was performed to investigate the issues involved in recording and maintaining a set of links between context and design. The case study also provided valuable information that was used to suggest at what stage in a design project it would be most appropriate to record the relationships between context and design.

The chapter begins with a description of the study. The conclusions reached from the study were used to define a non–exhaustive set of requirements for a tool that supports the recording and maintenance of the relationships between context and design. These requirements are described, followed by a description of a tool that was implemented in Smalltalk to support these requirements.

3.2 The IT Case Study

The Computing Science Department at the University of Glasgow runs a postgraduate Masters degree course in Information Technology. It is a large course with approximately 120 students studying full time for one year. The income generated from the course is substantial. Thus, the successful management of the course is crucial to the well being of the whole department. The management of the IT course therefore provided a number of different opportunities for studying a real life, business–critical domain. I decided to focus on the process by which applications for the course are handled, since this is a process that takes place throughout the year and is not tied to a certain period. This meant that I did not have to wait until the appropriate period in order to perform the study.

3.2.1 Initial Interviews

The different stakeholders involved in processing applications were identified (applicants, applications secretary, admissions officer, course head and faculty secretary) and interviewed where appropriate, employing the contextual interview technique described by Wixon and Raven (1994). The focus of these interviews centred on the current system used to process applications. I wanted to understand the process and the tools used by the different stakeholders. Each stakeholder was interviewed except for the applicants. Due to the number of applicants and the widely varying backgrounds of each applicant, there was insufficient time to interview a representative cross sample of applicants (see Kleimann, 1996). Instead of interviewing applicants, questionnaires were sent to the current set of students. The questionnaires asked them questions relating to their experiences in applying for a place on the course. This meant that it was possible to gather a lot of information, since each student was sent a questionnaire, but information relating to those applicants who had been refused a place could not be obtained. Given the time and resources available, this was a constraint that I had to put up with. However, I assumed that information from both rejected and accepted applicants would not be radically different.

3.2.2 Modelling the Information

The information gained from the interviews and questionnaires was modelled using Customer Centred Design. Context, flow, physical and sequence models were all produced. As was described in the previous chapter, different methods for modelling and using contextual information in design focus on different kinds of contextual information. When I tried to create a sequence model for the course head, I found that I had not collected the relevant information which would allow me to create such a model. Sequence models *"represent the sequence in time of actions for specific important activities"* (Holtzblatt and Beyer, 1993). Therefore, guided by the scope of the sequence model, I interviewed the course head again. I also suggested that the course head keep a track in a diary of the important activities she performs over a period of time. Unfortunately, due to the varied nature of the course head's job, she felt that it was not possible to model her job in this manner, since her job does not generally follow set procedures or tasks.

As has been described, modelling the data using the Customer Centred Design models was a useful exercise which identified some contextual information that had not been collected in the first set of interviews. However, information that had been collected did not easily fit into the models. For example, quantitative information, such as the number of interruptions the applications secretary experiences a day, could not be accommodated within Customer Centred Design. So that this information would not be overlooked, it was modelled using the checklist of contextual factors shown in Appendix A.

3.2.3 Problems with the current system for processing applications

Having modelled the data, it was analysed to identify any problems that existed in the current system for processing applications. Three such problems were identified:

> 1) Some applicants experience long delays between sending their applications in and receiving a decision from the University and do not receive enough feedback about their application.
>
> 2) Most information about applicants is only stored in paper files which makes it difficult and time consuming to do any queries over all the applications.
>
> 3) Difficulties in optimising enrolment (i.e., accepting the right number of quality students on to the course).

Some of the causes of the delays and lack of feedback received by applicants appeared to arise from the system being used to support the applications secretary in her job. At present, she uses MS Word, FileMaker Pro and paper files to store details about, and keep track of, an application. Since she uses three separate packages, she tends to process applications in batches, since to do otherwise would mean spending time switching back and forth between the computer applications and paper files, wasting time at each switch, reorienting herself to different contexts. By processing applications in batches, she can perform some task for each application in FileMaker Pro for example, then switch to MS Word and perform some other task on all the applications, necessitating only one switch between MS Word and FileMaker Pro for all the applications.

Some of the reasons for the lack of feedback received by applicants could also be put down to the support system used. Since three individual systems are employed, there is a difficulty keeping information consistent between each system with the result that most of the information regarding an applicant is stored in the paper file, including the status of that application (e.g., still waiting for referee letters, accepted offer). This means that it is virtually impossible to perform any detailed queries, such as finding all those applications for which no referee letter has been received. Hence it is impossible to provide applicants with any detailed feedback on the status of their application.

The lack of information stored electronically also explains some of the difficulties the department experiences in optimising enrolment. Optimising enrolment involves achieving targets set for both the size of the class and the proportion of overseas students while at the same time maintaining, or improving, the quality of students. The main difficulties in achieving these targets are that the admissions officer is never sure how many applicants who have been offered a place will turn up to take the class. Some applicants will have been offered places on other courses, or will have gained employment, and usually do not tell the department that they will no longer be able to accept the place. This means that if the admissions officer simply accepted as many applicants as there are places on the course then the course would be under booked, since some of those offered places would not turn up. Due to these uncertainties, the admissions officer must overbook the course. However, there is a certain 'critical point' which is reached when, if any more offers are made, it is likely that the course will have too many students, since even when some of them do not turn up for the class, the remainder will still be too large a group as too many applicants were offered a place originally. It is difficult though for the admissions officer to estimate when he has reached 'critical point' since the number of applicants who will not turn up varies every year. Since the administrator is the only one with real access to the files, in order to keep track of the numbers accepted, the admissions officer has to ask the administrator to go through the files occasionally and compile lists of those who have firmly accepted an offer and those who have accepted conditional offers, so that he can make a judgement on likely numbers on enrolment day. This causes difficulties since, as well as increasing the administrator's workload, the process is time consuming and error prone, meaning that the results will be received by the admissions officer some time after he requested them and that they may not be wholly accurate (other applicants offered places in the interim, mistakes in compiling the results).

3.2.4 Addressing the Problems

This section describes the four high level design decisions that were taken to address the problems identified in the previous section. The first decision is aimed at minimising the apparent delays that applicants experience. One way in which this can be done is to integrate more closely the facilities offered by FileMaker Pro, MS Word and the paper files, by providing some kind of front-end system so that the secretary can do everything such as entering information and creating standard letters from within one application. This means that time is not wasted switching between the different applications. It also means that the system can be specialised, i.e., designed with the purpose of processing IT applications in mind.

Another way in which delays can be minimised, and feedback applicants receive about their application increased is by storing the status of each application electronically. If this is done, then the computer system can be responsible for highlighting those applications in a particular status. For example, it would be possible to display all those applications which were still at the 'Waiting for second referee form' status. Then, appropriate action for each application could be taken, speeding up the process and minimising delays experienced by applicants. In the present system, this operation would be very difficult to do since the status of an application is only stored on a piece of paper in a physical file, meaning that each file would have to be retrieved by hand and kept aside if a second referee form had not been received. This would be a time consuming and error prone process.

Since the status of an application would now be stored electronically, letters detailing the current state of the application could be generated automatically whenever appropriate, for delivery to the applicant. For example, if the secretary was entering the details of a new application, the standard letter for replying to a new application could be generated automatically, with the system filling in the details such as applicant's address. This would increase the amount of feedback provided to applicants, and again would help minimise delays since the chances of forgetting to create a letter for an applicant would be reduced. It could also minimise delays which are caused by the 'inactive application' scenario. For example, if the status of a file has not changed for two weeks, a letter could be generated which would explain the situation to the applicant.

To address the problem of optimising enrolment, network access to the system could be provided to each stakeholder so that all members of the IT team could access the application database remotely and perform their own queries, reducing the secretary's workload and providing immediate feedback to the particular member. Again, each appropriate member of the IT team would have their own front end to the system. For example, the admission officer's might contain menu options for calculating how many applicants will turn up on enrolment day given a variable 'turn–up' rate such as 70% of those who have accepted an offer, 80% etc. This option would enable him to calculate how many more applicants he can accept on the course until the course is full. A facility could also be provided which would offer guidance on the likelihood of certain students turning up, based on experiences from previous years. For example, it might be the case that students from Glasgow are less likely to enrol on a postgraduate course in Glasgow. The system could attach a certainty value to each prediction it makes so that it might suggest that for example, 134 students will enrol with 75% certainty. The quality of applicants accepted could also be checked. Using past records the system could compile the average results achieved by applicants by country, institution, age, previous experience etc. This could then be used to check the likely quality of students being accepted for the next year. These features would provide further help to the admissions officer in his role but would also prove useful for marketing the course since the statistics collected can help identify countries and institutions which have provided good students and where further marketing of the course might prove advantageous.

Therefore, there are four main ways in which the new system attempts to address the problems described above. These are

1. Automatic tracking of the status of an application
2. Automatic generation of letters
3. Network access
4. Queries by role.

3.2.5 Refining the Design

I next started to design a system that would support the features described above. The QOC notation (Questions, Options and Criteria, Maclean *et al.*, 1991) was used to record the design rationale of the system. Most of the design decisions made some reference to contextual information. Whenever this was the case, I always tried to record the relationship between the design decision and the contextual information on paper. For example, in the QOC diagram shown in Figure 9, the decision on when to handle phone queries from applicants is based solely on the human context.

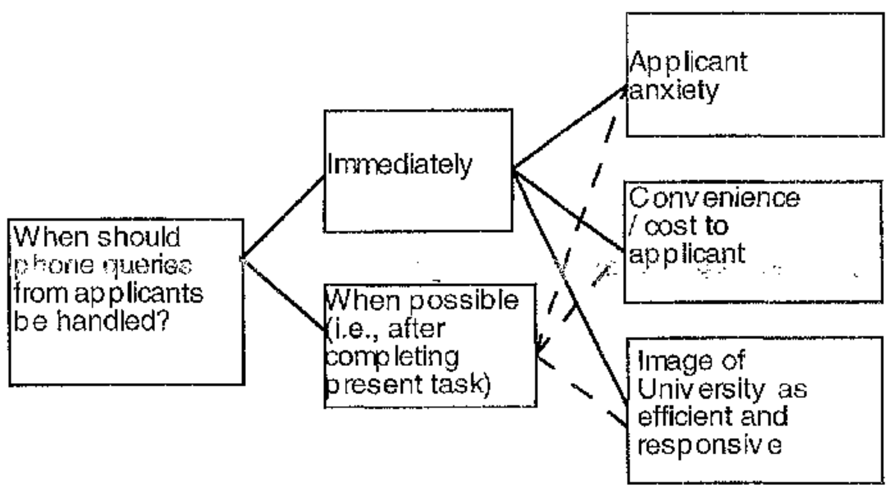


Figure 9. A QOC diagram showing the options and criteria for deciding when to handle phone queries.

Figure 9 addresses when the applications secretary should be able to answer a phone query from an application. The two options are (i) immediately or (ii) when possible. The criteria to evaluate these options are aspects of the system's human context. These may appear to vary in generality from the almost universal (anxiety, cost, convenience) to the specific (image of university). However, all are specific to the relevant context, which gives rise to anxiety, costs and inconvenience. Applicants are understandably anxious about their educational future, and would like immediate answers to queries (otherwise they would not phone). Cost is important, particularly if applicants are phoning from overseas. Lastly, universities should give applicants the best possible service.

I used envisionment scenarios to describe how I imagined the system would be used. For example, Figure 10 shows the scenario that describes how the applications secretary performs a query.

Scenario 2

The secretary receives a telephone call from an applicant who is asking about their application since they have heard nothing from the department in the last month. The secretary asks the student if they know their application reference number but they don't. The secretary asks for their name. She selects the 'Create query' menu option. After confirming that this is what she wants to do, each field in the current applicant view is made blank. The message box at the bottom of the screen tells her to enter details in the current application view. She moves the cursor to the Surname field in the current applicant view and types the applicant's surname. The related current query view is updated and shows summary details of all those applicants whose surname matches the surname in the current applicant view. The secretary scans this view but cannot see the applicant. She then types the applicant's forename in the forename field. This time, three applicants whose name (forename and surname) matches the forename and surname in the current applicant view are shown in the current query view. The secretary clicks on each one in turn until the correct one is found (The current applicant view displays all the details of each applicant clicked on in the current query view.)

Figure 10. A scenario describing the applications secretary performing a query.

All the scenarios were written in this concrete style which described the system and the various objects and actions that are present within it (e.g., 'current query view' and 'Create query' menu option). Each scenario was used both to validate the design with the context as understood (attempts were made to record the relationships between the scenarios and the context) and to identify the appropriate objects and actions that would have to be implemented in a computer implementation of the system. These objects and actions were specified using NUF (Notation for User Functionality, Cockton *et al.*, 1996) and UAN (User Action Notation, Hartson *et al.*, 1990) notations respectively. These notations were used due to the expertise of various members of staff in the department who were available to provide assistance.

Hence, an abstract description of a system that addresses the problems described in Section 3.2.3 was produced. As has been described, throughout the process of creating the abstract design attempts were made to record the relationships between context and the design. Experiences in recording the design, the context and the relationships between both on paper were used to generate an initial, non-exhaustive set of requirements for a computer-based tool that supports recording the design, context and the relationships between both.

The abstract description of the design and the context was also used as the basis for an IT project, undertaken by an M.Sc student, Saskia Koehler (Koehler, 1995). Saskia transformed the abstract descriptions into a concrete system which was implemented using HyperCard 2.3. Her experiences were also used to add to the set of requirements for a computer-based linking tool. Readers should refer to Saskia's dissertation for a detailed description of how she was able to transform the abstract description into a concrete system. The next section describes problems and issues that were raised by both the case study and the M.Sc project.

3.3 Reflections on the case study

The major problems with using the simple paper and pen tools in this case study and in the M.Sc project (Koehler, 1995) are described below. These problems were used to generate the set of requirements for a computer based support tool. The next section describes the problems in terms of the different types of activities that were performed throughout the case study. The list of different activities is presented below, followed by a detailed description of each along with the difficulties experienced in carrying out these activities.

Context elicitation and modeling

- Recording and structuring context;
- Identifying relevant context;
- Updating known context and the design;
- Transforming abstract design ideas into a concrete design.

Links

- Knowing which pieces of context have been used;
- Propagating changes in context/design;
- Evaluating how contextually grounded any aspect of the design is;
- Maintaining consistency.

3.3.1 Recording and structuring context

A combination of tools and simple free form textual notes were used to record details of the context in the IT study. Some aspects of context were simply committed to memory. When recording and structuring the contextual information, it was not always clear what information was important. Therefore some of the contextual information was not recorded until it became important. For example, in the initial interviews the status of an application was described by stakeholders a number of times. However, this was not initially recorded. Only when it was decided to automatically generate letters to provide more feedback to applicants did it become important to record the status of an application and how it changes.

Of the tools that were used to record the context, each had its own weaknesses, but when used in conjunction with one another, they complemented each other. For example, a checklist approach is useful for recording single, atomic details about the context such as experience of users, age etc., while graphical models or scenarios are better at capturing the ways in which these atomic details interact. Since context is so complex it would be difficult to derive a single tool or method which allowed designers to record all aspects of context in a way that allows easy access and understanding of the context by all members of the design team.

3.3.2 Identifying relevant context

One of the tools used during the study was a checklist of contextual factors. This list was a list of the factors that were thought to be relevant to this study. This was a useful aid for highlighting those aspects of context which had not yet been investigated. Any slot in the checklist which was left blank was either left that way because it had not been investigated yet, or because in actual fact, it was irrelevant to the IT study. The checklist was a reminder for those aspects of context which could still require investigation.

While the checklist was useful for identifying relevant context, certain design choices also made some aspects of context relevant. Certain design options made clear aspects of the context which required further investigation, but which had not been included in the checklist. For example, the dialog box presented to the secretary on start–up (see Figure 11), which shows those applicants which have not been updated recently, made the way that the secretary identifies applicants relevant, since there was limited space in the dialog box to display details of each applicant. Therefore, the checklist, or the idea of what aspects of context were relevant to this project was being constantly revised as the project progressed.

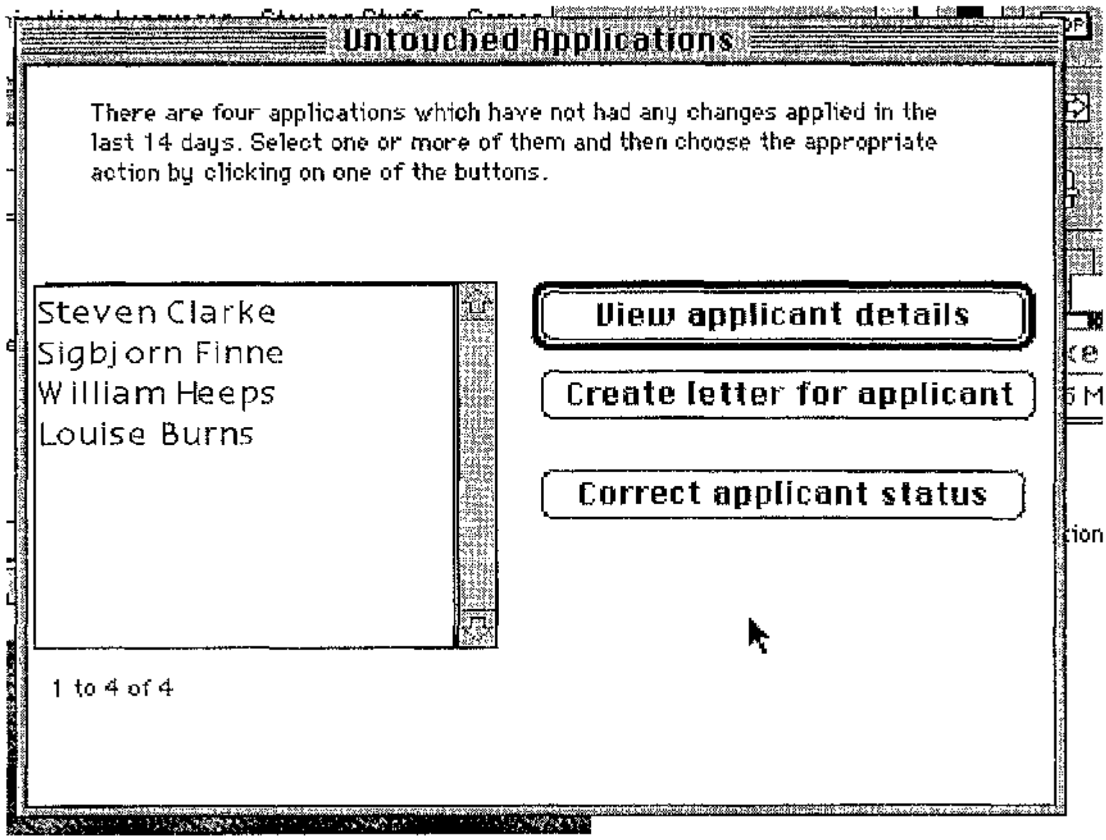


Figure 11. The dialog box presented to the secretary on start–up

3.3.3 Updating known context and the design

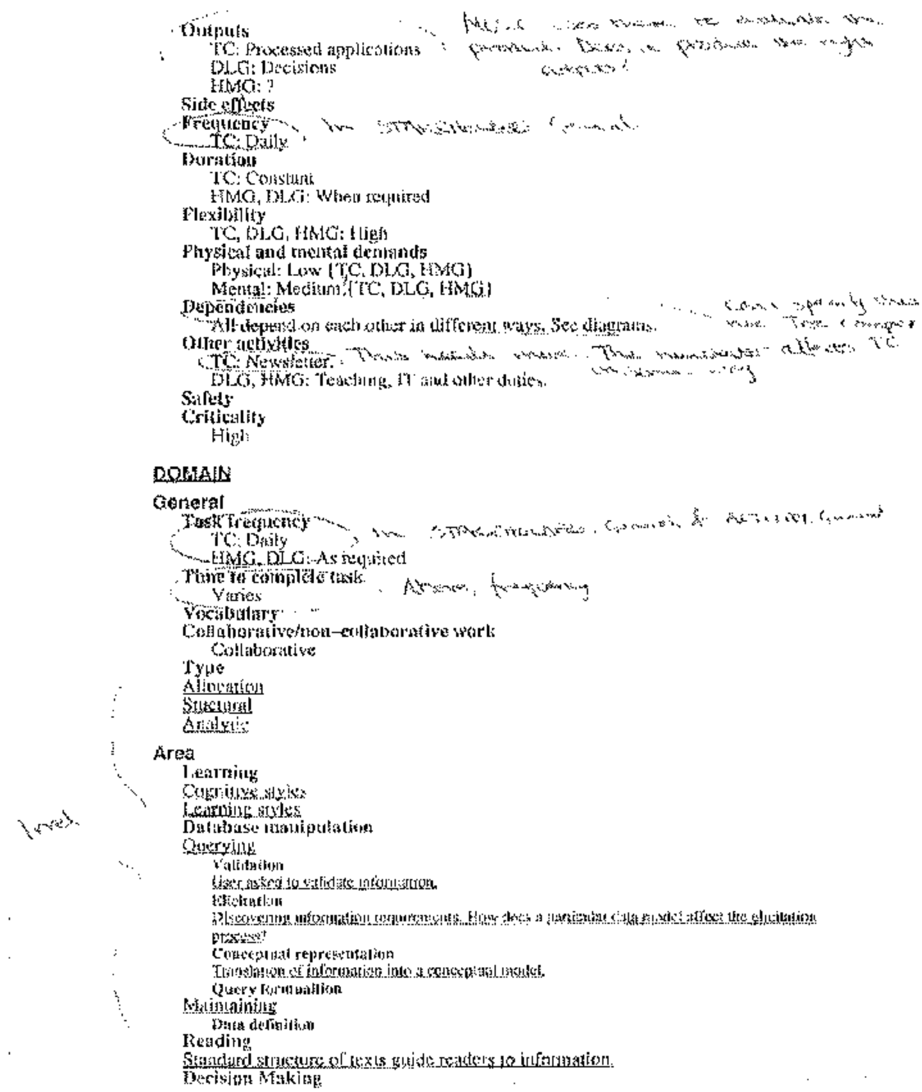
As development progresses, details of the relevant context will change. This is inescapable as design introduces new elements into the user's work, users improve upon their skills, different stakeholders become more or less relevant to the system etc. Designers need to be aware constantly of how the relevant context will change and what impacts these changes could have on the development work done so far. In the IT study, the relevant context was continuously changing. For example, users still had to do their work during the study so their experience with the different tools they used to do their work increased. Unfortunately, the full impact of these changes could not be evaluated with a large degree of certainty because there were no explicit means of showing how the context had influenced the design (or indeed other parts of the context) due to the paper and pen tools used and the difficulties in maintaining and traversing links between the context and the design on paper.

3.3.4 Transforming abstract design ideas into a concrete design

One of the problems that Saskia encountered when attempting to transform the abstract design idea (the design idea was specified using UAN, NUF, QOC and scenarios) into a concrete system was in bridging the gap between the abstract idea and the concrete system. The abstract specification specified such things as the main objects and commands that should be implemented (these are specified in the NUF specification) and the method of interaction between user and computer (this is specified in the UAN specification, and in this case, the scenarios). However, there was no reliable specification of the layout of screens, menu names etc. While some sample screens had been designed, these had not been validated against the contextual information since the contextual information that had been collected was not detailed enough to enable such validations. Saskia had to perform further contextual analyses, at a more detailed level, in order to find out for example, exactly what information the application secretary needs about an application and what the most important information is (Saskia needed to know this, since the amount of information required for each application did not fit on one screen).

3.3.5 Knowing which pieces of context have been used

In the IT study it was difficult to ascertain exactly which elements of context had been used to any extent in the design. Since the context was simply represented on various pieces of paper or committed to memory, being able to record which elements had been used and how they had been used required writing notes on top of the data, or beside it (see Figure 12). This would have become less and less feasible as the context changed, since the way it would be used would also change. It would also mean that it would be difficult to see at a glance how comprehensively the information collected about the relevant context had been used in the design, since newer notes could obscure older notes.



Outputs
TC: Processed applications
DLG: Decisions
HMG: ?
Side effects
Frequency
TC: Daily
Duration
TC: Constant
HMG, DLG: When required
Flexibility
TC, DLG, HMG: High
Physical and mental demands
Physical: Low (TC, DLG, HMG)
Mental: Medium (TC, DLG, HMG)
Dependencies
All depend on each other in different ways. See diagrams.
Other activities
TC: Newsletter.
DLG, HMG: Teaching, IT and other duties.
Safety
Criticality
High

DOMAIN

General
Task frequency
TC: Daily
HMG, DLG: As required
Time to complete task
Varies
Vocabulary
Collaborative/non-collaborative work
Collaborative
Type
Allocation
Structural
Analytic

Area
Learning
Cognitive styles
Learning styles
Database manipulation
Querying
Validation
User asked to validate information.
Elicitation
Discovering information requirements. How does a particular data model affect the elicitation process?
Conceptual representation
Translation of information into a conceptual model.
Query formulation
Maintaining
Data definition
Reading
Standard structure of texts guide readers to information.
Decision Making

Figure 12. Notes written over the checklist

A more fundamental problem related to knowledge about how the context was used involved the degree of trust that was held about the links. Links and relationships between other representations were drawn over the representations. Over time, these links and relationships would change (either the link itself would change, or the source and/or destination of the link would change in some way) in reaction to changes to the design or new knowledge gained about the context of use. As a result of some change, all related elements of design and context required investigation to see if they should also be updated, so that any links would remain valid. However, without manually recording the time that changes were made and the time that subsequent inspections of all related elements were carried out, it was difficult to tell if all the links remained valid and were all as up to date as possible, in terms of change updates.

Also, as the context and the design progressed and changes were made, it became time consuming and more difficult to maintain the links. Sometimes links were not written clearly. When this was the case it was difficult to tell what the link pointed to or indeed, if the link was valid or not (sometimes in the process of writing a link I would change my mind about creating the link. Whatever I had written up to that point would unfortunately still remain on the paper (e.g., link to 'DCM' at bottom of Figure 13).

3.3.6 Propagating changes in the relevant context

As the recorded context changed, its impact on the design was never totally clear. Since there was an amount of uncertainty about the influence of any aspect of context on the design, it was impossible to say for sure that every part of the design that might be affected by the change had been investigated. Therefore, there will most probably have been parts of the design that were overlooked as changes were made to the knowledge about/understanding of the recorded context.

3.3.7 Propagating changes in the design

Similarly, as changes were made to the design, it was difficult to accurately propagate these changes throughout the rest of the design and contextual information. The recorded context was considered for many design proposals, but since the context was either recorded on paper or simply memorised it was difficult to ensure that the full effects of changes were both anticipated and acted upon. Also, as was mentioned before, constructing new parts of the design can identify aspects of the context which need investigation. Introducing some new design element requires further investigation of the context in which the new element will exist. Given a design or part of a design which has been fully specified, it would be beneficial to investigate the context in which that part of design will be used to ensure that the design does indeed fit its context. In other words, it is beneficial to re-examine the context after each design change.

3.3.8 Evaluating how contextually grounded any aspect of the design is

As there was difficulty in maintaining explicit links between the context and the design, it was difficult to evaluate how 'contextual' any aspect of the design was. The best that could be done was to make a rough guess. It was difficult to judge both how fully the context had been used and whether or not any parts of the design needed further contextual evaluation. Figure 13 shows an example of two QOC diagrams that were used to work out how applicant details could be accessed. Links to the context are made by describing the contextual information that influenced the judgement of each option. A cryptic link to the DCM (the original name for the NUF notation) is also shown beside the chosen option in the lower QOC diagram. This example indicates the difficulties in keeping track of all the relationships between context and design and the difficulties in judging how contextually grounded the design is.

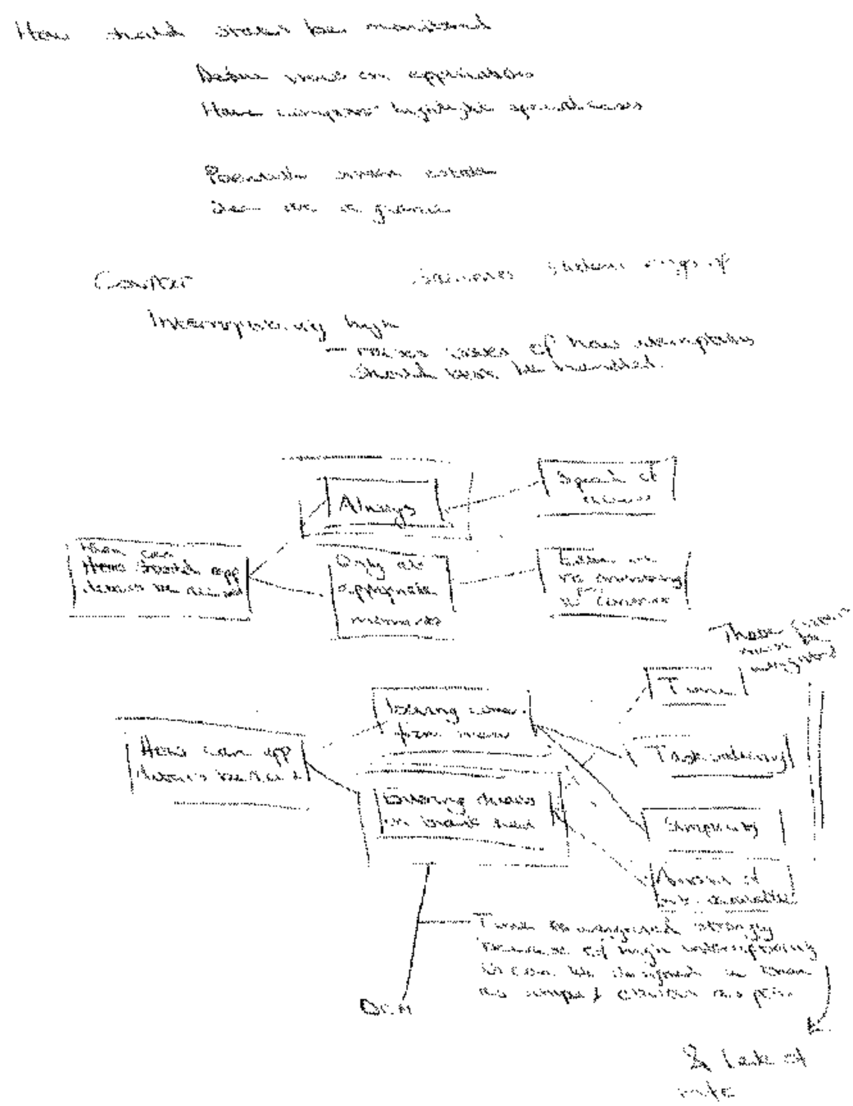


Figure 13. An example original QOC specification

3.3.9 Maintaining consistency

Lastly, it was difficult to make sure that as the recorded context or design changed everything else was kept up to date. In other words consistency was hard to maintain. This was due to the lack of internal links within the context and within the design as well as the lack of external links between parts of the context and design. Also, as aspects of the relevant context changed, it was difficult to ensure that other aspects of the context which may have been related were updated also. For example, as stakeholders experience with the system grew, their attitudes towards computers may have changed. Certainly, the ease with which they would adapt to a new system would certainly change. But since there were no explicit links between these items in the checklist, it was difficult to remember to update them all accordingly.

3.4 Tool Requirements

The problems and activities listed above provided input to the set of requirements for a computer based tool which supports recording and maintaining the recorded context, design and relationships between both context and design. The requirements are described at a fairly broad and general level such that they can apply to any tool which is to be used to help designers effectively and efficiently explore the design space suggested by a contextual analysis. Thus, they do not specify that certain notations or other tools must be used. Rather, for example, they simply say that a means for recording context should be provided. The general nature of the requirements is due to the fact that design teams need the freedom to do what they feel is most appropriate at any stage in the design process. They should not have their hands tied by an over–constraining tool which forces them to use a particular notation or tool against their better judgement.

The M.Sc project carried out by Saskia Koehler also provided some information regarding the appropriate stage in a design project at which a linking tool should be used. This is discussed briefly before describing the requirements for such a linking tool.

3.4.1 When should a linking tool be used in a design project?

Saskia's difficulties in transforming the abstract design ideas into a concrete system suggest that the linking tool should be used to provide input to a further stage of contextual analysis that collects detailed information relating to factors such as the layout, command names, menus and other detailed, low level interface features of the abstract design idea. In this way, validations of the detailed design are also possible by relating detailed design decisions and features to the newly collected contextual information. Other tools or methods may be required to perform such validations. One possible method is Systematic Creativity (Salvador and Scholtz, 1996). Systematic Creativity provides a framework in which designers can use contextual information to validate the way that users control the system (i.e., it validates command names, menus, buttons etc.). A linking tool may also be able to support this level of validation since it should allow designers to specify context and design at whatever level of detail is appropriate and then to validate the design with respect to the context.

While the case study and the M.Sc project provided information relating to the stages in a design project that may most appropriately follow on from the use of a linking tool no information was gained about the stages that would most appropriately precede the use of a linking tool. It is clear that some amount of contextual information needs to be collected before the linking tool can be used, but it is not clear if this is the only requirement. A further case study at Intel Corporation, described in Appendix F describes an investigation into how industrial designers use contextual information in design and how (and if) the linking tool might fit into this environment.

3.4.2 Recording and structuring collected contextual information

Designers need to be able to record the collected contextual information in some form. At the very least, this record serves as a reminder of the important contextual information. If no record is made of the collected contextual information it may be forgotten by the designers and overlooked in the design. The information can be recorded and structured in a number of different ways. This may take the form of graphical models, scenarios, checklists etc.

Designers also need to be able to use more than one type of model, notation etc. to record all the contextual data. As was argued in Chapter 2, and demonstrated in the case study (using checklists to record single attributes of context as opposed to using CCD), different models or theories of context have different scopes which will tend to focus on certain aspects of context while ignoring or neglecting others. If a method is used by a design team such that it neglects some aspect of context that the team have identified as relevant to their project, the design team need to be able to choose another method, notation or model with which to record the information. Consistency problems must be resolved however, if there is more than one model of context that can be linked to.

Designers also need to be able to see to what extent contextual information has influenced the design. At the very least, this could be a simple count of the number of times that a piece of context has been used in a relationship with the design. A more complex visualisation could be employed, for example, changing the colour of the representation of the contextual information, using darker colours as it is used in more links. These indications are useful to designers to enable them to evaluate the influence that the contextual information has had on the design. If the contextual information has not been used much, then the designers may want to look at how they could make better use of it in the design. If the contextual information has been heavily used, then they may be satisfied that the design is well grounded in the context.

3.4.3 Recording the complete design idea

Designers need to be able to record the complete design idea using the tool. Any notation, model etc. can be used to accomplish this. Scenarios, screenshots, prose or notations are all suitable means. If the design idea is not recorded completely, then difficulties will be encountered when the idea is transformed into a concrete system. Indeed, the difficulties may be so great that the feature may not be implemented at all. Saskia encountered a number of difficulties in transforming the abstract design produced as a result of the IT case study into a concrete system. For example, one idea that was considered and described in the scenarios was a 'To-Do' list that would keep track of all the tasks that the applications secretary still had to complete. However, it was never fully specified in the abstract design because of time pressures. When Saskia started to design and implement the concrete system, she tried to design the 'To--Do' list as it was described in the scenarios. However, since there was no detailed description of the feature, she decided to drop it from the final system.

One idea that was considered while performing the IT case study, but that was never pursued was the idea that designers may want to be able to record alternative or multiple design ideas. This arose as a result of specifying different options in the QOC diagrams as a result of some design question. I thought that it would be interesting to specify different design ideas, relate them to the context and then judge each design idea, not just simply by the options specified in the QOC diagram but also by how well the design idea fits the context. One of the benefits of pursuing this approach would be that as the recorded context changed, the judgements of how well suited each design idea is would also change. At some point, changes in the context might indicate that one design idea may now be inferior to another design idea. However, this idea was never pursued.

3.4.4 Linking between context and the design ideas

The literature review and experiences with the IT case study indicate that designers must be able to create explicit links between the representations of the contextual information and the design ideas, if they are to be able to make accurate judgements about the ways in which contextual information has been used in the design. Sections 3.3.3 to 3.3.9 all discuss problems in the IT case study that are related to the lack of explicit links between context and the design.

The way in which links are created is dependent on the representations used. For example, hotspots over a graphical model could be used as anchors for the links. Alternatively individual items in a checklist could be anchors. The key point is that some form of linking between context and design ideas should be supported.

The 'granularity' of links is dependent on both the representations used and the needs of designers. In some situations, designers may need to create links from very detailed parts of the contextual information (e.g., an individual entry in the checklist). If a general, high level model is used to capture the contextual information, then the links themselves may need to be annotated to describe the source or destination of the link in more detail. Alternatively, if a more detailed representation is used, it may be possible to anchor one end of the link in the appropriate piece of information, avoiding the need to describe the link in more detail. A tool should not dictate the granularity of links to users since it is likely that users will want to vary this depending on the links they are creating.

Design ideas need not be fully specified before links can be created. Specifying some part of a design can lead to the identification of further relevant items of context in order that the design can later be specified in more detail once the necessary contextual information has been collected (for example, the dialog box presented to the applications secretary highlighted as relevant contextual information, the way that the secretary identifies applicants – see Figure 11). Therefore, designers need to be able to specify parts of a design so that they can relate these to relevant context, collect information relating to the context and then complete the specification of the design. Stubs or placeholders can be employed so that, for example, if the context suggests a design idea, that idea need not be fully specified before the user can create a link from the context to it. Instead, a placeholder could be created, and the user could link from the context to the placeholder. The system could represent the placeholder in a way that distinguishes it from the rest of the design, reminding the user that that part of the design still needs to be completed.

While performing the case study, I became aware of the difficulties that may arise in updating either the source or destination of a link as a result of a change at the other end. When the source or destination of a link changes, the other end of the link should at the very least be inspected, just to see if it also needs to be changed. If the inspection is not carried out, then the link is not 'reliable'. That is, the link does not represent a valid relationship from the source to the destination, since one end of the link has been altered without the other end of the link being validated. The likelihood of this happening increases when there is more than one link from, say, an element of context. When that context is changed, the designer may change the design specification at the other end of the first link and then as a result of this change, may then start to investigate the links that the design specification is involved in. Without sufficient reminders, designers may forget to investigate further links from the element of context that was originally changed. Hence, a tool that supports the maintenance of links between context and design needs to track the reliability of the links.

3.5 The Literate Development Tool

The Literate Development (LD) tool has been designed and implemented and is based on the general requirements described above. It's main purpose is to enable further research into the issues involved in creating and maintaining a set of relationships between information about a design and its context. The tool has not been designed for use in any real design projects. Instead, it has been designed for use mainly by myself, so that I can carry out further research with it. Hence the tool supports context and design specification notations and techniques that I am most familiar with. The tool supports UAN and NUF editors to describe the design and scenarios and a checklist approach to capture the context. Lastly, a QOC editor is supported, which allows users to describe the design space and the decisions that were taken throughout the design process. As the requirements above suggest, it should be possible to use any appropriate notation, model etc., to describe the design and the context. An appropriate notation or model is one that at the very least allows designers to produce incomplete specifications (so that they can iterate between specifying the context and specifying the design), is easily understood by every member of the design team and one that can be linked to in some way via anchor points. This is by no means an exhaustive definition of an appropriate notation or model. Other features may also be required but these have not yet been identified.

3.5.1 A Computer Based Contextual Development Tool

The tool provides users with the ability to record both the context and the design. A description of the tool will be given first, followed by descriptions of each of the editors. The version of LD described here is a prototype. Chapter 5 describes an evaluation of this prototype as well as changes that were made to LD as a result of the evaluations.

The Linking Tool

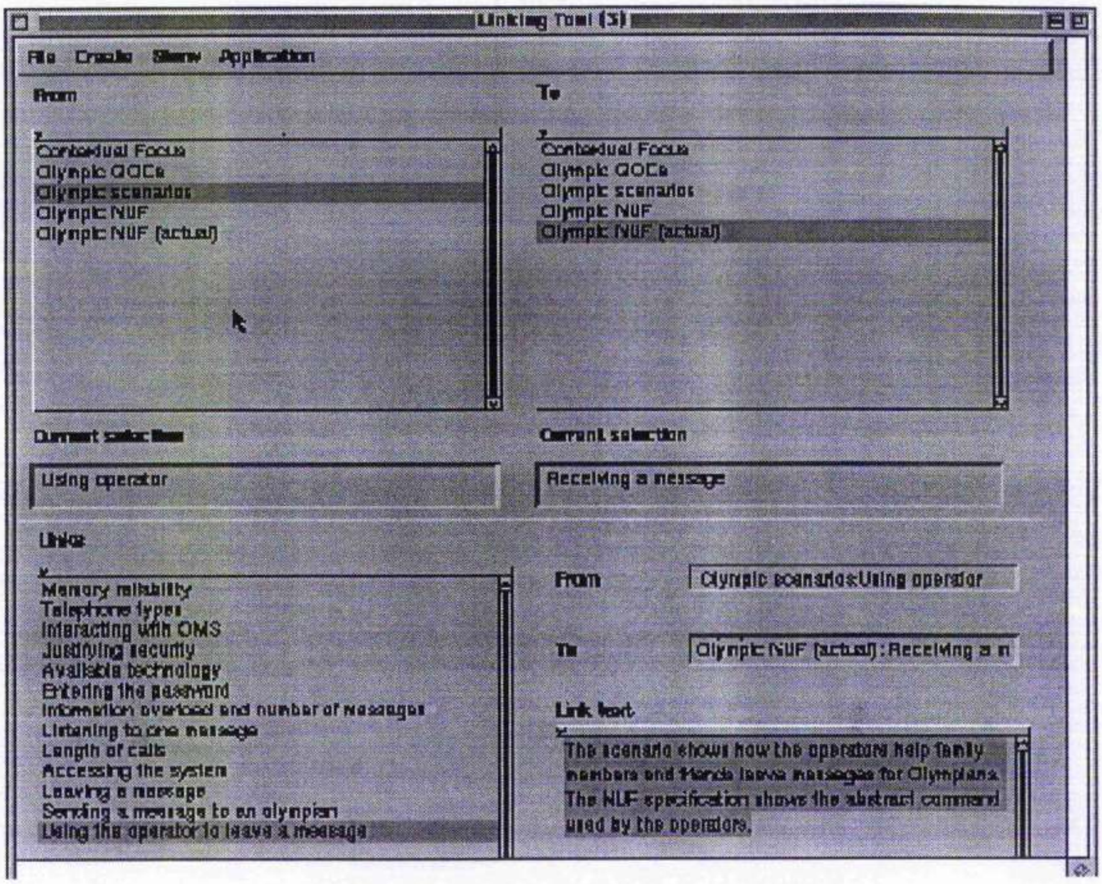


Figure 14. The Linking Tool

Figure 14 shows the main linking tool, the window from which all other functions of the tool are accessed. The menu bar at the top of the window provides access to the functionality of the tool.

Design and context specifications are created using design and context documents. Different documents specify different aspects of the design and the context by using different notations (i.e., UAN, NUF, QOC, scenarios and the context checklist). The user creates a new document by choosing the 'Create' menu from the menu bar. From this menu, the user can choose one of the five different types of document corresponding to each of the five different notations used. When the user creates a new document, they are first asked to supply it with a name. This name is added to the bottom of both of the 'From' and 'To' lists underneath the menu bar. These lists are used to create links between the various editors. The list on the left hand side of the window is used to select the document which will be the source of the link while the list on the right hand side is used to select the document which will be the destination of the link. Above each list can be seen the labels 'From' and 'To'. Before a link can be created 'parts' within the documents must be defined as anchor points for the links . In Figure 14, text fields below the two lists show the names of the currently selected parts for each document.

A link can be created by choosing the 'Create Link' option in the 'Create' menu. If the user makes a change to an object in an document, the system checks if that object is contained within a part which is the source or destination of a link. If so, a window is displayed which lists all the links that the changed object is involved in. The user can reconcile this list with the list shown at the bottom of the main linking tool, which contains all the links that have been created so far. The user can scan this list, searching for those links that are involved in the recent change. By clicking on each of the links that are involved, the user can see the details of the link in the bottom right panel of the main window. Here, two text fields show the source and destination of the link. Sources and destinations are listed in the form <NameOfDocument>:<PartOfDocument>. Therefore the user can identify both the document that is involved and the part of the document.

If a designer wishes to evaluate the cost of changes to a particular document, they can click on the name of the document in the left hand list (the 'From' list in the main linking tool) and choose the 'Show links for document' option from the menu. This results in a dialog box which lists all the links related to that document. This gives an idea of the potential ramifications of changes (for example, if there are lots of links related to a UAN specification say, there will potentially be a lot of work following changes to the UAN specification).

The user can also view the number of links between documents in the form of a table. By choosing the 'Show all links' option, the user is presented with a table similar to Figure 15

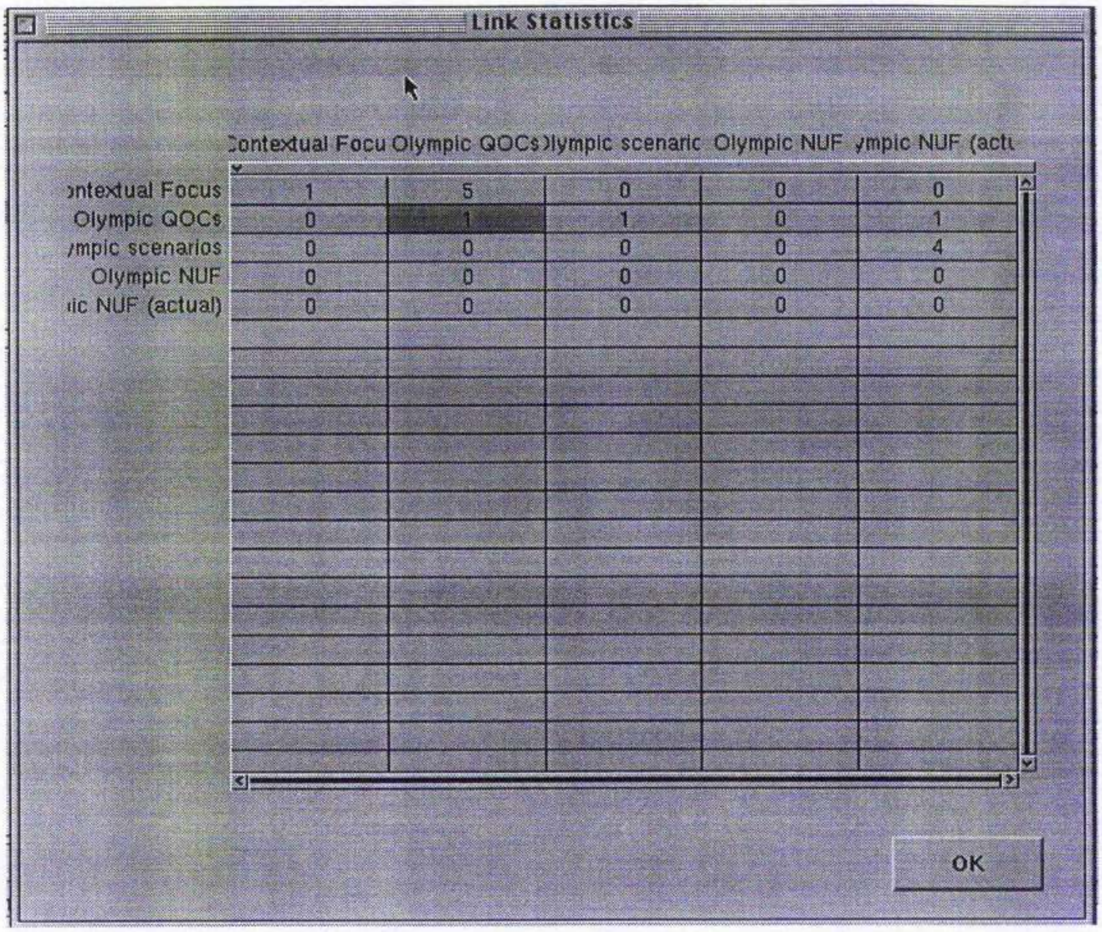


Figure 15. A table showing the number of links between different representations

This table presents a list of all the documents on the rows and columns of the table. The rows represent the source of links while the columns represent the destination of links. Hence, from the table above we can see that there are 5 links from the contextual focus to the Olympic QOCs but none in the other direction. This table gives users a global view of the design and can direct them to areas which may need further work (areas for example, where there are no or very few links).

3.5.2 The Editors

Editors have been designed and built for the QOC, UAN and NUF notations as well as for the contextual checklist. Each of these editors will be described in turn. Common to each editor is a panel which lists all the parts that have been defined for the document. Every editor has a panel identical to that shown at the bottom of the QOC editor below[5] (Figure

---

[5] Notice that in the following diagrams, these panels are shown only in the QOC, contextual focus and scenario editors.

16). Two lists are shown, labeled 'All collections' and 'Contents of collection'. The 'All collections' list shows a list of the names of parts of the document that have been created in this editor. The contents list shows the objects that are contained in the currently selected part. These parts are used as the source or destination of links. The three buttons to the right hand of the lists are used to add objects to a list, create a new list or remove a list respectively. Each editor has an identical panel in the lower area of the window. Hence, links are created in the same fashion across all editors. The rationale behind creating links from groups of objects rather than simple objects is that users can vary the granularity of the links to suit their own purposes rather than have it dictated to them. Allowing the users to create groups of objects from which they can create links gives them the opportunity to create a link from or to a whole specification or a small part of a specification.

QOC Editor

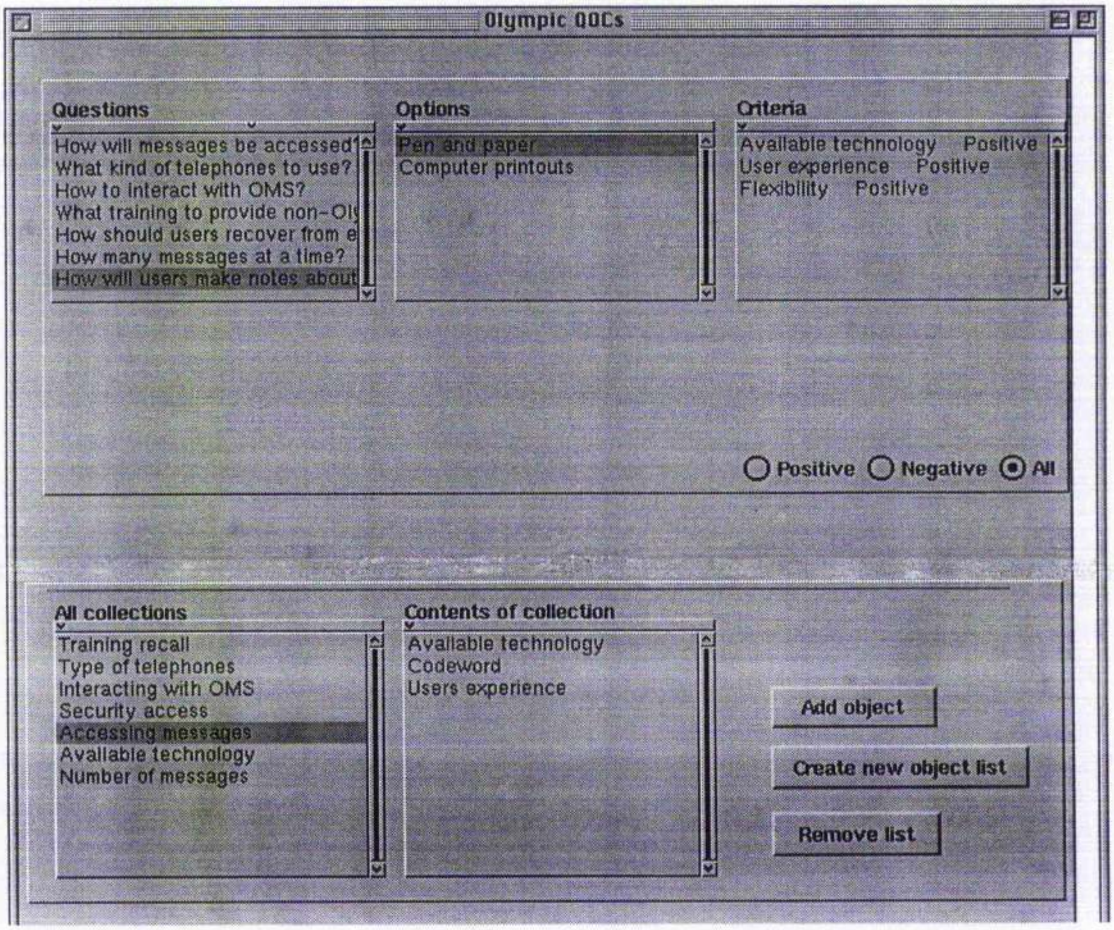


Figure 16. A QOC editor

The QOC editor lists all the questions that the user has had to work on in this particular design at the top of the window. When one of the questions is selected, the options and criteria lists (middle and far right respectively) display the corresponding options and criteria. When the user selects one of the options, the criteria are modified to show if they are judged positively or negatively against the selected option. The radio buttons underneath the criteria list lets the user specify that they only want to see all those criteria that are judged positively or negatively. Alternatively they can choose to see all the criteria, as is the case in the above example. Menus within each of the lists allow users to add, rename and remove questions, options and criteria with further menu options in the criteria menu allowing users to judge criteria against the selected option.

UAN Editor

The UAN editor is simply a text editor in tabular form. An example is shown in Figure 17. Notice that the common 'linking panel' is not shown here.

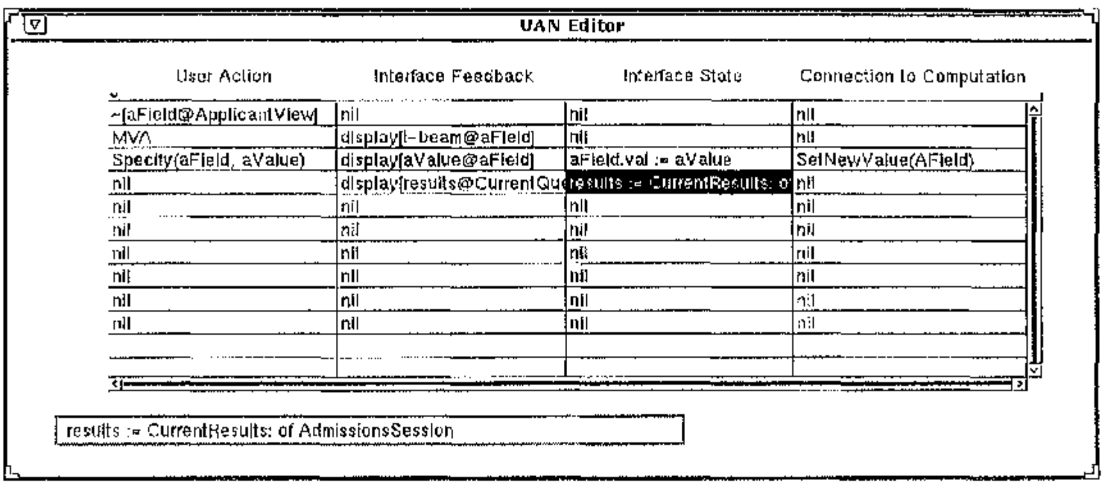


Figure 17. A UAN editor

There are four columns (User Action, Interface Feedback, Interface State and Connection to Computation) and an arbitrary number of rows. The user need simply click on one of the cells to select it (it becomes highlighted) then enter the text in the text entry box at the bottom of the window. When RETURN is pressed, the text is updated in the cell. No UAN syntax is expected – users can use whatever syntax they prefer. This is the way that the originators of the UAN notation prefer it to be used (Hartson *et al.*, 1990).

NUF Editor

The NUF editor is similar to the Smalltalk browser. An example is shown in Figure 18.

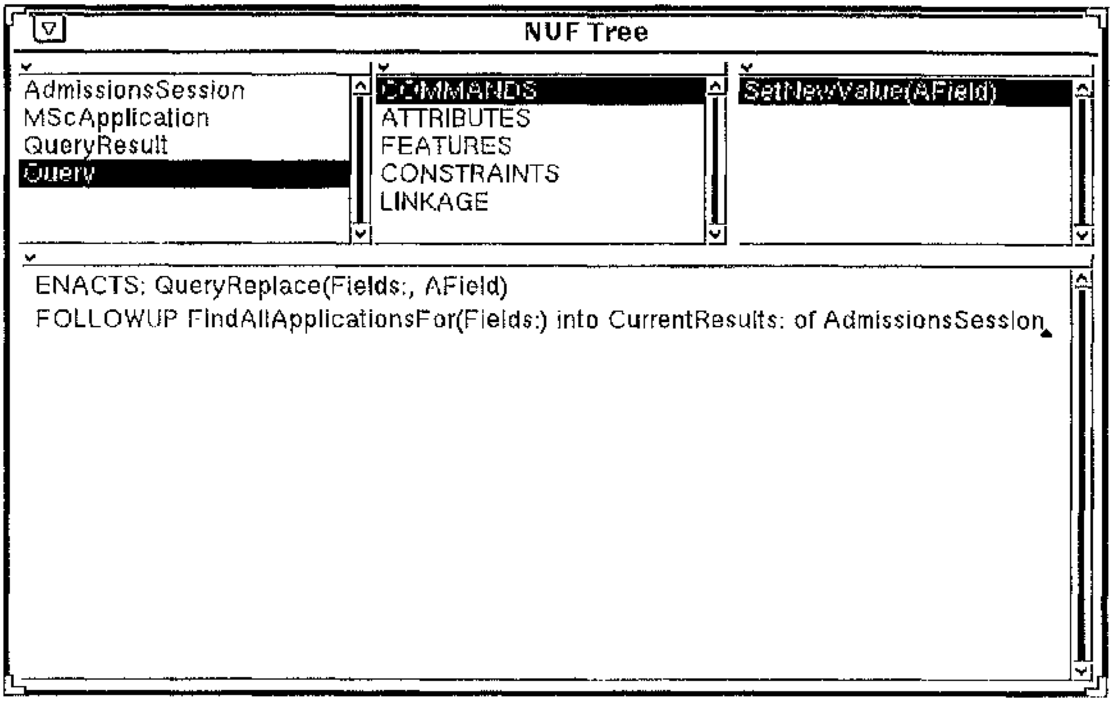


Figure 18. An NUF editor

NUF classes are entered and displayed in the top left hand viewing pane. So far, classes for AdmissionsSession, MScApplication, QueryResult and Query have been defined. When a class is selected, the middle viewing pane shows the frames for which slots can be defined. For each class, COMMANDS, ATTRIBUTES, FEATURES, CONSTRAINTS and LINKAGE slots can be defined. Selecting one of the frames allows the user to add a slot in the right hand viewing pane. The name of the slot is displayed in this pane, while the rest of the information (type, command sequence etc.) is entered in the lower viewing pane. The lower pane is a basic text editor. Changes can be made in here and accepted by choosing the accept menu option. The NUF editor does not demand that slots are defined. Once a slot is named, the editor will enter the text 'Undefined slot' in the definition of the slot which will be displayed in the lower pane until the user updates the definition of the slot. This is in keeping with the nature of the NUF notation (Cockton *et al.*, 1996) which does not enforce premature commitment by designers.

Describing the context

The contextual focus window (see Figure 19) provides users with a means to navigate and maintain the contextual focus (the hierarchy of contextual items that are relevant to the current project).

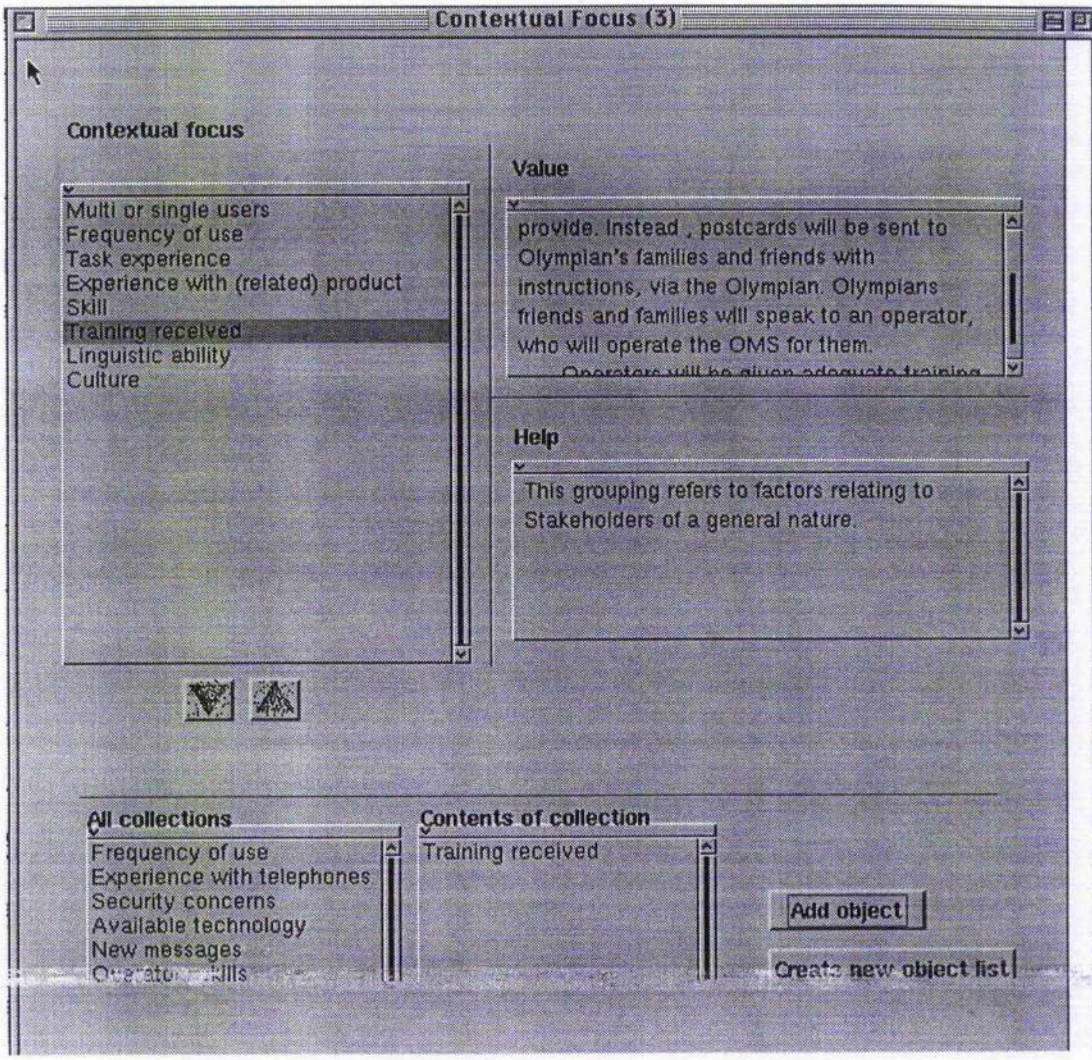


Figure 19. The contextual focus editor

The list on the left hand side shows the current position within the hierarchy of contextual factors. The buttons below this list allow users to move up or down in the hierarchy. When one of the items in the list is selected, the value text field on the right hand side shows the notes that the users may have entered for this particular item. If no note has been entered this field is simply blank. Below the value field, the help text field provides some explanatory text about either the current position in the contextual hierarchy or the currently selected item. As is the case with all other editors, the panel allowing users to create groupings of objects is located in the lower half of the window.

The Scenario editor

Finally, the last editor is the scenario editor (see Figure 20). A text editor is provided at the top of the window where the user can type in any text they like. Choosing the create scenario menu option allows them to designate the text in the text editor as a scenario. The user is prompted to name the scenario. The name is then placed in the scenarios list (the right hand list beneath the scenario text). The anchors list on the left hand side shows all those named portions of text from which a link can be created. The user highlights a section of the text and names it as an anchor. Then, these anchors can be used in a link by adding them to a group of objects in the same way as the other editors. When an anchor is changed in the text editor (for example, by removing a section of the anchor) the computer checks to see if the anchor is involved in any links, just as in the other editors.

No distinction is made in the scenario editor between scenarios that are used to describe context and scenarios that are used to describe designs. This provides users with the freedom to do what they wish with each scenario. Presumably if they wish to distinguish between different types of scenarios (e.g., envisionment scenarios, implementation scenarios), they could devise an appropriate naming convention.

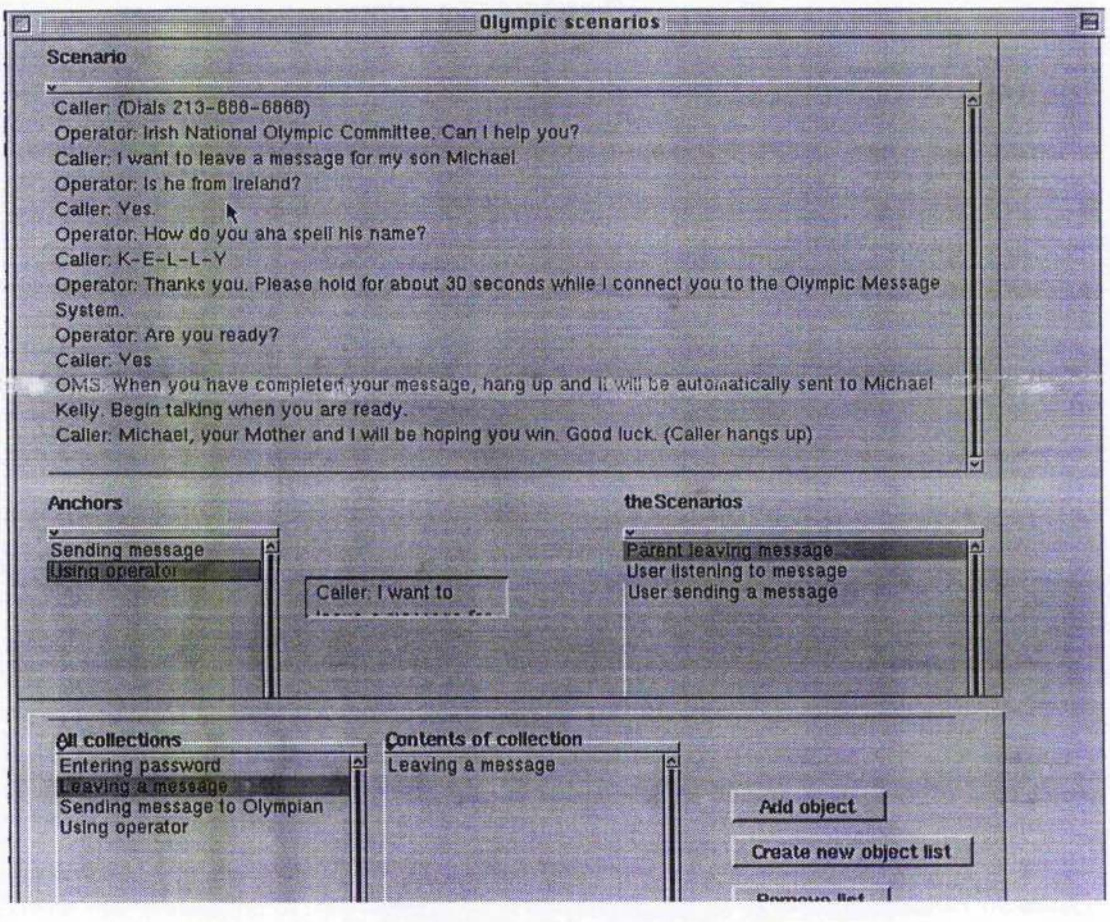


Figure 20. A scenario editor

3.6 Chapter Summary

This chapter has described the IT case study and LD, a tool which supports the exploration of the design space suggested by the contextual data. LD allows users to demonstrate design concepts and link from these concepts to the contextual information that suggested them. Therefore, judgements can be made about the validity of the concepts with respect to their fit to the context. The tool helps users maintain the consistency of the design and the context by easing the propagation of changes. It is hypothesised that the tool helps users keep on top of the evolving design and context descriptions by highlighting areas of context or design that may be affected by any changes made

The next chapter describes three case studies in which LD was used to describe three different designs. The case studies present an informal evaluation of LD, and an investigation into the advantages offered by the tool.

4. Chapter 4 Using LD

4.1 Introduction

Chapter 3 described LD, a tool that was built to support the recording of abstract design ideas, related contextual information and the relationships between them. In this chapter, three different case studies are described in which LD was used to maintain the relationships between three different designs and the related contextual information. The case studies present examples which describe how LD was used to specify design ideas.

4.2 LD Case Studies

Chapter 3 described the IT case study. This was a study of the process by which applications for the M.Sc in IT course, run by the Computing Science department, are processed. The study was performed using pen and paper. It was used to identify the required features for a computer–based tool that records and maintains the relationships between a design and its associated context. The case study identified a set of features that must be supported by the tool and these features guided the tool design. In this chapter, three case studies are described which make use of this tool. Each case study involved creating representations of a particular design and its associated contextual information, together with the relationships between the design and the context. Throughout each study, the primary focus was on examining the benefits and drawbacks that the tool offered. Particular attention was paid to any feature of a design, its context or the relationships between the two that the tool made clear or any such feature that was difficult to model using the tool.

The first case study replicated the original IT study. This time, instead of creating representations on paper and attempting to maintain the links manually, the computer–based tool was used for these tasks. By replicating the original study in this way, it was possible to compare representations and links created in the computer–based tool with representations and links created with the paper–based tools. This study was in effect an informal comparison between the methods used for the original paper and pen–based study and LD.

The second case study involved representing the tool itself and the context in which the tool is used. This was a useful exercise both to see if the tool could be used to describe itself and to determine if LD could be used to model a system from scratch. No contextual analysis or in depth analysis of the tool and how it would be used had been performed. The design of the tool was based solely on the experiences gained in the original IT study. In order to specify LD, an analysis of the context of use of LD would have to be performed and modelled using the tool. In effect, this case study would proceed as if the design of LD was a new project. Hence this case study would investigate informally how succesfully LD could be used in a project, from start to finish.

The third study examined the Olympic Messaging System (OMS), a system designed and built by IBM for the Los Angeles Olympics in 1984. The OMS allowed olympians to leave messages for friends and relatives and vice versa. Olympians could use kiosks around the Olympic Village to leave messages while relatives and friends from all over the world could phone a special number to leave a message. One of the most comprehensive descriptions of the design of the OMS can be found in Gould *et al.*, (1987). This paper was used as the source of information for all aspects of the project, from initial analysis all the way through to design and implementation. This study was a 'reverse engineering' exercise. It demonstrated that the tool could be used to analyse an existing system.

4.3 The LD Case Studies

Each case study involved creating and maintaining representations of context, designs and relationships between these different representations in a *post-hoc* fashion, for different systems which had already been implemented, but which had been specified and described to varying degrees. For example, the second case study involved LD itself, which had not been specified or described to any great detail prior to its implementation. In contrast, the first study involved the IT system which had been described and specified in great detail. Performing the case studies post–hoc allows a comparison to be made between the original design and the new design after performing the study. Such a comparison can identify any new insights that were gained through performing the study and using the tool.

4.3.1 The IT Study

The first case study involved creating and maintaining design and context representations with respect to the IT system which had previously been designed and implemented. A relatively comprehensive contextual analysis had been performed prior to and during the design of the IT system. The context was represented using the checklist described in Appendix A and the tools of the Customer Centred Design approach, described in Holtzblatt and Beyer (1993). The design was specified primarily using scenarios and the NUF notation. To a lesser extent, the QOC and UAN notations were also used. All the different specifications had been created using paper and pen.

In order to make a fair comparison between the original representations and the representations created using LD, no further contextual analysis or design was performed. The same design and context representations were used in LD as had been created on pen and paper. Hence any new insights gained by using LD could more confidently be explained with reference to the tool, rather than by referring to any extra analysis of the context or the design. Additional knowledge may have been gained from performing the original study in the first place but I tried to make sure that any new knowledge would not be represented in the tool by using the original representations only.

To begin with, the various context representations of the IT system were transferred from paper to LD. The information represented in the Customer Centred Design models was represented in the contextual checklist in LD, since Customer Centred Design modelling tools had not been implemented in LD at that stage. For example, the models represented such details as *roles* and *task allocation.* The checklist was examined to see if these factors were already included and to determine where they should be placed if they were missing. Many of the factors that the models represented were already present in the checklist, so the checklist did not require a major revision. This meant that for the IT study, LD could only represent the context in checklist form. This was not considered a major problem however, since all of the relevant context had been represented in the tool and the Customer Centred Design diagrams were still available to consolidate the contextual representations used in the tool.

As more and more contextual information was added to the tool, more attention was paid to how this information could be related to the design. The relationships between the context and design had been difficult to represent in the original study. Using paper and pen only it was difficult to maintain the different relationships, particularly as the representations evolved. Links were written over the representations (see Figure 5 in Chapter 3), sometimes not as clearly as they should have been. The effort required to maintain the links (keeping all the diagrams neat and clear after every change) meant that only the simplest, most obvious relationships between context and the design had been recorded.

In contrast to the original study where links were created and maintained by myself using paper and pen, LD handled the maintenance of the links. Hence I was more inclined to add links to the tool since each link added would not increase my workload in terms of maintaining the links as had been the case when doing the study using the paper and pen tools. Therefore I created more links between the representations than I had done with the paper based study. I examined the representations to see what new links I could add. For example, one of the new links concerned the environment and the shared resources that the applications secretary worked in. Since the secretary does not have a printer of her own, but has to share it with two other secretaries, the automatic creation and printing of letters could actually create more problems. Instead of ten letters being printed in one batch, each letter would take its place in a queue with print jobs from the other two secretaries. Hence the applications secretary would have to spend some time sorting out all the print jobs to retrieve her letters. Some of the letters could be lost either if the applications secretary does not find them in the pile of print jobs or if the other secretaries mistakenly picked them up with their own print job. Figure 21 shows some of the links for the IT case study. The link indicating the influence of the shared environment is selected.

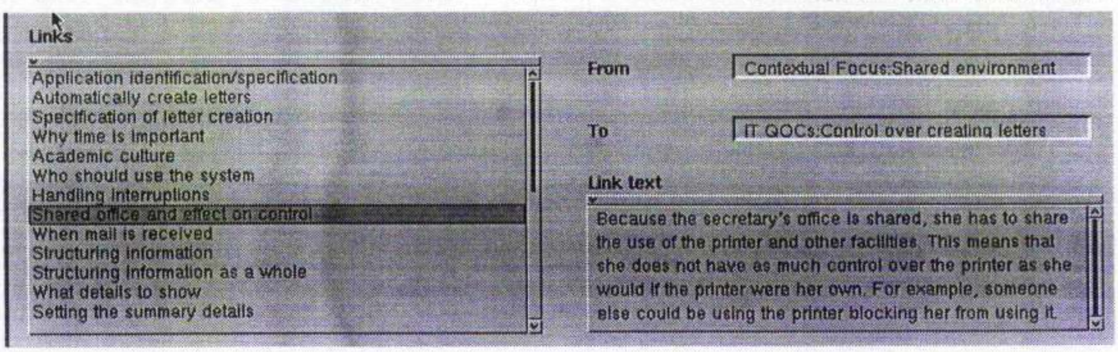


Figure 21. Link, indicating influence of environment on printing.

In my opinion, since I was relieved of the burden of maintaining the links, I was encouraged to add more links such as the one above. By doing so, I was encouraged to consider the context in more detail. As a result I gained a better understanding of the context. In the example given above, I created a link between the shared environment and the printing of letters in batches. In the original study I had observed that the applications secretary normally tried to perform various actions in batches, such as updating applications, filing and creating and printing letters. I originally ascribed this to the fact that the applications secretary was using two different computer–based applications and did not like to have to switch between the two applications. Indeed, when I asked the secretary about this at the time she agreed that this was the reason why she worked in batches. However, it is clear from the link created above that the secretary's environment also constrains the way she works such that creating and printing letters in batches makes it easier to retrieve the letters from the printer. In other words, by examining the representations in detail and looking for more links I learnt more about the way that the environment influences the way the secretary does her job. I believe that the effort involved in looking for more links between the different context and design representations increased my understanding of the context. The full reasons for the secretary working in batches may not have been clear in the original interviews since it may not have been apparent to the secretary that working in batches made it easier to retrieve letters from the printer. The reason for working in batches may simply have been put down to the fact that different computer applications were being used. Or it could be that I did not pursue this further in the interviews, being satisfied that the different computer applications used accounted for the method of work. Whatever the reason, I believe that using LD, which removed most of the burden of maintaining a list of links, encouraged me to investigate the context and the design further to identify more links.

Links involving the different design specifications (i.e., the UAN and NUF specifications) were added. Using LD in this way highlighted inconsistencies between the original specifications. For example, in the UAN specification of how the secretary starts the IT system, a reference is made to the command in NUF that starts the system. In the paper–based representations these commands were inconsistent with one another. The UAN specification made a reference to an 'OpenDatabase' command while the command in the NUF specification was StartUpForNameWithPasswd. I only became aware of these inconsistencies when I used LD. One of the reasons for this may have been that while performing the IT study, the difficulties experienced in working with lots of different pieces of paper and trying to update representations described on different pieces of paper meant that I wasn't aware of inconsistencies between representations. Perhaps when I was creating the UAN specification I couldn't find the NUF specification of the command that is used to start the system up. I may have decided to write a command name in the UAN specification that I thought matched the one in the NUF specification. In constrast, while using LD, it is difficult to misplace individual specifications (in the worst case, specifications may be hidden behind other windows) and so easier to maintain consistency between representations since it is easier to check the names of commands written in the representations (Figure 22 and Figure 23 show the UAN and NUF specifications from LD respectively). Also, when creating a link between design specifications, the user has to inspect the specifications in detail in order to specify the source and destination of the link. This detailed examination of the specifications may also help enforce consistency between representations.

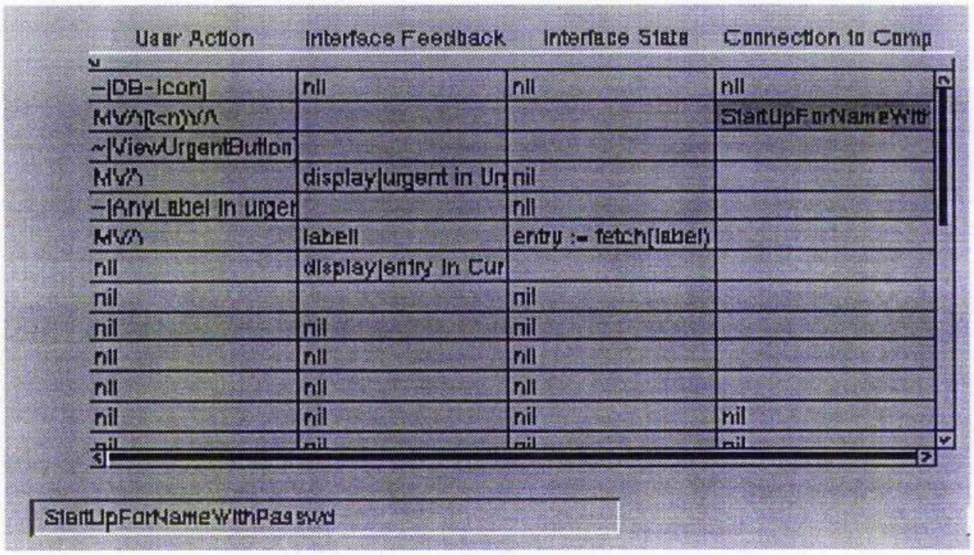


Figure 22. The UAN specification of the means by which the secretary starts up the IT system.

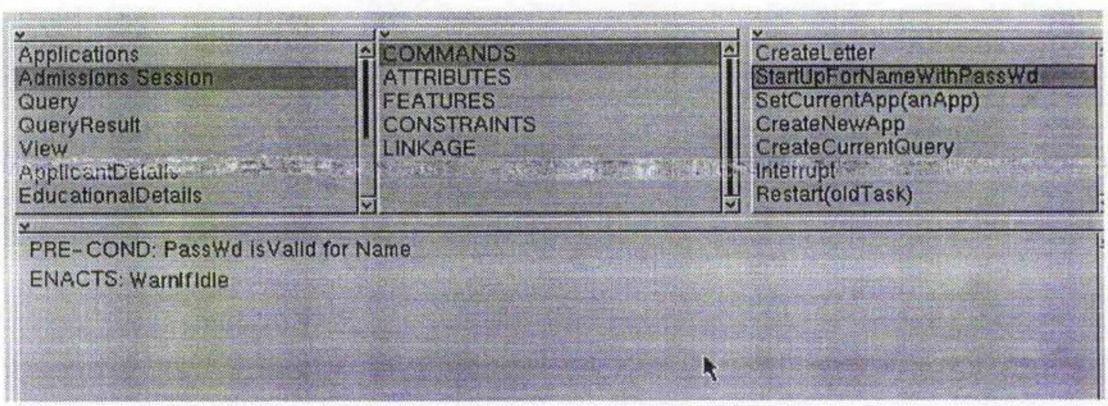


Figure 23. The NUF specification of the StartUpForNameWithPasswd command.

Instead of waiting until all the contextual information had been represented in the tool before thinking about how this information related to the design, relationships between context and the design were created as the contextual information was added to the tool. Whenever a new relationship seemed appropriate, it was created and added to the tool. This approach was taken to avoid the danger of forgetting particular relationships, which is more likely if relationships between context and design are investigated only after all the context is represented, which can be a considerable time after the potential relationship was highlighted. However, this approach meant that the different design and context representations were added to the tool piecemeal. Whenever an appropriate relationship became clear, it was often necessary to create the representation of design that the contextual representation would relate to, in order to add the relationship.

A number of iterations were performed, transferring contextual information from paper to the tool, identifying potential relationships, transferring design representations from paper to the tool if necessary and recording the relationship and using the relationship to identify what context should be transferrered in the next iteration. This process was repeated until all of the context and design had been transferred from paper to the tool. Further inconsistencies similar to those mentioned above were found. In one of the scenarios that were created relating to the IT system, the method by which the secretary enters information relating to individual applications was described. In the description, it was identified that the applicant's first and second names are entered into the database. However, in examining the link from the scenario to the NUF specification which described an individual applicant, it was discovered that the NUF specification made no distinction between first and second names. The NUF specification was found to be incorrect in other respects also. In another scenario, the secretary is described using the 'Show summary details' feature of the IT system. This feature was not specified at all in the NUF specification. Again, this was not realised when the paper based specifications were used but did become clear as the specifications were created in LD and relationships between the context and the design were being identified. In LD, since I was more inclined to explicitly create relationships between contextual information and designs, I examined the different context and design specifications in greater detail. In my experience, this leads to more consistency between the representations.

This study has shown how the tool could be used during a real project, from initiation through to completion. The next study shows how the tool could be of use in the situation where a design has been produced and implemented, but now needs to be properly documented, or made sense of. Some of the experiences with using LD in the IT study were used to develop and improve the tool in the next study.

4.3.2 LD Study

The second case study involved the design of LD itself. In this case, neither the relevant contextual information nor the design had been formally specified, since the original IT case study, using the paper and pen tools, had been the source for the design of the tool. The original IT case study had used current methods for contextual design, all of which were described in Chapter 2, and had attempted to maintain the relationships between context and design with paper and pen based tools. The difficulties in performing these tasks provided an initial set of 'requirements' for LD. The tool was implemented in response to these requirements. Hence there were no contextual diagrams or design specifications that could be referred to as there had been in the first case study. In contrast to the first case study which at times resembled a simple data entry exercise, the second case study was more similar to a reverse engineering exercise. Hence this study represents a useful investigation into the way that the tool can be used to investigate a current system and determine how well it fits its context.

In contrast to the first case study, this study began by reflecting on the design problems that had been encountered during the design of LD. Each problem was reflected upon and represented using the QOC editor in LD. This opened up the design space for LD and by so doing, helped identify some contextual information that may have played a part in suggesting design options or criteria by which to judge other options. These contextual factors were recorded using the contextual focus editor. Similarly to the first case study, the process of creating the different representations was again iterative. This time however, the iterations involved specifying design problems in the QOC editor and using these specifications to identify relevant contextual information. A number of design problems were identified that suggested modifications to LD. These were

- *Deciding what parts of a document could be linked to and from.* In the original version of LD (which preceded the version of LD described in Chapter 3) users could only create links to or from individual parts of a specification. For example, they could only link from one option in a QOC specification to one command of a NUF object. Or they could only link from one item of context in the contextual checklist to one cell in a UAN table. Being forced to think about the design by describing design problems in the QOC notation made me consider other options. One other option which had become clear while performing the first case study was that it would be more useful to link between any number of QOC options, NUF objects, cells of a UAN table or items of context. This would allow the user to specify the level or granularity of the link. If the link was fairly high level, it may involve a number of contextual factors having a general impact on the design. If it was at a lower level, it may involve one or two contextual factors having a more narrowly focussed impact on the design. The option of allowing users to link from and to more than one individual 'object' in any specification supports links of different granularity.
- *Identifying unused contextual information.* While using the tool to describe the IT case study, more links than had been created on the paper representations were added to the

tool. Because of the number of links, it was difficult for me to keep track of all the contextual information that had been used in a design. When there were only a few links it was fairly easy to remember what pieces of context had been involved in the links. However, as more links were added some other means of keeping track of the unused context was required. I thought of two different options, one which involved highlighting the context that had been used in the contextual checklist so that the unused context could be seen at a glance (it would not be highlighted). The other option involved adding a function to the tool that the user could invoke whenever they wanted that would list all those contextual factors that had not been used in any link. This would save the user from having to traverse the contextual checklist to identify the unused context. I decided to implement the latter function as it would provide the user with a simple list of unused factors as opposed to them having to create the list by inspecting the checklist for un–highlighted factors and it would be easier to implement.

- *Identifying duplicate links.* I tried to ensure that as I added links to LD that I was not adding a link that was a copy of some other link. I did not want to add duplicate links since these, when between some part of the context and a part of the design, make it look as if the context has influenced the design in more ways than it actually has (since two links are used to indicate the same relationship). Duplicate links can also increase maintenance effort. For example, if the source of a duplicate link is changed, the system will indicate to the maintainers that the destination has to be inspected more than once, which is a waste of effort. I was also concerned with ensuring consistency. If the same part of the context is used more than once, it is important to ensure that it is used consistently. It should not be used to justify one design feature and reject another similar feature. I wanted to find a way to identify these links also. I decided to create a function that the user could invoke at any time that would inspect all the links and list those links that originated from the same anchor point. I decided against only listing those that had identical sources and destinations since this would omit all those potentially inconsistent links that had the same source but different destinations.

- *Identifying affected objects.* In the first study of using the tool, the only way that I could see the effect of making a change was after I had made the change. Whenever I changed an object in any representation, if that object was involved in one or more links, LD would display a dialog box with the list of links that I should inspect to see if further changes are required in other specifications. However, it was clear that it would be useful to know this information before the change was made so that the effort involved in making the change could be evaluated. I decided to implement a function that the user could use to display the relationship paths[6] that originate from one object. The user could use this function to evaluate a change to an object before making that change.

By considering these problems, (which had emerged through using the tool to describe the IT study) the context of development of LD was considered carefully and recorded in LD using the contextual focus editor. Most importantly, the fact that LD was developed in order to gain insight into representing relationships between context and design was recorded. This had an effect on some of the design decisions that were made, for example, choosing to represent QOCs textually rather than in diagrammatic form. Since LD was not developed for use in a real design project but for learning about representing relationships, the focus was on getting something working fast, rather than getting something working well. Hence, QOCs were represented in textual form rather than in diagrams since this could be implemented quickly. So, the context of use of LD in a commercial development project was not considered during this case study as this had not influenced the development of LD. A further case study would be required to determine the changes that would need to be made to LD to enable it to be used effectively in a commercial project.

However, LD was modified to address the problems described above. Extra functions were added to the tool which address the problems. The 'Show' menu was modified to provide access to the new functionality. Figure 24 shows the new menu.

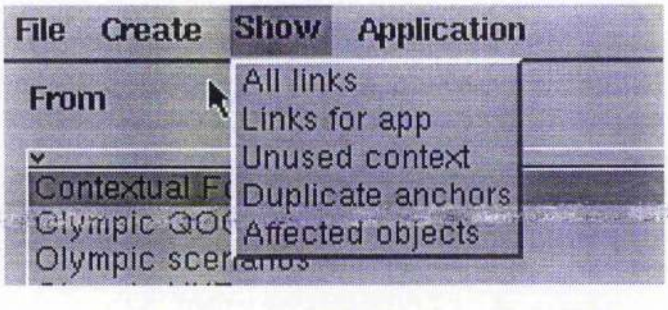


Figure 24. The new 'Show' menu providing access to the new functionality.

The new items added to the menu are.

- *Unused context*. When the user selects this menu option, LD inspects all of the links to determine which elements of context have been used in a link. From this information it

[6] If some object A is related to another object B which in turn is related to object C, then a relationship path exists between A and C via B. In terms of evaluating changes to objects, it is important to realise that as a result of changing object A, both B and C may potentially need to be updated, B because it is related to A and C because it is related to B.

then calculates those elements of context that have not been used (basically the difference between all the contextual factors and those factors that have been used in a link) and displays them in a simple dialog box.

- *Duplicate anchors*. When this menu option is selected, LD goes through all the links and retrieves those links that share the same source. These links are then displayed in a dialog box similar to the bottom half of the linking tool. The user can then inspect each link and determine if the link is a duplicate link or if it has been used consistently or not.
- *Affected objects*. When the user selects this option, LD calculates and displays the relationship paths that originate from the currently selected part of the currently selected document. This provides the user with a means to evaluate the cost of making a change to the currently selected part of the current document.

The new commands added to the tool support the 'goals' that arose through the grounded development of LD. By 'bootstrapping' LD and focussing on its development other issues and new commands for LD were identified.

Being both the designer and implementor of the tool, I had detailed knowledge of its design and implementation. Therefore the representations created in LD could be said to accurately reflect the actual state of affairs. However, being so closely involved in the design and implementation of the system means that only an *inside–out* view of the system was obtained. It is likely that another person performing the same analysis would create a different collection of design and context representations and of the relationships between the context and the design, since their view of the system would be *outside–in*. An interesting study would have been for another person to perform this study and to see what insights they gain into the background of LD through using LD. Through thinking about the relationships between the context and the design of LD and focussing on representing these relationships, it is likely that someone who was not involved in the design and implementation of LD would have a clearer picture of the way that context and design interacts in this system, than someone who did not focus on the interaction of context and design. Therefore, this study does not clearly demonstrate the benefits of LD in a reverse engineering exercise. What this study does demonstrate however, is that the tool provides a mechanism with which to collect, record and structure ideas and that new ideas and design features can be identified by recording original ideas and problems.

4.3.3 The OMS Study

In this, the third and final study, a system is reverse engineered from a paper that describes both the system and the design of the system. The system is the Olympic Messaging System, built and designed by IBM for the 1984 Los Angeles Olympic Games. The system allowed Olympians to receive and send messages from and to family, friends and other Olympians. It was used extensively throughout the Games by many of the Olympians, their friends and their families. Hence, this study represents a realistic example of reverse engineering.

The only access to the design of the OMS system was from a paper by Gould *et al.* (1987). The paper described the system itself, how Olympians would interact with the system and how the system was designed. According to the paper, the OMS design team used scenarios to work out typical tasks and how the OMS would support them. For the case study, the method employed by the OMS team was replicated, so copies of the scenarios published in the papers were made and entered into LD. These scenarios were used to identify some of the relevant context and issues. Just as in the previous case studies, the construction of the context and design specifications was iterated, this time via the scenarios. However, unlike the two previous case studies which had focussed on the contextual information and had used that to develop the design, this study focussed on the design as represented in the scenarios. Thus the scenarios, which described typical tasks and how they would be carried out with the OMS, identified some of the relevant aspects of context. The different focus encountered in the OMS study can be explained by noting that at the time the OMS system was designed, contextual design was not as widespread as it is now. Typical methods of design in the early 1980s focussed on tasks that systems would have to support. Through an analysis of these tasks, a system would be designed such that the tasks were adequately supported. Due to this task analysis focus, the paper describing the OMS and its design did not focus on contextual factors that might have influenced the design. To model the relevant context in the tool I first of all recorded those contextual factors that were either explicitly mentioned in the paper or that were in some way connected to the scenarios created by the OMS design team. I then added factors that I thought would have been relevant in the design of the OMS. If I felt the factors that I had added to the relevant context were relevant in any way to the original design specifications, I created a link to record the relationship. I did not use factors to suggest new design ideas or features or to alter the original design in any way.

Gould *et al.* (1987) describes other design issues that had to be resolved which involved a number of different contextual factors that played a part in the design of the system, such as the many different languages that would be spoken by the Olympians and their families and the wide range of computer expertise that would likely be found across all users. The design issues that had to be resolved related to training users how to use the system and how training could be provided to those users (Olympian's friends and families) who live vast distances away, potentially at the other side of the world. These design issues were specified using the QOC notation which again encouraged consideration of the design space and other alternative design ideas. In so doing, other contextual factors were identified that may have been relevant to the design of the system. Lastly, screenshots and pictures, as well as descriptions given in the paper, were used as the basis from which to specify the behavioural properties of OMS. Here, the NUF notation was used to specify how the system behaved and was 'held together'. The NUF notation was created during the iteration with the scenarios and the context representations. UAN specifications were not created for this study, since the scenarios presented a description of how the system behaved.

To show an example of how different representations were linked, consider the scenario editor shown in Figure 25. The scenario editor displays the scenario which the developers of the OMS system used to describe the way that an Olympian leaves a message for someone. At the top of the window is a text editor which displays the scenario. At the very bottom of the window is a panel which consists of two text lists (labelled 'All collections' and 'Contents of collection') and three buttons. The 'All collections' text list lists the names of all the parts of the scenario that have been declared as suitable sources or destinations of links. The 'Contents of collection' text list lists the contents of the currently selected part or collection[7].

The list directly below the scenario text field shows the anchors that have been created for this scenario. The user creates an anchor by dragging through the text they want to link to or from, chooses the 'Create anchor' menu option[8] and supplies a name for the anchor which then appears in the anchor list. The user can then select this anchor and add it to the currently selected collection by selecting the 'Add object' button on the panel at the bottom of the window.

[7] Unfortunately due to implementation constraints, the contents of a collection cannot be simple text (which would be the ideal solution given that links would be created to or from the text of a scenario). Instead, the text inside a scenario that the user wants to link to or from has first to be designated as an anchor.

[8] Accessible via the small menu bar at the top of the scenario text field and indicated by a small 'v'.

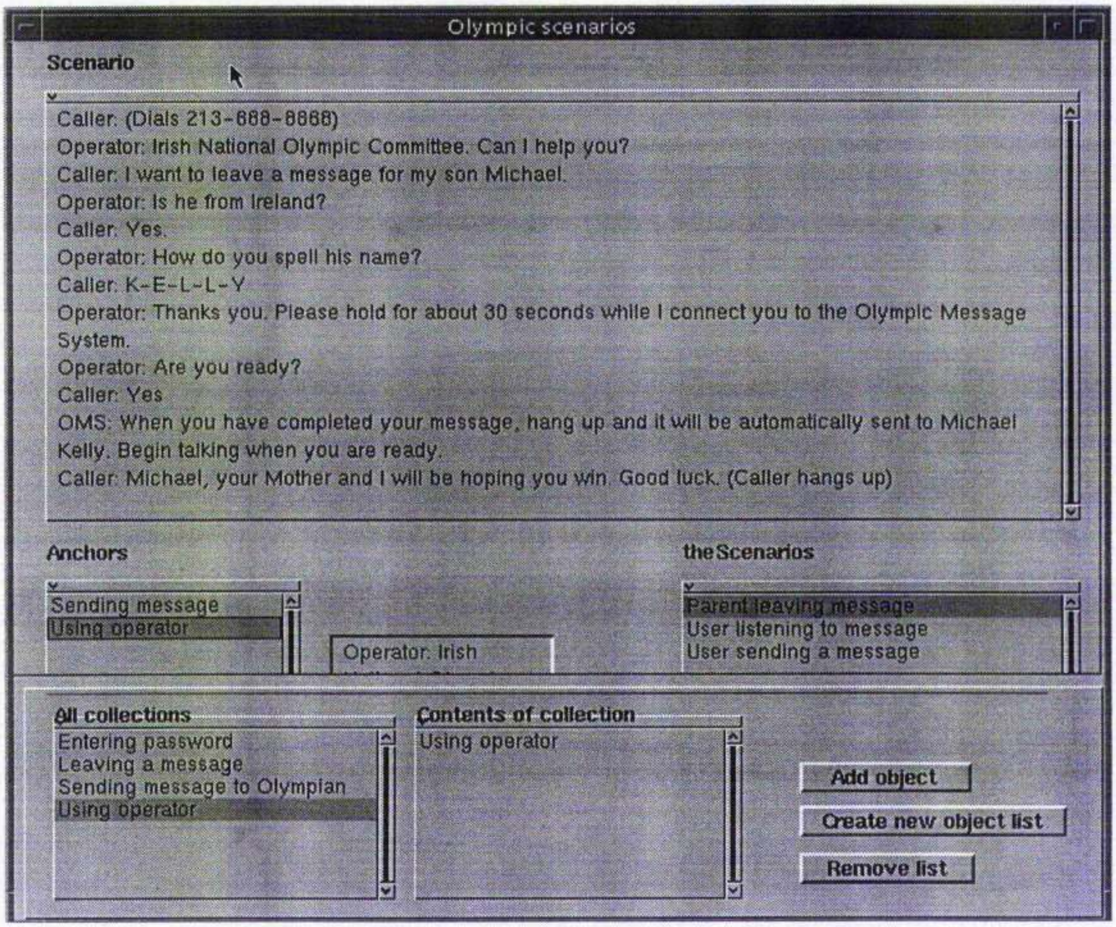


Figure 25. The scenario editor for the OMS system

In the case of the scenario shown in Figure 25, the user has selected the 'Using operator' anchor and has added it to the 'Using operator' collection. The user could add more anchors to this collection if they wish. This scenario shows how a parent uses the operator of the Olympic Messaging System to leave a message for their child.

If the user wishes to link this part of the 'Parent leaving a message' scenario to the NUF command that the operator uses to add the message to the Olympian's list of unread messages, they must first select the appropriate command in the NUF editor and add it to a collection. In Figure 26, the user has selected the Receive <Message> for <Olympian> command of the Olympic Session object. The three text lists at the top of the window show (from left to right) the objects defined in the NUF, the available 'slots' for each object and the defined commands, attributes etc depending on the slot selected. The text editor below the three text lists displays the text of the currently selected command, attribute etc. Having selected the Receive <Message> for <Olympian> command, the user then adds this to the 'Receiving a message' collection at the bottom of the window.

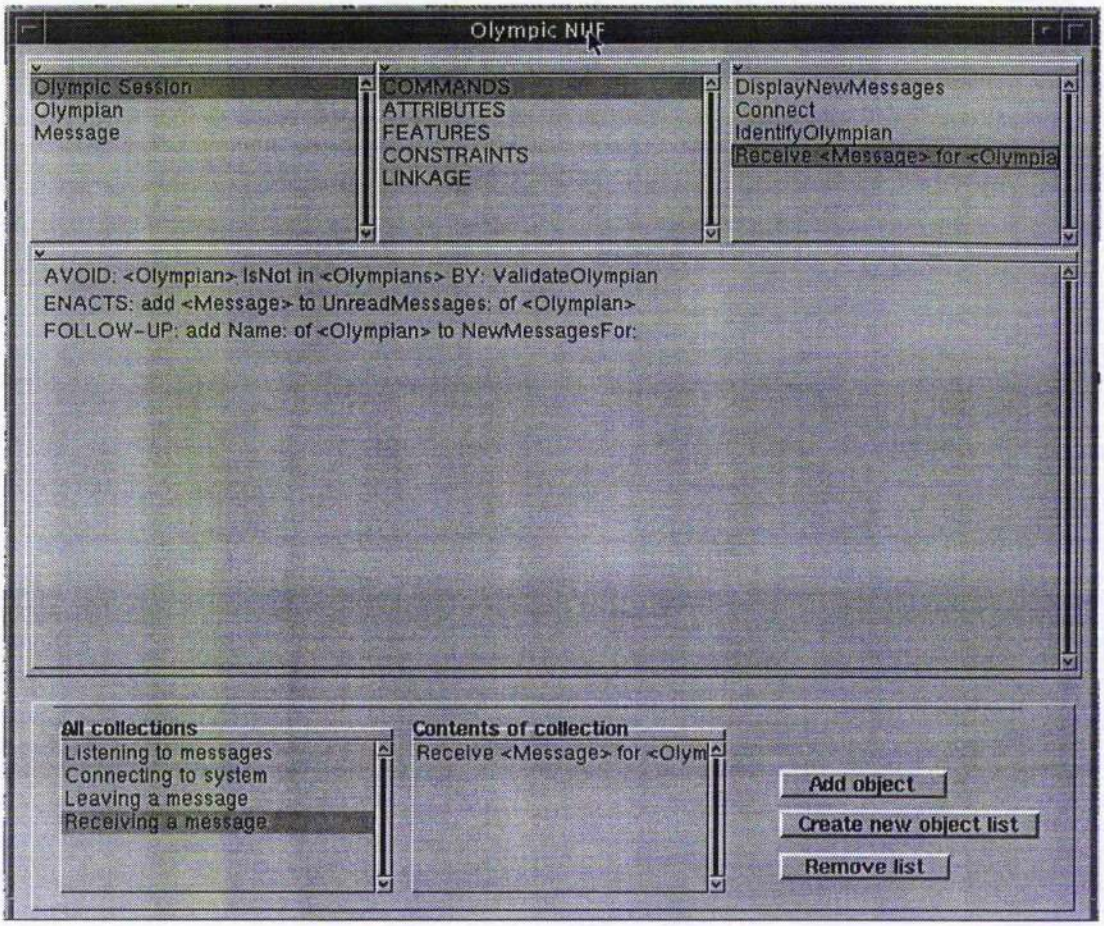


Figure 26. The NUF specification of the command used by the operator to leave a message for an Olympian

To create a link from the scenario to the NUF, relating the description of the operator leaving a message for an Olympian to the abstract command used in the NUF to perform the command, the user must use the linking tool (see Figure 27). The linking tool shows two lists at the top of the window. These list all the editors that the user has created. The list on the left is used to identify the source of links, while the list on the right is used to identify the destination of links. The user must select the Olympic scenarios in the left hand list and the Olympic NUF in the right hand list. They then create the link by choosing the 'Create link' option from the Create menu. After the user specifies a name and some explanatory text for the link, it is added to the list of links at the bottom of the linking tool. In Figure 27, the newly created link is selected at the bottom of the window.

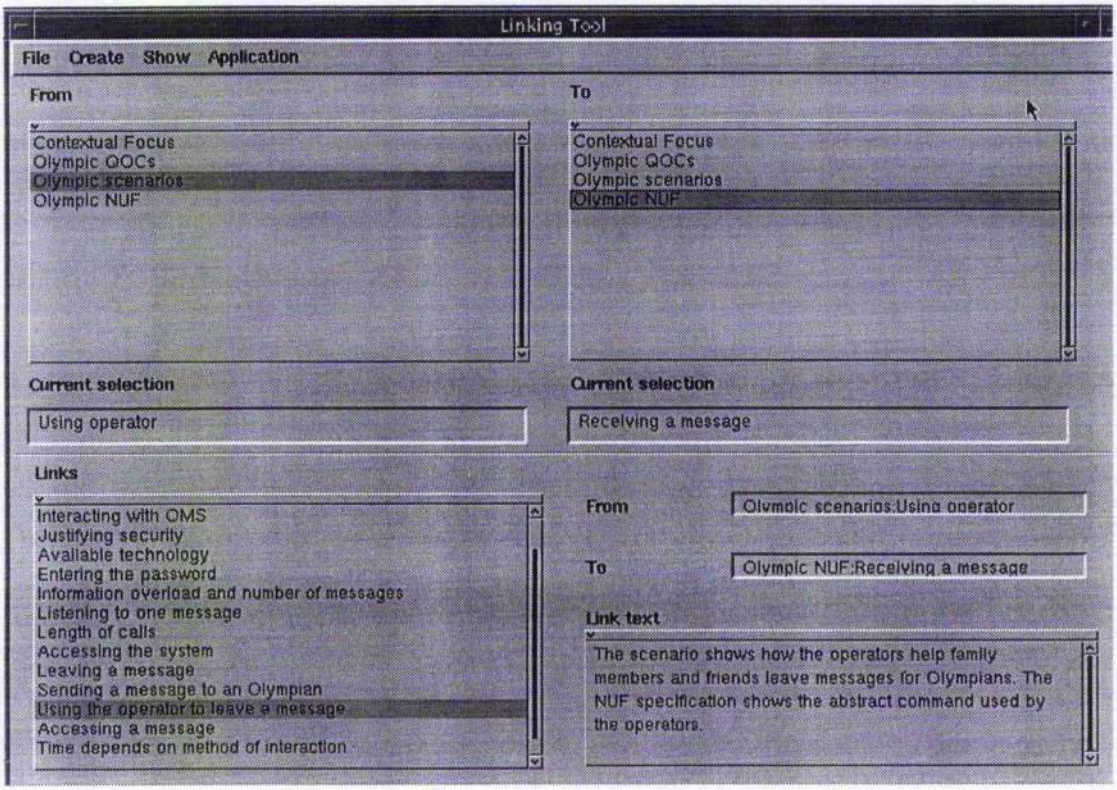


Figure 27. The linking tool

A similar procedure was carried out to create all the specifications and the links between specifications.

Once the specifications were as complete as possible, analysis of the specifications and the relationships between specifications was performed. The new features that were added to the tool as a result of the previous study were used to identify any unused contextual information. Many contextual factors had not been made use of in the design and one scenario had not referred to any contextual factors at all. Of the contextual factors that had not been used in the design, the most surprising was the factor that described the environment in terms of noise. Olympians would access the OMS from public kiosks, dotted around the Olympic Village in highly visible areas. These areas would tend to be noisy, perhaps at times leading to difficulties in hearing spoken instructions. This was noted in the contextual specification of the OMS in LD, but was not used in the design specification. In other words, no relationship between the noisy environment and the design of the OMS existed. There are many reasons for this, the most obvious one being that due to the restricted amount of space that the authors had to describe the OMS in the paper, they simply decided to focus on other aspects of the design rather that those aspects related to the noisy environment. However, this does not detract from the usefulness of LD, since it made clear that some aspects of context were not used at all, including the factor describing the noisy environment. This kind of information could be useful to designers who may have thought that they had indeed made use of these factors.

4.3.4 Comparing the studies

To conclude this section, a comparison is made of the location and number of links created in each study (see Appendix E for a list of all the links created in each of the studies). Such comparisons can be made using 'web diagrams' which show the location and number of links diagramatically. In each web diagram, arrows between representations indicate the presence of a link between the two representations. The arrowhead indicates the direction of the link.

In the IT study (see Figure 28), there are 23 links in total. The NUF, which specifies the abstract design of the whole system, is involved in 13 of those links or 54% of all links created. In the LD study (see Figure 29) the NUF is only involved in 7 links out of a total of 19, or approximately 36% of all links. In both studies, the diagrams illustrate that there are no direct links between the focus and the NUF. All *paths* from the focus to the NUF are either through the QOC or scenario editors. Interestingly, in the IT study, 27% of the links are between the scenarios and the NUF while in the literate development study, the figure is only 10%. This is best explained by the type of scenarios that were used in both the IT and LD studies. In the IT study, concrete scenarios were used which described the objects and actions used to access the IT system. Hence it was much easier to identify the objects and actions that should be implemented in the NUF much in the same way as Rosson and Carroll (1995) suggest. In the LD study, the scenarios were used more abstractly, to suggest some of the design issues that may need to be resolved. This is evidenced by the two links between the QOCs and the Scenarios (one in each direction). As a result of the more abstract scenarios being used to indicate some of the design issues, Figure 29 shows that the LD study involved more links between the QOCs and the NUF specification. Given the differences between the scenarios (i.e., the scenarios are used to identify objects and actions in the IT study and are used to identify some design issues in the LD study), the webs can tell us whether or not more investigation is required. The number of links between the scenarios and the QOCs in Figure 29 could indicate that more work is required here, to investigate if the scenarios could be used to indicate more design issues. However, it would not be fair to say the same for the IT study web in Figure 28, even though there are no links between the QOCs and the scenarios. In this case, the scenarios have been used for a different purpose.

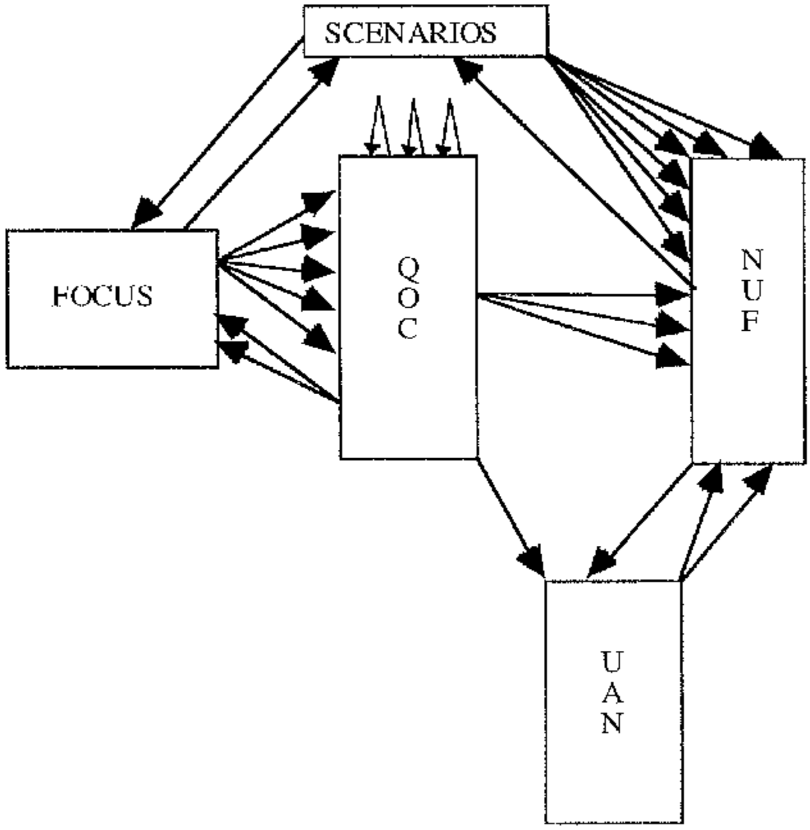


Figure 28. Literate development web from the IT study

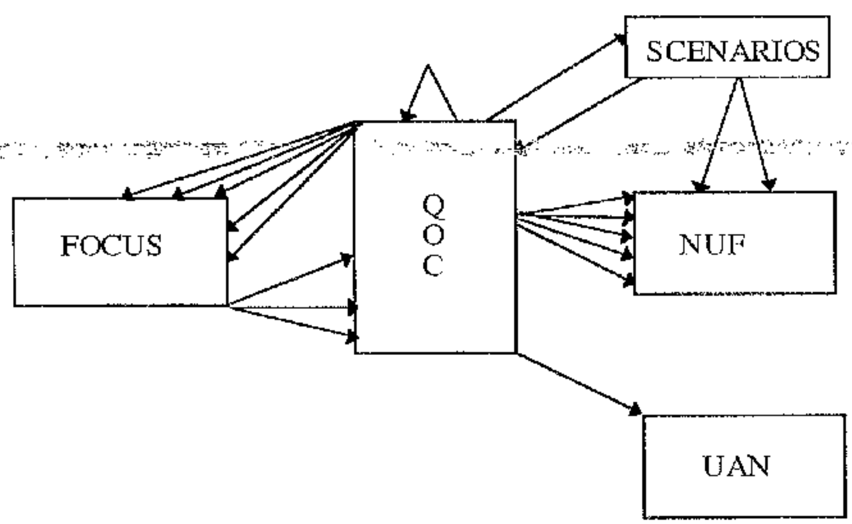


Figure 29. Web from LD study

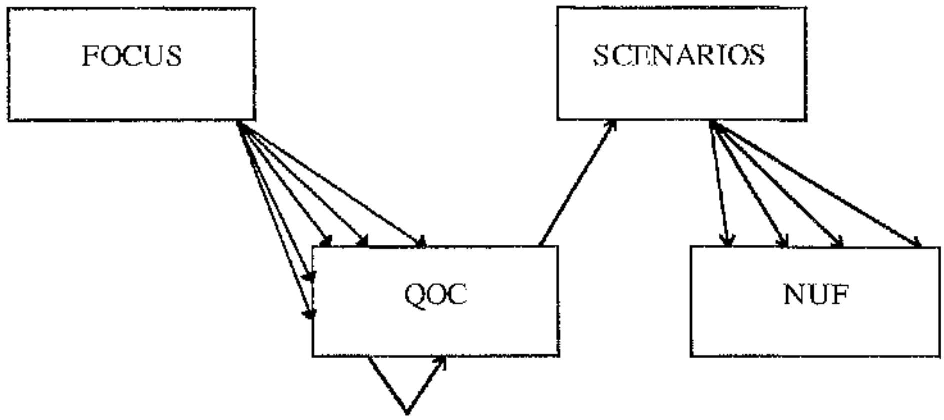


Figure 30. Web from the OMS study

A slightly different picture is seen in the web for the OMS study (see Figure 30). Here, paths go completely one way from the contextual focus, through the QOCs and Scenario descriptions to the final NUF specification which describes the OMS system. Note also the way in which the focus and scenarios have been used. The contextual focus has been used to identify issues which had to be resolved while the scenarios have been used to identify how the system will work. For example, the contextual focus described the differing backgrounds of potential users in that they come from all over the world, may never have used a phone before let alone a push button phone and each individual will speak one or more of many different languages. This kind of contextual information was used to back up the identification of training of users as a major design problem that had to be resolved in the design of the OMS. This mirrors the way in which the OMS team worked and also raises an interesting usage of scenarios. Scenarios can be used in many different ways, to represent designs or to represent the context of a system. Different types of scenarios result in different types of links. For example, in the OMS study the links from the scenarios are used to suggest that the design must be specified as the scenario describes. In the LD study, on the other hand, links from the scenarios are used as sources of design questions to be answered in a QOC.

The next section summarises the experiences gained and lessons learned from performing each case study. Many of the experiences were common across each study, so no attempt has been made to separate descriptions of the experiences with individual studies.

4.4 Experiences and lessons learned

The three case studies provide an excellent example of the flexibility of LD and the success with which it can be put to use in different projects with different purposes. Each study relied on different mechanisms to describe the design and/or the context. In the first case study, the contextual checklist was used to describe the context of use of the IT system. In the second study, the creation of the specifications was driven mainly from the design problems that were encountered while building the tool, by specifying the design rationale for the design. In the final study, scenarios describing the main tasks that would be performed by the OMS were used as the basis from which to drive the description of the context and the design. Hence these examples show that LD does not force designers to specify a system in any particular style or by following any particular method.

The three different studies also show that the tool can be put to succesful use in projects that have a range of purposes. In the first case study, attempts were made to replicate the actual design of the IT system, showing that LD can be used from the beginning of a project. In the other two studies, the tool was used in a reverse engineering capacity, to model the design and context of a system that was designed by myself and, in the last case study, to model the design and context of a system that was designed elsewhere and of which there is limited access to the background of the system. These three examples show that the tool can be used to support a design project as it proceeds, to structure and collect design ideas relating to a particular design and to model and investigate the design of an existing system of which little is known.

Thus LD helps in making effective use of contextual information in that it can be used in different projects for different purposes. Since context can be used in different ways and at different times throughout the lifecycle of a project, support tools, to be most effective, should make a contribution not just at one stage of the cycle but wherever required. These three studies have shown that LD can be used at different stages, from project initiation through to completion and after completion, when a system has to be reverse engineered in order to maintain it.

The case studies have also shown how the tool can help in inspecting the use of contextual information in design. Such inspections can be used as the basis from which to identify further areas of work (such as the need to collect more contextual information, the need to base more of the design on the context or the need to reconcile aspects of context and design with one another). In the IT case studies, inconsistencies and omissions were identified between specifications by identifying and closely examining appropriate relationships between specifications. For example, one inconsistency that was described above concerned the different names in the UAN and NUF specifications of the IT system for the command that starts the IT system up (see Figure 22 and Figure 23). One possible explanation is that the tool's support of maintenance of relationships between different specifications, encourages finding further relationships since users are spared the cost of maintaining them, thereby making the time spent identifying appropriate relationships worthwhile. Hence through encouraging users to identify possible relationships between specifications, the tool implicitly encourages detailed examination of the specifications which can identify mistakes. In the OMS study, the 'Show unused context' feature of LD (which was added after the second study in which the tool was used to describe itself and so was not available in the first two studies) identified a number of contextual factors and one scenario as having been completely unused in the design. Both these examples provide further evidence that LD can increase the efficiency with which designers make use of contextual information in design. By identifying and making designers aware of unused context, the tool reduces the amount of contextual information that is wasted. And by encouraging careful examination of the different specifications, the tool helps designers identify more ways in which contextual information can be used to influence designs.

4.5 Weaknesses of the Literate Development Tool

This section discusses a number of weaknesses that were identified through carrying out the three studies described above.

There were some difficulties experienced while using the tool, particularly when creating or editing a link, which LD uses to represent a particular relationship between specifications. For example, in each case study, it was difficult to tell the desired effect or meaning of any particular link. The only access to this kind of information was via the link text. In some cases, the link text was insufficient to enable a good judgement to be made with regards to the meaning of the link. Figure 31 shows an example of a link that does not clearly communicate if the relationship between the scenario and the NUF is one that must always exist or if the relationship can be changed. If the relationship was one that must always exist, then presumably if the scenario changes then the NUF would have to change as well so that the NUF reflects what the scenario describes. However, if the relationship is 'optional' then if one end of the relationship is changed, the other end need not be changed (see Figure 32 for an example of a link that could be considered optional). The relationship could be deleted or altered in some way. Relationships of the type shown in Figure 31 led to some uncertainty as to what should happen after a change has been made. Each time a link was displayed after a change had been made it was difficult to know what to do with the link (edit the source/destination, reject the change, edit the link, remove the link). In the next section, one such solution to these problems is discussed in which links are given a type and the type is used to determine what should happen as a result of a change.

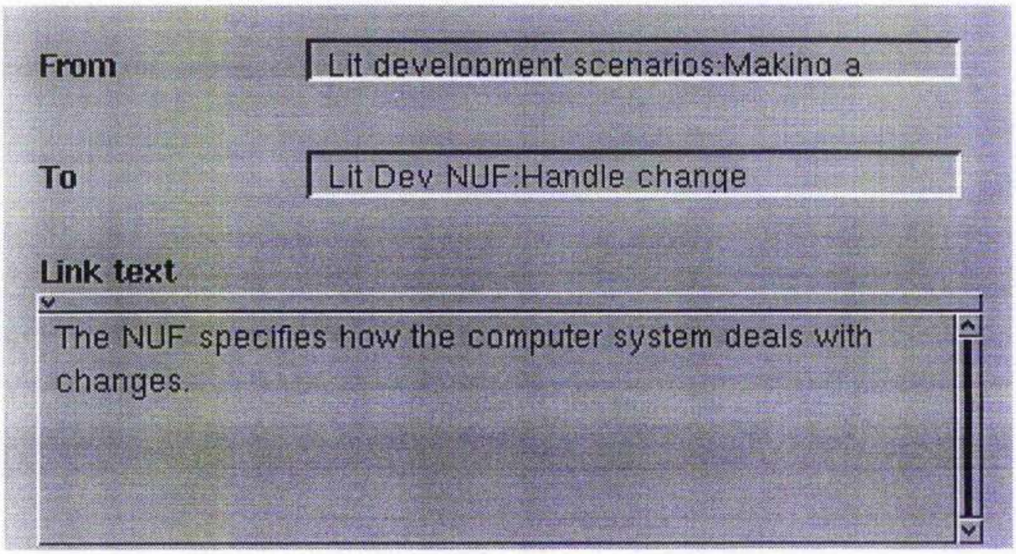


Figure 31 Unclear link

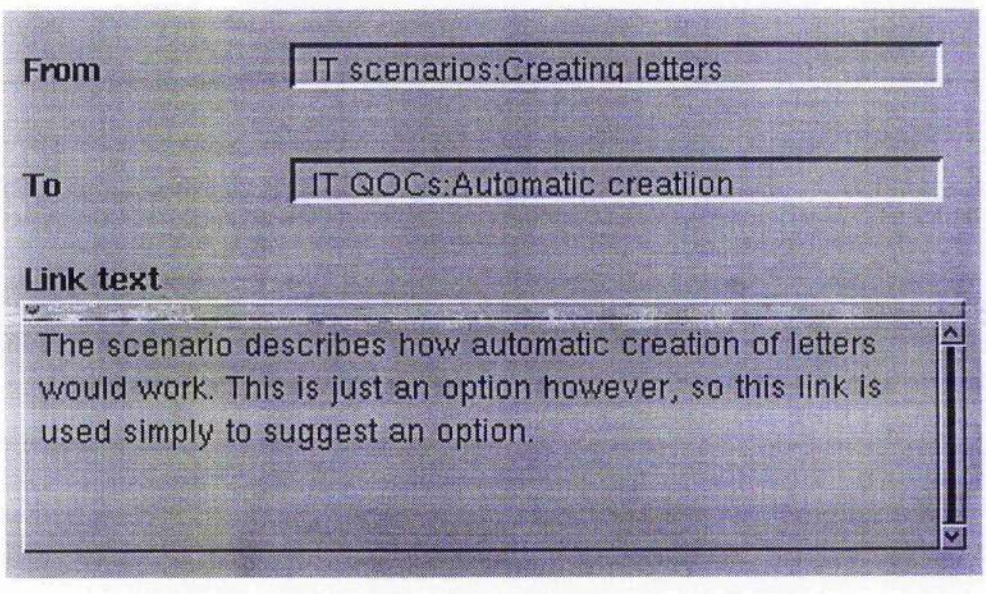


Figure 32 Clear link.

Difficulties in describing the meaning or type of a link also hindered evaluations of whether or not duplicate links were being created. It was difficult to know if similar links had been created when creating new links. While the link text can be used to judge similarity, this method depends on good descriptions being created. If the amount of influence that a piece of context has had on a design is measured by the number of links from that piece of context to the design then duplicate links make it look as if some part of the context has influenced a part of the design more heavily than it actually has. This could perhaps lead to designers making false conclusions about the way that the contextual information has been used in design, leading them to believe that the design has been sufficiently influenced by the context. They also make it easier to introduce inconsistencies in the specifications since one duplicate link may be updated appropriately during the project, but the other may not.

In the IT case study, I was uncertain about whether or not all of the contextual information should be recorded using the contextual focus editor, or whether this could be shared between the focus editor and the scenario editor. This was largely due to the fact that scenarios had been written to cover all of the tasks to be supported by the system *and* the context of use of the system. This meant that most items entered in the contextual focus would also be found somewhere in the scenarios. In LD study there was no such uncertainty, since the scenarios that were created only covered a subset of the contextual information. I felt that the contextual information was naturally recorded using both the focus and scenario editors. In this case, information that was recorded using the focus editor was not mentioned in the scenarios. Similar effects were found in the OMS study, since there were only a subset of all the scenarios used by the original OMS design team available for public inspection. Therefore, there was less chance of duplication of information since again, most things represented in the focus editor would not have been duplicated in the scenario editor.

Sometimes, while developing the specifications and the different relationships between them, it would have been useful to be able to link, not just to and from context and design ideas, but also from a link to another link, or from a link to the context or design idea. Links themselves sometimes depend on contextual information and may need to be changed as a result of a change in the context. Some piece of contextual information may have influenced a relationship between some other piece of contextual information and the design. For example, the link text of one link in the LD study contained the phrase 'to the best of my knowledge' (see Figure 33). It would have been useful to link from this phrase to some representation of 'the best of my knowledge' whether in the contextual focus or scenario editor such that when the information about current knowledge was updated then the link would also be updated. This link related the available technology to the decision about whether or not to support a multi or single user system. Crucially this judgement depended on my knowledge of the facilities offered by VisualWorks[9]. I was not entirely sure about the level of support offered by VisualWorks for multi–user applications. If at a subsequent date I discovered something else about VisualWorks which would suggest that it does support multi–user applications then it would have been useful to be able to see all the links that depended on my knowledge of VisualWorks. I would argue that this situation is similar to one that most designers would find themselves in. Designers often have to deal with indefinite and incomplete information. This should not stop them making decisions or creating relationships between representations. It would be useful if a tool were able to indicate all the decisions that a design team made on some uncertain or incomplete knowledge such that when that knowledge changes, they have access to the decisions that relied on that knowledge.

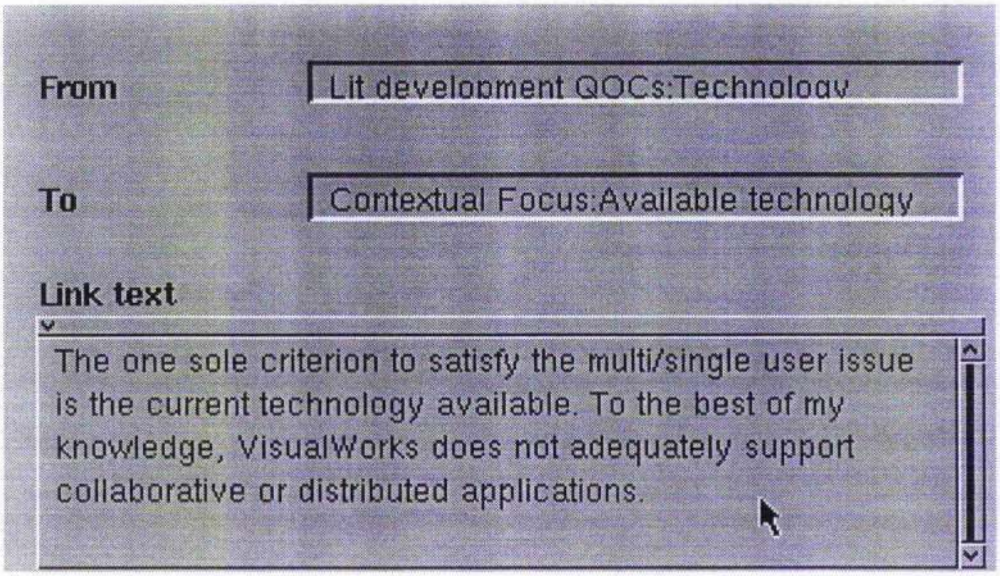


Figure 33. This link depends on knowledge about the support environment.

4.6 The implications for the future design and development of LD tools

The problems with discerning the meaning of a link and thereby deciding upon the appropriate action to take after a change suggests that links should be assigned some type. The type of a link would help suggest what should be done as the result of a change. Ideally, each link could be classified as being one of a certain number of types. This would

---

[9] VisualWorks is an application development toolkit for Smalltalk. I started to use it during the development of LD and so was learning about it as LD was developed.

allow consistent interpretations of the links to be made. The link text would then be used as a means to provide more detail as to the meaning of the link. There are further benefits to this approach. Designers could opt to see only those links that matched a certain type. Designers would be able to see if a particular point in the specifications has a particular effect elsewhere (i.e., by opting to view only a particular type of link from a particular point in a specification).

By performing the case studies a number of possible links have been identified. Links can *support* or *contradict* design options/specifications etc. For example, in the OMS study, knowledge about the expertise of users (low expertise) and their experience with telephones supports the option to provide some form of training. At the same time it contradicts the option to support complex functionality, such as setting up conference calls.

The question of defining suitable types such that the type itself determines what action should occur as a result of a change to the source or destination of a typed link is harder to answer. In many ways, the action to take following a change to the source or destination of a link depends on the change made and the particular objects that are involved in the link. For example, in the OMS study, if the knowledge of users changes such that they are now all expert users with the phone system, then this no longer supports the option to provide some form of training. However, if the knowledge of users changes such that they now have low to medium levels of expertise, then the link still supports the option to provide some form of training to users. There is however similar work which has been able to identify typed links which do suggest an appropriate action to take. Kaiya *et al.* (1995) discuss the design of a hyper–media tool to support requirements elicitation meetings. They have developed a tool which allows discussions to be recorded. The tool also allows the influence that the discussions have on other discussions to be recorded. They define a relationship 'influence' for describing the dependency between topics of discussion. If the conclusion of the topic of some meeting is changed, the influence relation is used to identify all those other meetings and discussions that were influenced by this discussion and that, according to the authors, should now have their conclusions updated also. However it remains to be seen just how strictly this can be adhered to. As in the OMS example above, a topic could be changed such that other changes in discussions that it influenced would not have to change (i.e., the wording of the topic may change but its meaning could remain the same).

In the realm of software engineering, Morrison *et al.* (1995) describe three different relationships between components of a software engineering environment. These are *causations*, *associations* and *links*. *Causations* are cause and effect relations. These are relations between two components A and B, where a change in A causes an indirect change in B. The authors use the example of a source program and its compiled version. After a change in the source program, the compiled program changes, but only after the source program has been compiled. *Associations* are more general relationships which rely on adherence to external conventions. The examples given above from the OMS system in which the knowledge levels of users is related to the option of providing training is an association since it depends on external conventions in order to retain its validity. The relationship relies on the creators of the link ensuring that the link remains valid after a change to the source or destination. With association type relationships, automated updates cannot be provided since the update relies on external conventions. Lastly, *links* between two different components exist if a change in one component causes a change in the corresponding component without the need for any intermediate process.

Most, if not all, of the relationships defined in the three tool studies described in this chapter would appear to be associations. Therefore automated change support is most likely not possible for these types of links. Further work needs to be done to investigate what kinds of relationships could be classified as causations or links, in the terminology used by Morrison *et al.*

The two types (*support*, and *contradict*) could be used whether or not the relationships are associations. However, designers may not always be able to define a suitable type for a link. In this case, the link text could play an important role as it could be used to help identify the nature or type of the link. However difficulties in describing the meaning of a link through the free form text were described above. Hence, in keeping with the flexible nature of the tool, users of LD should not be enforced to provide a type for a link. Furthermore, instead of providing a simple set of pre-defined types for the relationships between context and design, it may be worthwhile to allow users to define their own types for particular projects. Designers could agree on a suitable typing convention for the relationships between context and design that they define. This is similar to the way that most programming languages allow programmers to define their own types over the predefined base types. Programmers can agree on a type that makes sense to their own program. For example, a group of programmers working on a program that deals with classes of students and their marks may define a type called *ClassMarks* that represents the marks of the class. In so doing, the programmers can write their program so that it makes more sense with respect to the functions that it is supposed to perform. The user defined types help explain what the different variables of that type represent.

With regard to encouraging the creation of links in LD, the effort required in creating anchor points for the source or destination of a link (conscious consideration, physical manipulation) meant that they would not be created unless they could potentially be used in a link. This could imply that few links would be created, since the effort involved in creating anchor points would deter designers from creating them. This suggests that either the method of creating links should be changed to make it easier to create anchor points or that the design method itself should work in such a way that it motivates users of the tool to spend the effort in creating anchors. However, a balance has to be struck being making it too easy to create anchors and too difficult. If it is too easy to create links, designers may perhaps just be inclined to create links all over the place, not paying too much attention to whether or not the link is valid since it is easy to add links. On the other hand, if it is too hard to create links then links will not be created and important relationships will not be recorded leading to difficulties and inaccuracies in assessing the way that the context has been used in the design.

The tool provided further evidence in support of the main arguments of this thesis, that the relationships between context and design should be made explicit. There were no direct relationships made between the focus and the design. Instead, such relationships or links were indirect, in the form of paths through the QOC or scenarios. However, the indirect links between context and design are due to the fact that the contextual information was never used to immediately influence the design. It was used to suggest design issues or to influence scenarios. Contextual information can be used in many ways throughout a design project and influencing design ideas is one such valid use of contextual information in design. It is clear that when contextual information is used in this way, most of the links between context and design will be indirect and a linking tool would be of benefit. Without representation of the paths between context and design, particularly when the paths are indirect as is the case in the three studies described here, it would be difficult to determine exactly what influence the focus has on the design. With the tool, a path (or paths) can be followed from an element of the contextual focus to its eventual destination. Each 'node' visited, is a node which will be influenced by the source of the path, the element of context in the contextual focus.

The tool could also be used to suggest areas for further investigation with regard to the creation of links. In all webs, there are virtually no links from the NUF (the design) to the context (the scenarios and the focus). In the IT study there is only one such link, from the specification of applicant status to the scenario describing the use of a waiting list. The lack of links in this direction is highlighted by the tool and suggests that further investigation may be necessary. Again, if these links were not made explicit, the need for further investigation may not have been so obvious.

4.7 Chapter Summary

This chapter has described three different case studies in which LD was used to specify the context and design specifications and the relationships between the specifications for three different systems. Each case study represented a different use of LD and provided evidence of the benefits of the tool and its flexibility. The case studies demonstrated the following benefits:

- LD encourages greater consistency between representations. In the IT study, differences between design specifications created on paper were identified when the same representations were created using the tool;
- LD can be used to reverse engineer current systems. In its support of the QOC notation it can be used to improve existing designs by defining alternative design ideas with reference to the contextual information. By doing so, greater understanding of the contextual information is encouraged as in the example from the IT study related to printing letters in batches or individually;
- LD can be used to identify elements of context that have not been used to influence a design. In the OMS study, LD identified a number of items of relevant contextual information that were not used to influence the design, such as noise levels in the areas around which the OMS would be used.

Each of these benefits can be exploited to make more effective use of contextual information. By encouraging consistency between representations, the tool encourages the consistent use of contextual information, since context will most likely be used consistently in consistent representations. By supporting and encouraging the specification of alternative design ideas, greater understanding of the context is encouraged. By identifying unused contextual information, the tool supports designers in making judgements about whether the unused context is irrelevant or not and should be used to influence the design.

A number of problems were identified with the tool also and these were described. However, the three case studies described represent an informal evaluation of LD in an academic context. The next chapter describes an attempt at a more rigorous evaluation of LD.

5. Chapter 5 Experimental Validations

5.1 Introduction

In Chapter 4, three case studies were described in which LD was used to specify three different designs and their associated context, together with the relationships between the design and the context. These studies demonstrated that LD provided some benefits when making efficient use of contextual information in design. For example, the case studies demonstrated that making the relationships between context and design explicit can identify ways in which contextual information has not been used in a design. The case studies however did not evaluate LD systematically. This chapter describes a number of attempts that were made to evaluate LD in a more systematic and rigorous fashion.

Rigorous evaluation of the tool, whereby the hypothesis that linking between representations of context and design encourages more efficient use of context is tested by controlling and manipulating different experimental variables was difficult. In order to gain maximum benefit from the tool, users require a minimum amount of knowledge regarding the design project and corresponding design specifications specified in LD. Without such knowledge, users are unable to utilise the information provided by LD since they need to relate this information to the design project to make effective use of it. Other issues that make rigorous evaluations of LD difficult are the skills required by users to use the tool. For example, users need to know about and understand the different notations that are used in LD to represent both contextual information and the design. Given the time available to perform these evaluations, it would have been impossible to recruit the required number of participants, train them in the design and context notations used in LD and provide them with background knowledge relating to the design project that LD is used in and then to perform the evaluations. To address these difficulties, the problem of performing one complex rigorous evaluation of LD is broken down into three different, less complex evaluations.

The first such evaluation is a formative evaluation of the tool and was originally a pilot study for an assessment of the claims made for the tool. Many of the difficulties of performing a comprehensive evaluation of LD were experienced during the pilot study. The pilot study encouraged consideration of other ways in which to evaluate LD. It was clear from the study that such a large complex evaluation would be impractical. However, the findings from the study were useful and were used to provide a formative evaluation of LD.

Many difficulties that were experienced in the first evaluation related to difficulties in understanding both how the tool worked and the functionality supported by the tool. The second evaluation attempted to address these difficulties by using paper based representations of context and design and using these paper based representations to evaluate the benefits of making the relationships between context and design explicit. By removing the tool from the evaluation, usability problems were eliminated. Other difficulties concerned the design and context representations used in LD. These were simplified in the paper based representation by describing the context and design using plain English prose, Hence, difficulties that participants experienced in understanding the different notations used in LD were removed. Lastly, the design project for which LD was being used was simplified by reducing the amount of design and context description, such that the time taken to understand the design project was reduced. By reducing the complexity of the design project, the representations involved and the mechanisms by which these representations are accessed, the amount of time required to perform the evaluation was reduced. Therefore it was easier to recruit a sufficient number of participants to perform the evaluation.

However, one of the main reasons for the development of LD was to support the maintenance of relationships between context and design for authentic development projects. As the original IT case study demonstrated, this is a difficult task to achieve succesfully using paper based tools alone due to the number of relationships between context and design and the large size of the design and context representations. Hence, in the paper based evaluation, the representations involved are necessarily much simpler and smaller than those created during a real development project, so that participants in the evaluation can understand and use the representations during the time allotted in the evaluation. In order to evaluate the benefits of explicit relationships between context and design in an authentic development project, an evaluation involving the tool is required. Therefore the third evaluation makes use of LD, in an informal, subjective evaluation, taking problems experienced in the first evaluation into account. In this evaluation, participants are presented with descriptions of a design and associated context and are asked to use the tool to review the influence that context has had on the design and then to use this review to suggest and make one or two design changes. This is an informal, subjective evaluation of the tool since only a small number of participants take part in the evaluation and data collected relates to their experiences using the tool, rather than other more objective data such as errors, time taken to perform certain tasks etc.

This chapter begins by describing the original pilot studies, which informed a formative evaluation of LD. The difficulties in performing an evaluation of LD are then described. These difficulties were made clear as a result of performing the original pilot studies. The formative evaluation suggested a number of modifications should be made to LD and these are described. The paper based evaluation is then described followed by the informal evaluation of the tool. A summary of the three evaluations and their implications concludes the chapter.

5.2 A formative evaluation of LD

LD was designed and implemented to address a number of problems with using contextual information effectively that were identified in Chapter 3. During the IT case study, attempts were made to relate contextual information with associated representations of a design but a number of problems were encountered. These problems were described in detail in Chapter 3. The problems were used as the basis for the design and development of LD that supports the creation and maintenance of a set of relationships between context and design. Hence a number of claims can be made for LD that relate to the problems identified in Chapter 3, Section 3.3. These claims are listed below.

- LD tools let users accurately identify potentially irrelevant pieces of contextual data;
- LD tools help users in making judgements about how consistently contextual information has been used in design;
- LD tools help users in making judgements about how appropriately contextual information has been used in design;
- LD tools help users in identifying the affected objects of a change;
- LD tools help users to make relative judgements about the amount of context used in design;
- LD tools help users make relative judgements about the influence of contextual information over the design.

These claims were informally evaluated in the Chapter 4 case studies (LD was used to describe three different designs and associated contextual information). This section describes an attempt that was made to formally validate these claims.

The claims were used as the basis from which testable hypotheses could be derived and tested in a formal validation of the tool. Such an experiment was designed and two pilot studies were performed.

The original experiment involved the participants performing simple tasks with LD that corresponded to each hypothesis. Each hypothesis was described such that it related to the tool alone. For example, the following example is the hypothesis that was derived from the the first claim described above[10].

H1: Users of LD tools can identify pieces of context that have influenced the design.

Notice that, while the first claim made for LD suggests that the tool can be used to help designers identify irrelevant context, the hypothesis does not suggest that the tool alone can be used to identify irrelevant context. What the hypothesis makes clear is that the tool alone only identifies those pieces of context that have influenced the design (and conversely, those that have not). The tool does not identify the irrelevant context. Only the designer can make such an assessment using their knowledge of the design project. However, these assessments can be made more accurate by the use of the tool since it identifies those pieces of context that have been used to influence a design.

Similar hypotheses were developed for the remaining claims and were used to develop a number of simple tasks for participants to complete. Participants would be given one of two sets of representations with which to carry out the tasks. Participants in the first group would be given a set of context and design representations and a set of relationships between these representations. In the second group, participants would be given the identical representations. They would not however have access to the set of relationships between the context and design representations. Hence any differences between the answers returned by both groups of participants could be attributed to the presence of the set of relationships between context and design, given that participants matched on all other variables such as design experience etc. In line with the hypotheses, the tasks were very simple. For example, to test the first hypothesis, partticipants were asked to perform the following task.

Make a list of contextual factors that are not related to the design in any way.

For each task, participants were asked to make similar lists, or to count different items of context etc. The answers given by the participants would then be checked against a set of ideal answers compiled by myself.

5.3 Experimental difficulties

After performing an initial pilot study, it was clear that the original experimental evaluation was infeasible. There were procedural problems, such as the amount of time required for participants to learn and understand the tool and the design problem that the tool was being used for. There were other problems as well, such as the novelty of LD and of the concept

[10] All of the hypotheses can be viewed in Appendix B.

of explicitly relating contextual information with design. These problems are described in detail below.

- *Understanding the concepts behind the tool.* The idea of explicit relationships between context and design representations was difficult for participants in the pilot studies to grasp fully. Combined with the complex representations of the relationships, this meant that the participants had difficulty both in recognising a relationship and in interpreting what that relationship meant.
- *Understanding the design that is represented in the tool.* In the sixty minutes that each participant took to perform the pilot study, it was impossible for them to inspect and comprehend every part of the representations given to them and to relate these representations to the 'real world' that they represented. This lack of understanding of the design made it difficult for them to understand the relationships between the different representations. The difficulties which were experienced in understanding the details of the representations translated into difficulties understanding the relationships between different representations, since if the representations could not be understood, it was difficult to understand the relationships between representations.
- *The usability of the tool.* The tool was designed to be used by myself to enable me to investigate the issues related to linking between representations of context and design. Addressing usability issues within the tool was not the main focus of the work. The only criterion that was used to judge the design was the ease with which I could use and understand the tool. However, this meant that I was prepared and able to put up with a number of problems in the tool which the participants in the pilot studies were not able to do.
- *The time allotted to each study.* As has been mentioned above, the time allotted to each study (sixty minutes) was not enough to allow participants to understand the concepts behind the tool (what the different representations meant), the system being designed through the tool and the tool itself.
- *The validity of the tasks that participants were asked to perform* . In order to keep the length of the study short and to evaluate the tool, five different tasks were designed that would evaluate the tool alone. The tasks were very simple and did not require that participants fully understood the different representations in order to complete the tasks. However, the simplicity of the tasks meant that they did not relate to activities that would occur in a real design project. Participants questioned the validity of the tasks and in some cases, thought that the tasks had hidden complexities and searched for more complex solutions to the tasks than the simple solutions that were required.

As a result of the pilot studies, some information was gathered that allowed a formative evaluation of LD. Many of the problems that the participants experienced with the tool related to the inconsistent terminology that was used throughout the tool. In particular the Show menu (see Chapter 4, Figure 4) used two different words for the same concept. To refer to a part of a design or context document, the show menu used the words 'objects' and 'anchors' (i.e. in the menu options 'Duplicate anchors' and 'Affected objects'). The menu also used 'app' as shorthand for application, which was the terminology used to describe context and design documents. This shorthand was not clear to the participants (it was not clear that 'app' stood for application) and furthermore, describing a document as an 'application' did not make much sense to the participants. Instead, participants felt more at ease describing each 'application' as a document.

Throughout LD, the relationships between context and design were referred to as links. Since the experiment used the word relationships to explain how context and design could relate to each other, the participants felt uneasy with the word 'link'. They thought that a link was something different to a relationship and questioned the meaning of a link. The most confusing aspect of the tool however was the notion of creating a link from individual object lists of each document. Participants had difficulty in understanding what an object list was and how it was associated with a document. In explaining the object lists to the participants, it became clear that what made most sense to them was to think of the object list as a part of a document. Then relationships would be created between parts of a document and a document could be involved in many different relationships since a document consists of many different parts.

As a result of this evaluation of the tool, the tool was modified to reflect the suggestions that had been made by the participants. Figure 40 shows the new linking tool, which is now called a relationship editor. The labels for the fields at the bottom of the window have been changed to indicate that the tool is representing the relationships between context and design instead of a set of links.

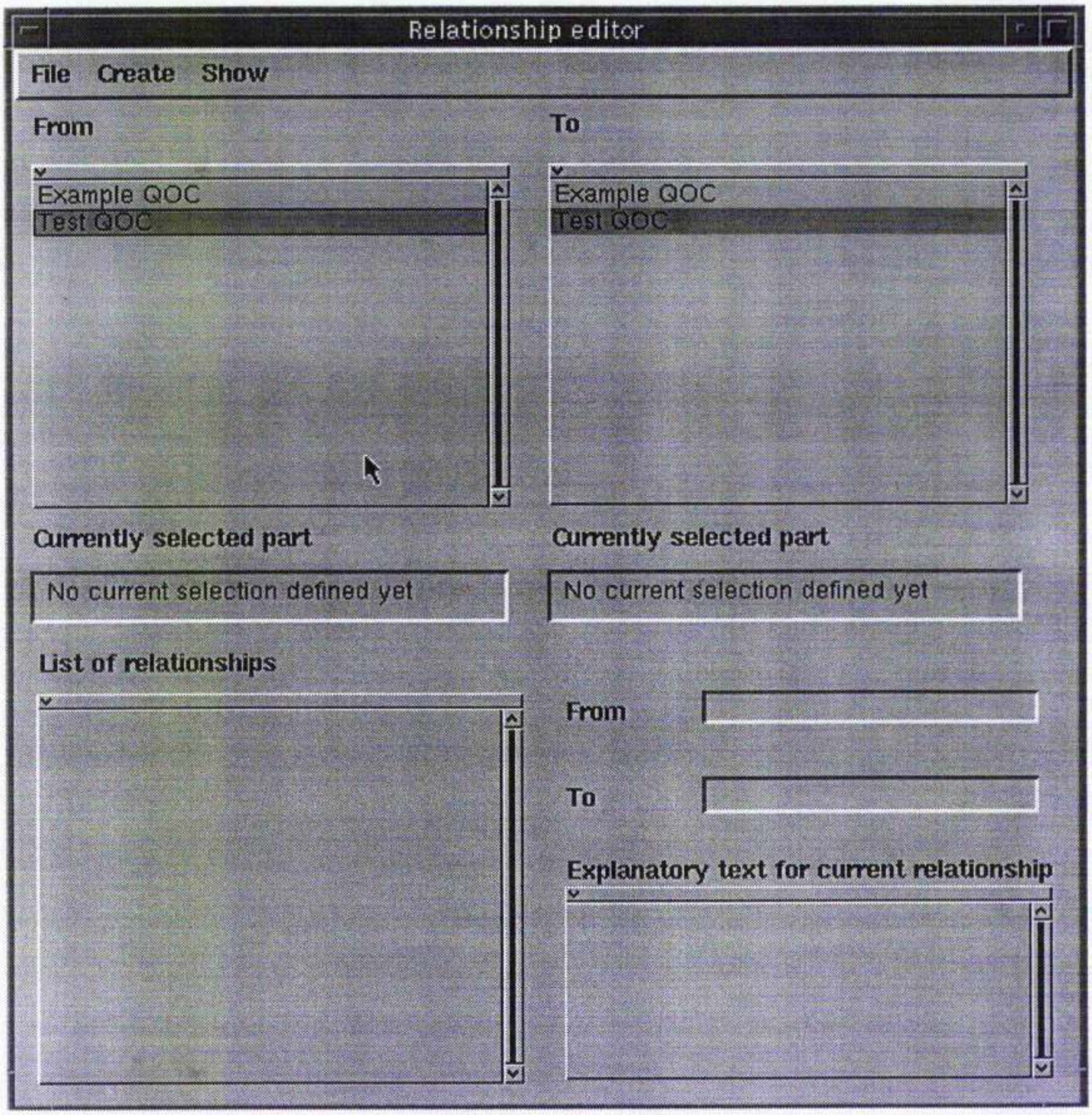


Figure 40. The modified linking tool

Figure 41 shows the new panel which is displayed at the bottom of every document. Instead of representing object lists and the contents of objects lists, the panel represents parts of documents. The buttons have also been changed to reflect these modifications. Notice that this panel is shown as an independent window but in actual usage, this panel would be attached to the bottom of every appropriate document window.

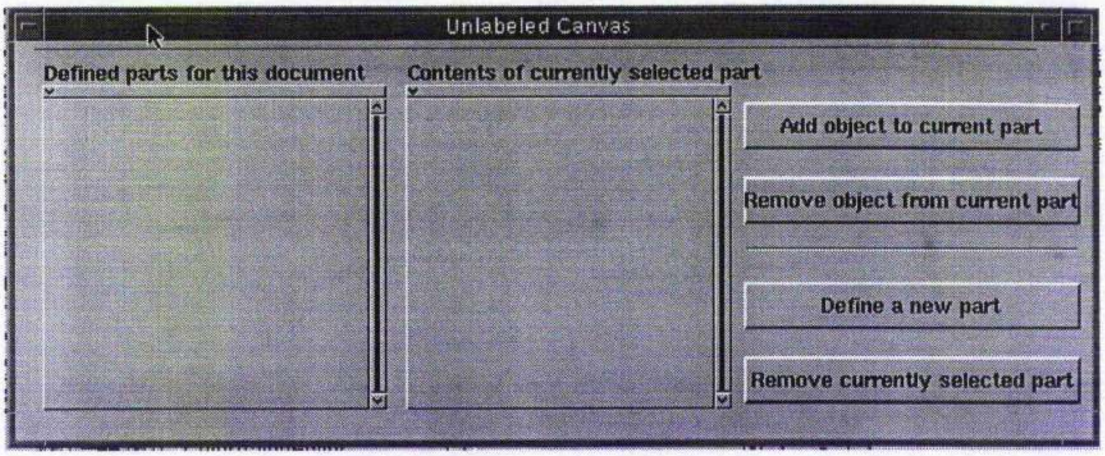


Figure 41. The new panel for defining document parts

5.4 An alternative experimental validation

The difficulties that participants experienced in the pilot study suggested that some alternative experiment should be carried out. The alternative experiment should satisfy the following conditions. It should

- use real world tasks e.g., evaluating the cost of a design change or proposing a design change so that the participants are performing valid tasks;
- contain enough background material such that participants can become familiar with the design situation described in the context and design representations;
- use simple representations that the user can easily manipulate. This suggests that the experiment should not use LD since it is a complex tool and is time consuming and difficult to learn to use;
- evaluate the main claim of LD i.e. that making the relationships between context and design explicit encourages designers to make more efficient use of contextual information.

Such an experiment is described below.

5.5 A paper based validation of the benefits of explicitly representing the relationships between context and design

5.5.1 Hypothesis

The hypothesis being tested in this evaluation is as follows.

> If the relationships between context and design are made explicit, then designers will find it easier to accurately identify unused contextual information and those parts of the design that have not been influenced by contextual information.

5.5.2 Method

Sixteen participants (N = 16) were selected from the research students in the Department of Computing Science. All participants had a minimum level of experience of software design defined as having either obtained a B.Sc in Computing Science or an equivalent qualification. Each participant was randomly assigned to one of two groups of size N/2. Participants in the first group were supplied with descriptions of the context and design of the Olympic Messaging System (Gould *et al.*, 1987; see also Chapter 4, section 4.3.3) together with embedded descriptions of the relationships between context and design. Participants in the second group were supplied with identical descriptions of the context and design of the Olympic Messaging System but these descriptions did not contain the embedded descriptions of the relationships between context and design. Thus the presence of embedded relationships was the independent variable for this evaluation and performance of participants in the evaluation was compared for two levels of the independent variable. Either the embedded relationships were present in the documentation given to participants or not.

To compare the performance of both groups of participants, each participant was asked to study the design and context descriptions they were given and to complete the following tasks

- make a list of contextual factors that have not been used to influence any aspect of the design;
- make a list of parts of the design that have not been influenced by the contextual information;
- judge how easy or difficult it was to create both of these lists;
- estimate how long it took to compile both lists;
- suggest a way in which the design could be improved such that it takes at least one of the unused contextual factors into account
- suggest some aspect of context that is relevant to at least one part of the design that hasn't been influenced by contextual information;
- estimate how much of the contextual information has been used throughout the design;
- estimate how much of the design has been influenced by the contextual information.

The answers returned by members of each group were scored according to certain measures which were validated by HCI experts. Various statistical analyses were used to compare results returned from both groups. A sample copy of the materials supplied to participants in both groups is shown in Appendix C.

5.5.3 Results

Results returned concerning the lists of unused contextual information and design features that were not influenced by context were scored according to the following formulas.

Let $|A|$ = number of elements in set A

Let $A \cap B$ = common elements in sets A and B

Let $| A \cap B |$ = number of common elements in sets A and B

Let $A \setminus B$ = those elements that are in set A but not in set B

Let $| A \setminus B |$ = number of elements that are in set A but not in set B

Then, measures of effectiveness, validity and invalidity can be defined as follows

Effectiveness = $| A \cap C | / | C |$ where A = participants' answers and C = the correct answers

Validity = $| A \cap C | / | A |$ where A = participants' answers and C = the correct answers

Invalidity = $| A \setminus C | / | A |$ where A = participants' answers and C = the correct answers

The tables below summarise the results for effectiveness, validity and invalidity for both sets of results (see Table 10 a, b, c, d). Set A refers to the answers obtained from participants who were given the documentation without the embedded relationships present. Set B refers to the answers obtained from participants who were given the documentation with the embedded relationships present. Measures were calculated for questions 1 and 4. Question 1 asks participants to list those items of contextual information that have not been used to influence the design. Question 4 asks participants to list those design features that have not been influenced by the contextual information. Averages for each measure are calculated and displayed at the bottom of the table. The presence of blank lines in any of the tables indicate that a participant did not answer that particular question.

Set A - Without embedded relationships

Question 1

Effective (%)	*Valid (%)*	*Invalid (%)*
50	64	36
14	50	50
43	75	25
21	50	50
78	43	57
28	66	34
36	100	0
79	73	27
43.625	65.125	34.875

Question 4

Effective (%)	*Valid (%)*	*Invalid (%)*
66	50	50
0	0	100
33	33	67
0	0	100
0	0	100
0	0	100
33	50	50
18.86	19	81

Set B - With embedded relationships

Question 1

Effective (%)	*Valid (%)*	*Invalid (%)*
57	72	28
29	57	43
50	70	30
36	55	45
86	100	0
21	50	50
100	93	7
100	100	0
59.875	74.625	25.375

Question 4

Effective (%)	*Valid (%)*	*Invalid (%)*
67	67	33
0	0	100
33	100	0
100	100	0
33	33	67
0	0	100
100	100	0
100	100	0
54.125	62.5	37.5

Table 10 (a, b, c, d). Measures of answers returned by participants.

On average, answers returned by participants using the documentation containing the embedded relationships are substantially more effective than those returned by participants using the documentation without the presence of the embedded relationships. However, application of the Mann–Whitney test (the distribution of the results is not normal, hence a non–parametric statistical test needs to be employed) on the results indicates that the results are not significant, and that the differences in the average scores could have happened by chance. The same is true for measures of validity and invalidity. While the measures are favourable with respect to set B, they are not significant.

A similar situation exists when the time taken to compile the lists for questions 1 and 4 are compared. While the average time taken by participants to compile the lists using the documentation containing the embedded relationships is less than the average time taken to compile the lists using the documentation that does not contain the embedded relationships, these results are not significant. See Table 11 for the results.

Set A - Without embedded relationships

Time to complete question 1 (mins)	*Time to complete question 4 (mins)*
15	5
43	
15	10
13	3
25	5
15	13
7	8
24	1
19.625	6.429

Set B - With embedded relationships

Time to complete question 1 (mins)	*Time to complete question 4 (mins)*
6	4
15	5
6	2
15	8
7	4
20	15
20	6
6	2
11.875	5.75

Table 11 (a, b). Times taken to complete questions 1 and 4 for embedded and non-embedded documentation

Lastly, participants were asked to rate the difficulty of compiling the lists for questions 1 and 4. They were asked to rate the diffculty by placing a mark in the box that best described both how difficult and how complicated they felt the task of compiling each of the lists was. For example, if a participant felt that compiling the list in response to question 1 was slightly easy and quite simple, they should mark the table as shown in Table 12 (In this example, the participant has indicated that the task was slightly easy and quite simple).

	extremely	quite	slightly	neutral	slightly	quite	extremely	
easy			√					difficult
simple		√						complicated

Table 12. Rating how difficult the participants found it to compile the lists of answers.

Participants ratings were scored from 1 - 7, corresponding to each rating, where 1 represents extremely easy/simple and 7 represents extremely difficult/complicated. The ratings given by each participant are shown in Table 13. Blank spaces in the tables indicate that participants did not supply an answer for this question. Averages for each dimension are shown at the bottom of the table.

Ratings of difficulty (1 - 7)
Set A - Without embedded relationships

Question 1		Question 4	
Easy - Difficult	*Simple - Complicated*	*Easy - Difficult*	*Simple - Complicated*
5	5	5	6
2	5	1	2
3	4	2	2
6	6	5	5
4	4	2	2
4	4	4	4
3	3	5	3
6		2	
4.13	4.43	3.25	3.43

Ratings of difficulty (1 - 7)
Set B - With embedded relationships

Question 1		Question 4	
Easy - Difficult	*Simple - Complicated*	*Easy - Difficult*	*Simple - Complicated*
4	2	2	2
4	4	4	4
4	4	3	3
2		1	
5	5	2	2
5	5	4	3
6		2	
2	2	2	2
4.00	3.67	2.50	2.67

Table 13 (a, b). Ratings of difficulty given by participants with respect to the effort involved in compiling answers

There are no significant differences in ratings of difficulty between the two different types of documentation.

Lastly, participants were asked to rate how much of the contextual information as a whole they felt had been used to influence the design and to rate how much of the design had been influenced by the context. Unfortunately too few participants answered these questions to make analysis of the results received worthwhile. This was due to the design of the question sheets. These final questions were on another sheet of paper which some participants failed to notice.

5.5.4 Conclusions and Discussions

While the results returned by participants in terms of efficiency, validity, invalidity, time and difficulty all appear to favour the documentation containing the embedded relationships, none of the results are significant. Hence, it is not possible to make any firm conclusions about the effect that the presence of embedded relationships has on participants' abilities to understand context and design documentation described in a similar manner to that used in this experiment. This is an important distinction to make, since the lack of significant results could be due to the relatively simple design descriptions used in the experiment (plain English prose, less than 1 page of A4 to describe the main features of a design) and the relatively small number of relationships represented between context and design (13). However, it would have been difficult to use more complex design and context documents and still be able to recruit a sufficient number of participants. Had more complex design and context descriptions been used in the evaluation, each participant would have had to spend more time on completing the evaluation. They would also require greater levels of expertise and experience in design, in order to be able to make sense of the complex design and context descriptions. Thus, these requirements would have limited the number of participants that would be able to take part in the evaluation since there are few people who have the necessary design skills and are able to give up a significant proportion of their time to perform the evaluation. Furthermore, the more complex the design documents are, the more participants rely on their design expertise and experience to make sense of the documents. Hence, participants' expertise and experience would likely interfere with their abilities to answer the questions asked in the evaluation, thus making it less clear that any differences between answers supplied by both groups would have been solely caused by the presence of embedded relationships in the documents.

Perhaps one of the reasons for the lack of significant results between the answers returned by participants is that explicit relationships are not required when participants (designers) are given design documents that contain very little in the way of detailed decriptions of either the context or the design. If this is the case, the question then arises of determining the minimum level of complexity of design and context documents for which the presence of explicit relationships between both types of document will provide benefit to designers. The next evaluation partly addresses this issue by utilising design and context descriptions that are more detailed and that contain more relationships between context and design. The evaluation uses LD to evaluate how other people can use it to make sense of contextual data and its relationship to a design.

5.6 A study of LD in use

The above evaluation used paper based representations of context, design and the relationships between the context and the design. No firm conclusions were reached about the benefits of explicitly representing the relationships between context and design. One of the possible reasons for the lack of significant results was the fact that relatively simple (in comparison to standard design projects) design and context descriptions were used. This section describes a study that attempts to fix the benefits provided by making the relationships between context and design explicit, and at the same time evaluate the use of LD in an authentic development project. Participants were asked to use LD to review a set of design and context descriptions concerning the Olympic Messaging System (OMS). This evaluation provides data concerning the way that people use the tool, what they think of the tool and what they believe are the advantages and disadvantages of making the relationships between context and design explicit.

5.6.1 Method

Due to the time that participants were expected to take to complete this evaluation (at least an hour each) and the levels of design skills and experience required, it was not possible to run the evaluation with the required numbers of participants to achieve significant results given the poor availability of suitable participants for this evaluation. Hence this was an informal evaluation of LD. This evaluation was used to collect data regarding users' subjective opinions towards the tool and the concept of explicit relationships between context and design.

Five participants were selected to take part in this evaluation. Four of the participants are research students in the department who are working in HCI, Databases or Information Retrieval. Through their research projects, each participant has experience of software design and has had some exposure to HCI practices. The fifth participant is a software designer working in industry, who has wide experience in applying HCI techniques both in academia and in industry. None of the participants took part in the paper based evaluation described above.

Approximately half an hour was spent prior to the evaluation describing LD to the participants. Participants were shown what LD does and how it works and could ask any questions. Participants were then given a brief description of the OMS for them to read, together with a set of questions to answer in order to demonstrate their understanding of the OMS. They were then given a Project Summary document which described the current state of development of the OMS.

Participants were then given access to the set of documents created using LD which describe the OMS system and its context, as well as the relationships between the different descriptions. All descriptions were represented and accessed via LD. NUF, QOC and scenarios were used to describe the design of the OMS. Participants could access the appropriate editor for each notation to examine the design. The Contextual Focus editor was used to describe the relevant context of use for the OMS. Relationships between the context and the design had been created prior to the evaluation and these could be inspected at will by the participants by examining the relationship editor. Participants were asked to use each of these editors in whatever manner they saw fit to determine the influence that the contextual information had had on the design. They were then asked to use the result of this investigation to suggest and make a design change such that the contextual information would have a greater influence over the design.

As participants were performing the task, they were asked to describe their actions and their reasons for performing those actions. Throughout the evaluation participants were able to ask questions relating to the tool (for example, how do I update this contextual factor?) but not related to the meaning or interpretation of the context and design representations, nor the relationships between the context and the design. Hand written notes of participants' problems related to using LD and interesting events that occured during each evaluation were taken by myself. Of particular interest were occasions when LD did not behave as participants expected it to, when participants were unable to predict the behaviour of LD or when participants attempted to use LD in a new and novel way. Additional functionality was added to LD for the purposes of this evaluation which enabled LD to compile a log of actions directed at LD by the participant. The log recorded menu selections, button presses in windows, item selections etc. For example, if the user selects a new relationship in the list of relationships in the relationship editor, the log appends the entry "Opened new link :" followed by the name of the link into the log file. Similar entries are created for other actions (e.g., when the user inspects a contextual factor (by selecting it in the Contextual Focus editor) or when the user selects one of the menu options from the relationship editor). The time of each action was also recorded alongside the action in the log.

5.6.2 Results: Overview

In contrast to the first evaluation described at the start of the chapter, every participant was able to use LD and complete the task specified. However, each participant commented that there was a lot of information for them to absorb in the time given for each evaluation. On average, each evaluation took approximately 90 minutes, including the 30 minutes spent explaining both LD and the OMS prior to the evaluation proper. The main results from the five evaluations are summarised below.

The results are split into three main categories, 'Relationships between context and design', 'Design' and 'Context'. Within each category are a number of sub–categories which collect together common issues that were raised during the evaluations.

5.6.3 Relationships between context and design

This section collects together all those issues that arose during the evaluation concerning the relationships between context and design. Issues arose concerning inspecting, representing, understanding, using and comparing relationships.

Inspecting relationships between context and design.

Every participant understood the concept of making the relationships between contextual information and design explicit. When inspecting the relationships, each partcipant used the name of the relationship and the explanatory text to try to understand what the relationship meant. However, with the exception of the last participant, participants did not focus on inspecting the different relationships. On the whole, participants seemed only to inspect relationships after they had created a new relationship, and in most cases, this was only to ensure that the relationship that the participant had just created had been added to the list of relationships. This would suggest that the representation of the relationships did not clearly represent the information that participants desired most or that the focus of concern (i.e., on individual relationships) was inappropriate. Participants may have preferred to focus on design and context documents such that they could inspect the list of relationships that involve a certain document. In other words, a focus on the 'web' of relationships between documents may have been more appropriate rather than a focus on individual relationships. The next section provides support for this view.

Representing relationships between context and design

All participants suggested that the representation of the relationships was less than ideal and that this may have been one of the reasons for the lack of attention paid to the relationships by each participant. Participants expected to be able to access the relationships from the actual design document they were inspecting (in other words, they wanted to focus on the design document and see the web of relationships that this document is involved in). For example, participants would have preferred to be able to inspect an individual question in the QOC document and to manipulate this question directly such that the participant could be shown all the relationships that question is involved in. Currently, relationships are less direct, in that participants have to look for a 'part' of the QOC document that contains the required question and then go to the relationship editor to inspect all the relationships, searching for any relationship that contains the required part. The individual question from the QOC document is not directly involved in the relationship. Because of this, there is a certain amount of 'overhead' involved in interpreting a relationship. Participants first of all have to identify the document and the particular part that the relationship involves, then they have to open up the document, look for the part that is involved in the relationship, inspect the contents of that part and then find the contents within the document. This effort may have discouraged participants from investigating relationships in any great detail. Furthermore, even if this overhead had been considered acceptable by participants, they would still have had to use the relationship editor to inspect the relationships rather than inspect the relationships from the document that they relate to. Hence, the overhead in working out which part of the document is involved in the relationship and the indirection between the list of relationships and related document led to a less than satisfactory representation of relationships between context and design.

One of the consequences of the poor representation of the relationships has already been described (participants tended not to inspect the different relationships). The next section describes another (possible) consequence of the representation of the relationships between context and design, namely that of understanding the relationships.

Understanding relationships between context and design

The representation of the relationships was such that the meaning of the relationships was not always entirely clear, at least to one participant. He questioned the meaning of some of the relationships between context and design. As he was creating a new relationship, he wondered if the relationship should be viewed as a general guideline, suggesting that some piece of contextual information should have some kind of influence over the design, or if the relationship should be viewed as describing the historical development of a design, in other words to back-up design decisions and design features of a system. He decided that relationships could be used to describe a guideline and the historical development of a design, but that it may be difficult to tell the difference between both.

Making use of the relationships between context and design

However, the poor representation of the relationships between context and design did not prevent participants from seeing the benefits that could be gained from the relationships. Even though most participants did not inspect individual relationships, there was a general feeling that creating and maintaining a set of relationships was worthwhile. Two participants said that the relationships could be useful in maintaining a system, although one of them suggested that creating the links would be annoying, adding to the designers workload and hence may put designers off from doing so.

Another participant thought that the relationships could be useful in creating a design, as well as maintaining a design. She discussed how she tends to design software in an iterative fashion, and how some design features can make elements of context relevant. She attempted to use the tool to indicate that some features of the OMS (such as the requirement to enter a password and remember a country code) had an influence on the amount of training required by users of the OMS. She felt that the different design features of the OMS made this feature of context important, since as more features were added to the OMS, more training would be required. She was encouraged to think hard about ways in which the design could relate to context because the relationships between context and design were explicit. Therefore, it wasn't enough just to say that there is some kind of a relationship. The details of that relationship had to be made clear. Thinking in this way helped identify aspects of context that were relevant, which had previously been considered irrelevant, and which would now relate to some part of the design. Now that a new piece of relevant context had been identified, other parts of the design (new or existing) would have to be identifed that relate to the context. The process repeats iteratively, matching the participants description of how she tends to tackle a design problem.

Another benefit of the explicit relationships in such an iterative process, as well as encouraging the process, is that they serve as reminders of work that has been done and work that remains to be done. For example, if a piece of context is involved in many relationships with the design, then it is unlikely that there are further relationships that this particular piece of context will be involved in. On the other hand, if the context is only involved in a few relationships, then it is likely that there are more relationships that should be specified between the context and the design. Hence the number of relationships serves as a 'reminder' of further work that may need to be done.

Another participant suggested that the presence of the relationships and the ability to manipulate them in LD, makes it possible to do certain analyses that previously were very difficult to do. For example, two different designs could be evaluated in terms of how much of an influence the contextual information has had on them. Assessments could be made with respect to how well designs fit certain elements of context. For example, different designs could be evaluated in terms of how well they fit a certain environment. Or, the designs could be evaluated in terms of how well they match the capabilities of different types of users or stakeholders. These evaluations could then be used as the basis for merging different designs and retaining the best elements of each such that the resulting design matches both the specified environment and group of stakeholders for example.

Using relationships to show a lack of influence

While the relationships can be used to show the influence that context has had on the design, one of the participants asked if the relationships could be used to show that some element of context must *not* influence the design. For example, he was concerned that cultural issues should not interfere with the design of the OMS and wanted to create some kind of relationship from the 'Culture' factor in the contextual focus such that the relationship would indicate that culture has not influenced and should not influence the design. He suggested that a relationship be created to 'bottom' or 'zero'. In this way the relationship indicates that culture must have no effect throughout the design. This is different from simply not using a particular contextual factor since a factor may not be used because it is irrelevant. If a piece of context is irrelevant, then it should have no bearing whatsoever on the design. In other words, it does not influence the success of the design in one way or another. However, relating an element of context to some 'zero' or 'bottom' value indicates that the success of the design is affected by the element of context and that the success of the design is proportional to the lack of influence that the element of context has on the design. In other words, the more influence that some element of context (say cultural issues) has on the design, the less successful the design will be. In the case of the OMS, culture clearly is relevant to the design of the OMS system since cultural issues must not interfere with the design of the OMS.

This relates in some sense to the understanding relationships section since using a relationship to show a lack of influence in the style suggested in this section is similar to using a relationship as a guideline.

Comparing relationships

One of the difficulties experienced by participants concerned judging the significance or the weight of each relationship. Since relationships are created from and to different parts of a document and different parts can contain any number of primitive elements of a document, it is just as easy to create a relationship from a part of a document that contains ten elements of that document to a part of another document that contains five elements of the other document, as it is to create a relationship from a part of a document that contains only one element of a document to a part of another document that contains one element of the other document. It is clear that these relationships are different, since in the first relationship, ten elements of a document are related to five elements in another document as opposed to the second relationship in which only one element of a document is related to one element of another document. However, in LD, this difference is not made clear. Participants suggested that such differences should be made obvious, such that the influence of context over the design and the significance or 'weight' of individual relationships can be more accurately determined.

Summary

With respect to creating and maintaining relationships between context and design, the main result of the evaluation was that the representation used for the relationships was inadequate and did not encourage participants to inspect the relationships nor did the representation help participants in understanding or comparing relationships. However, this did not prevent participants from seeing the benefits that could be gained from making the relationships between context and design explicit, nor from identifying novel and interesting uses of the relationships.

5.6.4 Context

This section collects together issues relating to context. It describes issues involving interpretation of contextual factors and making context explicit.

Group interpretation of contextual factors

While none of the participants performed their evaluations simultaneously, it was possible to examine some of the issues that might arise if LD was expanded into a multi–user tool and used by a design team. Each participant interpreted some of the contextual factors differently from one another. One of the commonest factors to be interpreted differently was the 'frequency of use' factor. Some participants thought that it referred to the number of times that the OMS would be used, while others thought that it referred to the number of times that individual users would use the OMS. Clearly in a group situation, differences in interpretation could lead to problems, with some members of the group working with one interpretation of the context and other members working with a differerent interpretation of the context. Disagreements could arise about the relevancy of individual factors, simply because one group believe the factor means one thing while another group believe the factor means something else.

Making context explicit

These differences in interpretation were highlighted while participants inspected the contextual focus, an explicit listing of individual contextual factors that are potentially relevant to the design. Without such an explicit listing of potentially relevant contextual factors, differences in interpretations may not have been made so clear, since participants may not have felt compelled to provide an interpretation for factors that aren't listed. The explicit listing of potentially relevant factors also suggested a number of factors to at least one participant that they had never thought about. For example, the lighting conditions around the OMS kiosks had been identified as potentially relevant to the design of the kiosk. One participant said that they would never have thought of that factor, but after seeing it, could understand how it would be relevant to the design of the kiosk.

Summary

With respect to the contextual information, the main results from the evaluation concern interpreting context and making context explicit. Both are clearly related. By making context explicit, participants were encouraged to think about how different elements of context could be relevant to the design. They were also encouraged to discuss what each individual element of context meant. In effect, making context explicit means it is 'up for grabs' or identified as a potential discussion point. Hence, explicit descriptions of context could encourage consensus amongst a group of designers, regarding the particular interpretations to be placed on the context.

5.6.5 Design

This section collects together those issues that relate to the design. Issues involve understanding the design, describing the physical configuration of a design and querying a design.

Understanding the design

In general, participants seemed to focus on the contextual information and the QOCs to try to understand the design and how it related to the context, instead of inspecting the relationships. This can be seen from the log files in Appendix D. Each line beginning with "Moved to factor" or "Moved to group" indicates that the user moved to some other area within the contextual focus[11]. Lines beginning with "In Olympic QOCs" indicate that the user inspected some aspect of a QOC representation. All the log files demonstrate that the majority of the interaction focussed around the contextual focus and the QOCs. Given a better representation of the relationships between context and design, as described earlier, participants may focus more on the relationships and less on the context and QOC documents. However, this analysis would suggest that the information provided by the

---

[11] Due to the implementation of the contextual focus view, if a participant wanted to move from one area of the focus to another separate area, this could necessitate a number of successive moves (eg two moves up to get from the current position to the top level of the hierarchy and then two moves down to get from the top position in the hierarchy to some other leaf node). Hence to move from one position to another could use up four moves. So, the number of lines containing "Moved to factor" or "Moved to group" do not indicate the exact number of inspections of individual contextual items, but they do indicate the relative number of inspections, whether there were many or just a few such inspections.

context and QOC documents is useful in understanding a design and that this kind of information should be available to designers.

It would appear that what users examine in LD will depend on their goals. In this evaluation they were attempting to understand the design so they focussed on the QOC diagrams and how they related to the contextual focus. It is likely that this focus would be different if users had different goals. For example, users may focus on the design specifications and how they relate to the contextual information when they are designing.

Describing the physical configuration of a design

During the evaluations a participant came across the contextual factor 'Lighting – sunlight' in the contextual focus. The participant said that they would never have thought of this factor had they been asked to identify the relevant context, but after seeing the factor, could understand how it was relevant. The participant attempted to use this factor to modify the design of the kiosk, such that the video screen on the OMS kiosk had an anti--glare screen but LD only provided a means to describe the computer based system and not the physical appearance or layout of the OMS. Therefore it was not possible to show how the lighting conditions could influence the design of the OMS given the current set of design documents relating to the OMS. Another participant came up against a similar problem when they wanted to enclose the OMS kiosk in a sound proof booth so as to minimise the disruption caused by the potentially noisy environments that the OMS kiosks could be situated in. Again, since LD only provides access to the design of the computer based system, this design modification could not be represented with LD.

Querying the design documents

Most participants suggested that some form of querying over the design documents would be useful such that individual specifications could be located easily. One of the reasons that this may have been an issue for the design documents but not for the context or the relationships betweeen context and design is that the design is represented via a number of different representations. Hence, in order to search for part of a design specification, participants had to first of all determine what notation the specification would be described in and then search through this notation for the required item. In contrast, the contextual information is all collected in one place, meaning that if a participant wanted to look for an individual part of the context specification, they would only have to look in one place. Hence searching for part of a design specification is not as straightforward as searching for a part of the context specification, and requires tool support.

Summary

With respect to the design, there were three main issues. These relate to understanding the design, describing the physical configuration of a design and querying the design documents. The evaluations suggested that the context and QOC documents were heavily used in order for participants to generate an understanding of the design. It suggests that the information represented in the context and QOC documents is of definite value to designers. The evaluation also suggested that in order to make full use of contextual information, designers must be able to use it to influence all aspects of a design, including the physical appearance and configuration. Lastly, on a technical issue, participants suggested that it would be useful to be able to perform queries over the design documents, in order to locate individual specifications.

5.6.6 Summary

In summary of all the issues described above, all participants agreed that making the relationships between context and design explicit was worthwhile since the explicit relationships made it possible to accurately judge the influence of context over the design but that the relationships should be made more direct and should be visible from the different design documents, instead of one central area. Participants used relationships in different ways (to record the design process or to maintain the system) and with different meanings (as a guideline or to describe the development of the design). Participants suggested that the 'significance' or 'weight' of individual relationships should be made clear since this would help in increasing the accuracy of the different assessments that are made possible by explicitly recording the relationships between context and design. It was also suggested that relationships should not only show the influence of some context but can be used to show the deliberate lack of influence of context. Differences in interpretation of individual contextual factors were clear and lastly, the requirement for other design descriptions that describe the physical appearance and layout of a system was identified.

5.6.7 Conclusions and open issues from third study

From the results described above, there are a number of conclusions that can be reached. These are summarised below. The results also raise a number of issues from which conclusions cannot yet be reached. These are also described.

Creating direct representations of relationships

It is clear from the evaluations that the representation of the relationships between context and design needs to be improved, such that relationships are directly related to the context and the design. Major problems were experienced related to the current representation of relationships. Participants felt that they wanted to see the relationships in the documents that they were inspecting, rather than in some separate document. Participants suggested that design documents could be annotated or modified such that the relationships were directly related to the design documents. One participant suggested that a hyper–text representation could be used, perhaps underlining areas of a design document that are involved in relationships with other documents. In this way, users would be able to inspect both the context or design document and the relationships that involve that document, at the same time. Currently, users can only inspect a document and its associated relationships at different times since the relationships are represented separately from the design and context documents.

However, while it is clear that a more direct representation of relationships is necessary, it is not clear if direct manipulation of the relationships is necessary. If relationships are represented similarly to hyper–text links, should users be able to navigate the different design documents in typical hyper–text fashion (clicking on underlined areas)? Certainly, no participant suggested that this would be useful, but this could be simply because the direct representation of the relationships was not presented to participants, meaning that they would be less likely to think about navigating in this way. Allowing users to navigate via the relationships would provide a different view and understanding of the design (i.e., it would allow users to understand the design from the point of view of the relationships between different parts of the design) but this could be at the expense of different types of understanding (i.e., understanding what each part of the design means). Allowing users to navigate through design documents also introduces the possibility of users getting 'lost' in the maze of relationships and design documents. Currently, this is a problem that does not exist in LD since it does not support direct representation and manipulation of the relationships between context and design.

Relationships could be represented directly, by underlining areas of design documents or by other appropriate modifications, without allowing direct manipulation or navigation through the relationships. Instead, users could perhaps click on an underlined area of a design document to see where this document is related to. The user could then manually inspect the appropriate related document, by opening it themselves in the usual way (from the main menu or by double clicking the document icon). Since this requires extra effort and thought on the part of the user, in contrast to the direct navigation style, users may be able to maintain a better 'map' of the collection of documents and how the document they are inspecting relates to the other documents. However, no firm conclusion can be drawn about this issue at the moment and further work is required to determine both the ideal representation of relationships and the kind of manipulation users would prefer given a more direct representation of the relationships between context and design (see Chapter 9 for a detailed discussion of further work).

The evaluations also suggest that further work is required to determine how best to represent relationships such that the weight or significance of a relationship is clear. This is something that LD does not currently support, since it represents every relationship in the same way, even if one relationship relates many parts of a document to many parts of another document while another relationship relates one part of a document to one part of another document. Participants suggested that the difference in the weight or significance of a relationship contributes to differences in assesments about the relationships.

Usefulness of relationships

It is encouraging to note that no participant suggested that making the relationships explicit was not worthwhile. Indeed, one participant saw the benefits offered by the relationships in that they presented an opportunity to perform a number of different analyses over the design and context representations (e.g., determining how well a design fits certain aspects of context in order to perform detailed comparisons with other designs). However, the difficulties that were experienced in relation to the representation of the relationships mean that no firm conclusions can be reached about the worth of making the relationships between context and design explicit.

Flexibility and creative use

It is possible to conclude from the evaluations that flexibility in the use of LD is desirable and that the current LD tool supports this. Furthermore, this flexibility encourages creative use of LD in its users, in that they are not heavily constrained in terms of what they can and can't do with the tool. For example, some participants felt that the tool would be useful in maintaining a design and others felt that the tool would be useful while actually performing the design. Since the tool does not force any conventions on its users in terms of the way that they use the relationships between context and design, users can adapt it to their needs.

Further examples of the flexibility and creative use of the tool are provided by the different meanings that at least one participant ascribed to the relationships. He wondered whether the relationships were used to guide the development of a design or as an archival tool, to describe the historical development of the design. He realised that LD allowed relationships to be interpreted in both ways, since the tool does not enforce any particular interpretation on relationships. Hence LD allowed him to be creative and to interpret relationships in the manner he saw fit. Hence the flexibiliy of the tool allows users to be creative. However, the tool does not support all potential uses. The next section describes ways in which LD constrains the creativeness of users, by restricting what they can describe.

Restricting what can be described

It is clear from the evaluations that one of the major ways in which LD does force a particular convention or restriction on its users is in terms of the design documents supported by the tool. In the current LD tool, only QOC, UAN, NUF, Scenario and Context editors are supported. Users cannot use any other kind of notation or technique to describe the design or context. Chapter 3 identified the need for LD to be adaptable, such that any appropriate notation or technique can be used by a design team.

The evaluations indicate that LD must support this feature. When certain contextual factors such as 'Lighting - sunlight' and 'Noise' were identified as being unused in the design, most participants thought about changing the physical layout or structure of the OMS kiosk so as to accomodate these factors. One participant thought about enclosing the OMS kiosk in a sound proof booth, so as to minimise the effects of the noise from the environment surrounding the kiosk. However, it was not possible to describe this feature in LD since the notations supported by the tool only allow computer based specifications to be described. It is clear that contextual information can have a wider reaching impact than a computer system alone, so designers must be able to describe other non–computer based features of a system and how these relate to the contextual information.

Further work is required to identify methods for supporting the use of any appropriate notation in LD, where appropriateness is defined by the design team using LD, such that the design team can describe what they believe are all the important features of a design and how it related to its context.

Being explicit

The evaluations suggest that explicitly listing the potentially relevant context will make users interpretations of these factors explicit, thus exposing differences in interpretation amongst a design team. As described in Cockton *et al.*, (1995), making the assumptions about relevant context explicit helps in creating consensus amongst the team about what the context means and how it can be used. In the individual evaluations, different participants used different interpretations of some of the contextual factors. Had these factors not been explicitly listed, the differences in interpretation may not have been so clear since participants may not have felt compelled to think about the factors.

However, by being explicit, there is a danger that people will be less inclined to think about factors that aren't listed, but perhaps should be. The evaluations bear this out, since no participant thought about adding factors to the list of potentially relevant context or trying to identify factors that weren't in the list but perhaps should have been.

Therefore, the evaluations suggest that being explicit about the potentially relevant context can be useful in creating consensus amongst a design team about what the factors mean and how they should be used, but that it can also lessen the amount of discussion concerning missing factors from the list. Further work is required to identify ways in which a list of potentially relevant context can be made explicit in such a way that it encourages the design team to think about and discuss other relevant factors not included in the list.

5.7 Discussion of the evaluations

Evaluating LD (via the hypothesis that making the relationships between context and design explicit will encourage designers to make more efficient use of contextual information in design) was not a trivial task. The complexity and unfamiliarity of LD and the idea of explicit relationships between context and design meant that it was not possible to recruit a sufficient number of participants to carry out a formal evaluation of LD. Instead a compromise was reached, in which LD was evaluated in an informal, subjective evaluation and the hypothesis concerning explicit relationships between context and design was evaluated by using simple paper and pen based representations of the context, the design and the relationships between context and design.

The obvious drawback to the compromise reached is that no formal evaluation of LD was ever completed, and hence no formal evaluation of the hypothesis concerning relationships between context and design was completed in a real world (ecologically valid) setting. However, experiences in attempting such evaluations suggest that a formal evaluation of LD would require a considerable period of time and effort in which to recruit and train participants in the use of LD. Recruiting the 'right' participant could be a time consuming task since there are many confounding variables that need to be controlled. For example, levels of design experience and expertise need to be controlled. Experience and expertise in the notations used in LD also need to be controlled. Further confounding variables are availability of participants, background knowledge, experience with any other similar systems, fatigue and motivation. These last two factors will have a substantial influence on the evaluation, since each evaluation could last a considerable amount of time.

In effect, the compromise that was reached reduced the number of confounding variables by reducing the complexity of the evaluation, but at the same time reduced the validity of the evaluation. The second evaluation that was described suffered in that the tools and tasks that participants were provided with did not relate to the tools and tasks that would be used in a real world setting. Hence, it is not possible to generalise the findings from this evaluation to other tasks, users or working environments.

In contrast, the informal subjective evaluation of LD required less effort, time and participants and yet uncovered more valuable information. Five participants were asked to use LD and describe their reactions to it. This evaluation uncovered a lot of interesting information regarding the ways that people may attempt to use LD and their reactions to recording and using explicit relationships between context and design. While the data obtained from this evaluation would not stand up to statistical analysis, it was of more use than the data obtained from the formal evaluation.

In general, formal evaluations of interactive software or systems may only be worthwhile when well specified, narrow aspects of the system are being evaluated and when the system being evaluated is straightfirward and familiar to participants. Preece *et al.*, (1994, p. 642) suggest that formal experiments tend to produce findings that are too specific and narrow to be of much use to designers in the process of design.

5.8 Chapter Summary

Three attempts at evaluating LD (and the hypothesis that making the relationships between context and design explicit encourages designers to make efficient use of contextual information in design were described). The evaluations were useful in suggesting ways in which LD could be improved and in identifying some of the issues involved in using LD. However, it was impossible to gather statistically significant data concerning the relationships between context and design, primarily due to the complexity and novelty of LD and the effect that this had on the availability of suitable participants to take part in the evaluations.

6. Chapter 6 Thesis Contributions

6.1 Introduction

This chapter reviews the main contributions of this thesis to the use of contextual information in the design of interactive computer systems. The contributions are organised in two categories. The first collects contributions related to the main aim of the thesis. These represent the findings of an investigation into creating and maintaining an explicit set of relationships between contextual information and design specifications throughout a design project. The second category of contributions relate to the investigation of other issues concerned with making *efficient* use of contextual information in design.

Before describing the contributions made by this research, it is worthwhile re–iterating the scope of the research from Chapter 1. This research is primarily concerned with creating and maintaining a set of explicit relationships between contextual information and a design. As a result of this focus, issues concerned with making efficient use of contextual information in design have also arisen, but this list of issues is by no means exhaustive. It is likely that a different focus, perhaps on using prototyping methods to make efficient use of contextual information in design, would raise a different set of issues. However, by focussing on creating and maintaining a set of explicit relationships between context and design, this research has identified several novel issues that are relevant to many design approaches.

6.2 Investigating contextual design

This thesis makes several contributions to issues related to creating and maintaining a set of explicit relationships between context and design. Since this is an approach which, to the best of my knowledge has not been thoroughly investigated prior to this research, the conclusions in this thesis lay the foundations for future work.

The thesis also contributes to issues related to using contextual information in design in general, whether or not attempts are made to explicitly relate contextual information with design. Most importantly, the framework for contextual design described in Chapter 6 adds to the current work on contextual design with its emphasis on generating an understanding of contextual information and using that understanding to produce a number of alternative design ideas.

While both areas (i) explicitly relating context and design and (ii) contextual design in general are related to one another, this section will review the contributions made to each area separately.

6.2.1 Explicitly relating context and design

This thesis has made several contributions to the work on explicitly relating context and design. These are listed and described below:

- Explicit relationships between context and design are worthwhile;
- Relationships exist both from context to design and from design to context;
- Available resources influence relationship creation;
- Comprehensive webs are not essential;
- Explicit relationships support design analysis.

Explicit relationships between context and design are worthwhile

Making the relationships between context and design explicit has been demonstrated to be a useful practice by this work. Three case studies were described in Chapter 4 that used LD to relate explicitly separate bodies of contextual information with associated designs. The case studies demonstrated that such explicit relationships are a useful aid in design, as inconsistencies and omissions between and within different design documents were discovered. The work in the case studies also suggested that active consideration of the various relationships between context and design:

- encourages greater consistency between design documents;
- promotes a better understanding of the way that the design is structured and its relationship with the contextual information.

In discussions with designers at Intel (Appendix F), designers agreed that creating and maintaining a set of relationships between context and design would be worthwhile and that it could help them perform validation tasks (with respect to determining how well a design fits the contextual information that it is based on) that they had experienced difficulties in performing previously.

In the validations of the tool described in Chapter 5, some participants felt that the explicit relationships would be helpful to a team of designers both during the actual design process and afterwards, during maintenance.

This thesis has demonstrated, using distinct and complementary methods, that there are advantages to be gained from explicitly relating contextual information and designs.

Relationships exist both from context to design and from design to context

As well as relationships from contextual information to design, relationships also exist from a design to associated contextual information. There are several types of relationships, each with different meanings. For example, there may be relationships between:

- key elements of context and key design features;
- ungrounded design features and elements of context relevant for a particular feature;
- ungrounded design features and elements of context relevant for any feature;
- 'incidental' context and design features (see below).

Of these relationships, the first two were encountered during the IT study in Chapter 3 and in discussions with designers at Intel (Appendix F). The other two were not encountered but are possible, as described below. Other types of relationships are also likely, but these have not been identified.

Relationships between key elements of context and key design features occur as a result of initial contextual data collection, which identifies key elements of context which must be addressed in the design. In the IT study (Chapter 3) one such key element of context related to the need to handle applicants' phone queries quickly and competently. As a result, one of the key features of the design was that the applications secretary should have quick and easy access to all relevant information, by storing the data centrally, rather than distributing it between two different computer applications and paper files stored in a filing cabinet. In this case, the context was used to identify a particular design feature.

Design features need not always be specified as a result of identifying some key context however. Design features may be specified because designers believe they are relevant and useful, but not necessarily related to the contextual data. Designers at Intel (Appendix F) described how some features of the prototype they developed in response to the families with young children ethnographic study were not suggested by the study, but rather by their own intuition and experiences. However, these *ungrounded* design features need some grounding in relevant contextual information. This may involve further data collection. This data can then be used to justify the design feature.

In some cases, contextual data collected as a result of some ungrounded design feature will only be relevant to that design feature. If the design feature is removed from the design at a later date, the related contextual data will no longer have any relevance to the design as a whole and can be discarded. In other cases however, contextual data collected as a result of some ungrounded design feature will be relevant to other design features. In this case, even if the original ungrounded design feature is removed from the design, the related context still has some influence over the rest of the design and should not be discarded. Hence a distinction is made between relationships from ungrounded design features and relevant context for those particular design features, and relationships from ungrounded design features and relevant context for any feature.

Lastly, *incidental* context can be used to provide additional support for a design feature. Incidental context is not crucial to the design and development of any design feature but may perhaps be used to 'fine–tune' a design. It is important to distinguish between incidental context and key context since gains provided by concentrating on incidental context are not likely to be the same as those provided by concentrating on key context.

It is important to distinguish different types of relationships between context and design, as although this cannot comprehensively determine the appropriate action that should be taken given any change to the design or context descriptions, they can provide designers with information that may be of use when making design decisions.

Available resources influence relationship creation

Making the relationships between context and design explicit requires extra effort on the part of members of the design team. Designers at Intel suggested that if the effort required was great, then designers may be less inclined to record all the possible relationships and only record those that were discussed at design meetings (Appendix F). In effect, designers would be recording relationships between context and design in order to keep a record of what was discussed at the design meeting, rather than as a way to show how the context and design can and should influence each other. One of the participants in the informal evaluations (Chapter 5) also raised this concern. However, both he and the designers at Intel agreed separately that recording the relationships between context and design would still be worthwhile, as long as the effort required was kept as low as possible.

Comprehensive webs are not essential

The benefits that can be gained from making the relationships between context and design explicit are not fully dependent on exhaustive and comprehensive coverage of all the relationships. Designers can create and use relationships when needed, on demand. Identifying and describing the important and most influential relationships between context and design is essential. Describing *all* the relationships between context and design is not. In the case studies in Chapter 4, an iterative design process was followed. Hence relationships were created as a result of the particular foci of each iteration. No efforts were made to exhaustively list all of the possible relationships. Only those relationships that were judged important were recorded.

By focussing on the important relationships and the important elements of context and prioritising these, designers can more easily manage project development constraints such as completion target dates by only concentrating on the most important elements of context in the time available, rather than trying to take account of all elements of context.

Explicit relationships support design analysis

Making the relationships between context and design explicit makes possible different analyses over the context and the design. One participant in the informal evaluations (Chapter 5) discussed how it would be possible to evaluate designs with respect to different elements of the context. For example, a design could be evaluated with respect to how well it fits the environment in which it is designed for, simply by focussing on those relationships that relate the design to the group of contextual factors describing the environent. Different, perhaps alternative designs could be compared this way. Such an evaluation might suggest that design X fits its environment well but that it does not fit the capabilities and attributes of its users. In contrast, another design, design Y, could be shown to better match the capabilities of the users but less so the environment. The two designs might be appropriately merged so that a resulting design fits both the environment and its users.

6.2.2 Using contextual information in design

This section collects together those contributions that the thesis has to make with respect to using contextual information efficiently in design. These are listed and described below:

- Explicit descriptions of relevant context encourage discussion;
- Structure of contextual data influences discussions;
- Contextual data supports generation and validation of alternative design ideas;
- Abstract definitions of context are of limited value;
- Designers need flexibility in representing context.

Explicit descriptions of relevant context encourage discussion

The thesis has shown that explicit descriptions of the relevant context and the ways in which the context has been used encourage discussion about the context, what it means and how it has been used, which in turn can help a design team reach consensus about the importance and value of contextual information. In the informal evaluations (Chapter 5), participants were asked individually to examine context and design descriptions for a particular design project and to propose modifications in response to a change in the relevant context. Each participant inspected the list of relevant contextual factors and created their own definition of each factor, so that they could determine how the factors should best be used in the design. While each participant participated in the evaluations individually, I sat beside them to offer any help during the evaluation and to note any significant events. Therefore the differences in participants' understanding and definitions of the relevant context were made clear to me, since I was able to compare each participants' definitions.

In contrast, the Customer Centred Design approach suggests that no explicit listing of individual relevant contextual factors should be made since such lists can bias designers when they initially investigate the context. Cockton *et al.* (1995) agree that explicit descriptions of relevant context can bias designers, but suggest that designers will never be able to investigate context without a pre–conceived idea of what that context can comprise. These tacit ideas will just as likely bias the designers as an explicit listing of contextual factors would, but at least when the relevant context is explicitly listed and described it is available for discussion so that inconsistent interpretations of the context can be identified and resolved between designers. The LD evaluation (see Chapter 5) makes evident that such discussions are possible, indeed likely, in appropriate circumstances.

Structure of contextual data influences discussions

GENEVA focuses on one, unified model of context from which to generate discussions about the context. Discussions are tied to only one model so that there are no concerns about designers being unaware of other related discussions taking place. This means however, that considerable effort may need to be spent in collecting contextual data into one model. This may not always be possible. Customer Centred Design uses a variety of models and it is difficult to see how each of these models could be merged into one coherent model. In contrast to the contextual checklist (Appendix A), both the Customer Centred Design models and the ethnographic models used by designers at Intel (Appendix F) create a certain amount of abstract structure over the contextual data which eases manageability of discussions, since discussions too can be structured and abstracted over also. In essence, the structure of the model of context will affect the discussions that take place about that model.

Contextual data supports generation and validation of alternative design ideas

The informal evaluations in Chapter 5 and the discussions with the designers at Intel, described in Appendix F, demonstrated that generating and validating alternative design ideas is a worthwhile practice. As proponents of design space analysis suggest, creating alternative designs opens up the design space, presents new possibilities and highlights problems in a current design, since there is now an alternative design which the current design should be compared with. The third stage in the framework for contextual design, described in Chapter 6, adds to traditional design space analysis in that it encourages designers to base their design decisions on relevant contextual data. In discussions with the designers at Intel, it was clear that validating design ideas against contextual information is something that they already attempt to do and for which computer based support would be extremely useful.

Abstract definitions of context are of limited value

Abstract definitions of context such as the definition offered by Wixon and Raven (1994) may not be of much use to designers. Such definitions are unlikely to be able to identify all the relevant contextual factors for any particular design project since each currently available definition focuses on some aspects of context at the expense of other aspects, which could be just as relevant in a design project. Chapter 2 discussed the problem of *scoping* with respect to different contextual design methods. Cockton *et al.* (1995) also discussed this issue and Clarke (1996) suggests that designers must be aware of differences in scope of each definition of context so that methods and definitions can be combined, thus ensuring complete coverage of all the relevant contextual information for a particular project. Without an awareness of the differences in scope of definitions of context, designers may be unaware of potentially relevant context that the definition they are employing simply ignores. A more concrete 'map' of context, such as the checklist shown in Appendix A, may help avoid the problem of ignoring relevant context, since it explicitly lists different elements of context, and can be referred to at any time by designers. In contrast, abstract definitions of context do not support explicit descriptions and rely on the designers to consistently interpret the definition in terms of identifying relevant and irrelevant context.

Designers need flexibility in representing context

Contextual data is varied and can be put to many uses. Designers need flexibility in the tools they use to represent context and the way it is used in a design. This was demonstrated in the informal evaluations (Chapter 5) when participants wanted to be able to describe the physical appearance and structure of the Olympic Messaging System in order to be able to show the influence that the context had on the design. A collection of flexible, interchangeable tools is probably of most use to designers in that the deficiencies of one tool can be compensated for in another. Such tools can also be tailored for specific subtasks. Just as one definition of context is unlikely to be adequate, so too one computer based tool is unlikely to be adequate to allow designers to do everything they want and need to do with regard to using contextual information efficiently in design.

6.3 Chapter Summary

This chapter has reviewed the main contributions of this thesis. They are split into two categories. The first category concerns the contributions made with respect to explicitly recording the relationships between context and design. It demonstrates that recording and maintaining the relationships between context and design provides benefits to designers in that it encourages them to think about and discuss the ways that context has been used. Such discussion and reflection can highlight inconsistencies and omisssions within and between particular design documents. By resolving these inconsistencies and omissions, designers start to make more efficient use of contextual information in design.

The thesis has shown that relationships exist both from context to design and from design to context. Each type of relationship conveys a different meaning and may require different actions as a result of some change to either the source or destination of the relationship.

Effective use of context is supported by the different analyses over context and design documents that are supported by explicit relationships between context and design. Such analyses can identify areas where further work in specifying the design, collecting further contextual information or using that information in the design would be beneficial. Computer based support could be necessary in order to maintain a set of relationships between context and design and such a support tool was described and evaluated, both in a number of case studies and informal empirical evaluations.

The second category of contributions relates to using contextual information in design in general, whether or not designers attempt to create and maintain a set of explicit relationships. Issues such as encouraging discussions about contextual information, and defining context were discussed.

The next chapter describes the future work that has been suggested by this research. In effect, the future work is another category of contributions. Given the small amount of research that has been carried out into explicily relating context with designs, this thesis has possibly raised more questions than it has answered. The next chapter describes these questions and issues and suggests ways in which they may be addressed.

7. Chapter 7 Further Work

7.1 Introduction

This chapter describes further work that has been suggested by the work presented in this thesis. Chapter 5 described a series of experimental validations that were performed on LD. These validations demonstrated the potential benefits offered by LD but also raised a number of questions about the tool.

7.2 The Literate Development Tool: Further Validations and Extensions

The previous chapter described a series of validations that were performed in an attempt to demonstrate the benefits offered to designers by LD. These validations raised a number of issues that need to be addressed in the next stage of development of LD.

7.2.1 Increasing the flexibility and openness of LD

There are many different representations and techniques used in HCI development. For example, notations such as QOC (MacLean *et al.*, 1991) and UAN (Hartson *et al.*, 1990) are used to describe the rationale and behaviour of a design respectively. Storyboards and scenarios (Carroll, 1995) are used to describe situations in which systems are used and how these situations may affect the design of the system. Each of these techniques was used in at least one of the case studies described in Chapters 3 and 4, and Appendix F.

Different design groups use different methods for several reasons. One design group may favour a particular notation since each member of the group has been trained in the use of that method. Another group may prefer to use a software application such as Microsoft PowerPoint to draw up a series of storyboards or scenarios of use quickly.

Thus design teams typically have good, practical reasons for their choice of design tools. Therefore, if LD is to be adopted by a design team, it must be compatible with existing tools used by the team.

The current version of LD supports the QOC, UAN and NUF notations, and also provides support for recording scenarios and descriptions of context. In order to provide support for other notations and tools that design teams use, these tools need to be integrated with LD in some way. For example, using CORBA (Object Management Group, 1996) compliant software and tools, design editors and tools can share data with tools such as LD and can be distributed across the workplace. Other distributed, component software technologies exist, such as Apple's OpenDoc (Apple Computer, 1995) and Microsoft's OLE (Williams, 1994).

While component software technologies will allow different applications to share objects (therefore allowing LD to be notified of changes to design specifications recorded in different tools etc.), LD must also represent relationships between design and context specifications without requiring major (if any) modifications to the tools used by designers. For each separate design tool or editor, LD could maintain a transparent layer containing representations of the relationships that involve the particular tool or editor. By drawing this layer over the editor, the user will be able to see both the contents of the editor and the involved relationships (see Figure 36). For example, a rectangle could represent the source or destination of a relationship. The LD layer simply contains a rectangle of appropriate dimensions and position that, when placed over the design editor, highlights the appropriate portion of the design specification as source or destination of a relationship.

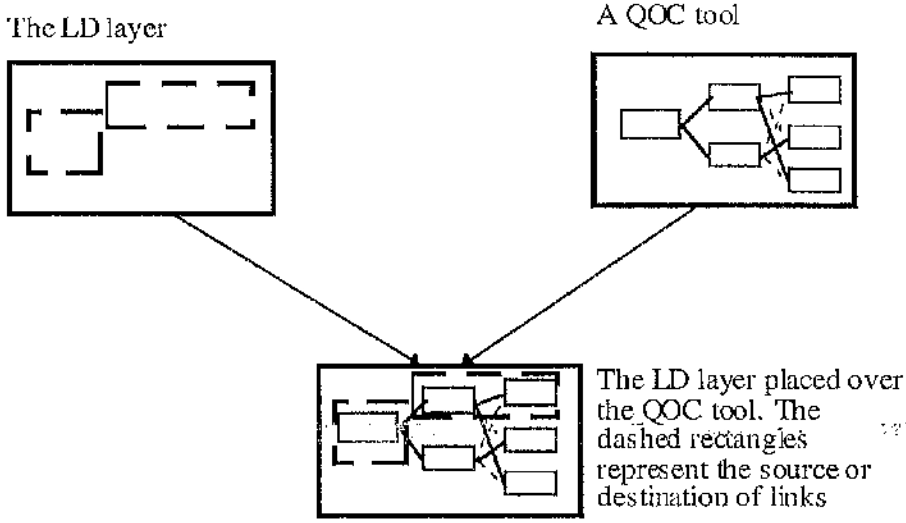


Figure 36. An example of an LD layer placed over a QOC tool.

Some form of communication between the design editors and LD is required, so that the web of relationships between objects can be maintained. When one object changes, other objects that are related to it need to be inspected and possibly modified. Since the only way that these objects are tied together is through LD, communication between individual design tools and LD is necessary. Communication is also required for lower level changes. For example, changes in the editors (e.g., scrolling or moving parts of a specification) need to be reflected in appropriate changes in the LD layer. Furthermore, manipulation of the LD layer, (e.g., double clicking a rectangle) might necessitate some action in the design editor (e.g., opening up another design specification or revealing details of the current specification). Such communications could be supported by use of inter–application scripting, such as AppleScript for Macintosh computers.

Therefore, to increase the flexibility and openness of LD such that designers can represent designs using tools and techniques they are most familiar with, tools must suport inter–application communications and object sharing. While this necessitates use of computer based tools, designers could still use paper and pen to specify their designs and then translate these representations into electronic form, using an image scanner for example. The resulting scanned images could then be manipulated using an editor that supports communication and object sharing.

So far, the focus on increasing the flexibility and openness of LD has concentrated on technical problems. However, further work is also required to identify non–technical requirements that different notations and techniques must support such that they can be successfully used. For example, the notations and techniques must easily provide a means to create anchors for the sources or destinations of relationships. It should be possible to specify these anchors at varying levels of detail with respect to the notation or technique used. It is not yet known what further requirements will be, but these need to be determined so that the scope of LD can be mapped out and so that developers do not waste time building CORBA or AppleScript compliant design tools, only to find that the tools are not compatible with LD.

7.2.2 Investigating the requirements for representations of context and design

The contextual checklist that was used in LD was compatible with the implementation of the relationships between context and design. Identifying individual parts of context that could be combined into the source or destination 'anchor' of a relationship was relatively straightforward. It is not clear however, if this will be the case with other representations of context and design. It is possible that designers may wish to use some medium in which to represent contextual information that may not be well suited to the particular representation of relationships used in LD (for example, video). Further work is required to identify the requirements that a contextual or design representation must satisfy if it is to be used successfully in the current implementation of LD. Without any knowedge of such requirements, it is difficult to say with any confidence whether or not a particular representation can be used.

Given such a set of requirements, it may be possible to identify modifications that are required to the implementation of relationships between context and design used in LD. Such modifications will help increase the flexibility of the tool by increasing the types of representations that can be used. The effort involved in making these modifications needs to be accurately estimated, so that the design team can judge whether or not the effort is worthwhile. The requirements for representations could be derived such that they help in making these judgements.

7.2.3 Improved representation of relationships between context and design

The validations demonstrated that participants would prefer a more 'direct' representation of the relationships between context and design so that relationships would be tied to the part of a design document that they originate from, rather than being described separately and removed from their source and destination design documents. The best representation for the relationships needs to be investigated.

The validations identified two dimensions across which representations could differ. These are *directness* and *navigability*. Directness refers to how closely the representation of a relationship is tied to the actual source and destination documents and associated parts. A representation that is not tied to the documents is said to be *indirect*. In contrast, a representation that is tied to the documents is said to be *direct*.

Navigability refers to the ability to manipulate the relationships to view other documents. If a user can manipulate a relationship to view individual documents, then the representation of the relationship is *navigable*. If the user cannot manipulate the relationship to navigate between documents and instead must use some other method, (such as manually opening a design document) then the relationship is said to be *non-navigable*. Thus there are four different kinds of representation according to the two dimensions identified. These are

- Direct and navigable. This type of representation most closely matches hypertext links in that relationships are directly tied to the associated documents and manipulating a relationship (i.e., clicking on it) takes the user from one document to another.
- Direct and non-navigable. This representation ties relationships to their associated documents but does not allow users to navigate between documents via the relationships. As an example, a user could view a pop up window displaying the appropriate relationships when they click on part of a document but would have to manually inspect the documents listed in the pop up window to investigate the relationships.
- Indirect and navigable. In this representation, relationships are represented separately from their associated containing document. For example, relationships may be presented

in a separate list. Users can scan this list and inspect the associated documents by clicking on the appropriate relationship.

- Indirect and non–navigable. This describes the representation used in the current version of LD. The relationships are represented in a list separate from the design and context documents. The user can only inspect these relationships and cannot manipulate them to open the associated documents.

Participants who took part in the validations suggested that the indirect representation used in LD impeded them in their attempts to understand the context and the ways in which it had influenced the design. Therefore, further work is required to investigate the form that the direct representation should take and, more importantly, whether the representation should be navigable or non–navigable.

Such an investigation should provide answers to some of the important questions regarding LD. For example, what are the different uses that the tool can be put to and how can these uses be best supported? In the validations described in Chapter 5, a participant questioned whether he could use the tool to define design policies or to create a historical record of a design project. Further uses of LD, if any, should be identified since these will have an effect on the relationship representations. If users use LD primarily to understand a set of design documents, navigation around the documents may be important. However, if users use LD to record the history of a design, then navigation may not be so important. The question of whether or not to provide users with the ability to navigate around the design documents using the relationships is an important one, since the familiar 'lost in hyperspace' problem, whereby users follow so many hyper-links that they become lost and do not know how to get back to the original document, will become an issue if navigation is supported. In particular, if users are using LD to navigate around a set of unfamiliar documents (say in a maintenance situation) then this problem could be severe, since users are not likely to understand all of the relationships, or even be aware of them, if they were not fully involved throughout the original design.

7.2.4 Change management using the relationships

During the evaluations described in Chapter 5, the possibility of relationships between context and design defining an agenda for developers was raised. It was suggested by one of the participants that the relationships could either be viewed as a historical account of the design process or as defining policies or guidelines that the design should adhere to. Currently, the representation of relationships in LD does not allow developers to easily distinguish between historical relationships and policy relationships. Therefore, it is difficult for developers to define an agenda for future design work, since it is difficult to tell if relationships are defining how the design *has* evolved or how it *should* evolve. One way in which this may be addressed is by assigning types to relationships.

Two different typed relationships have been identified as a result of this work. In Chapter 4 the types *supports* and *contradicts* were identified as a result of performing the case studies described. For example, in the OMS study, knowledge about the expertise of users (low expertise) and their experience with telephones *supports* the option to provide some form of training. At the same time it *contradicts* the option to support complex functionality, such as setting up conference calls.

Types provide further information about relationships and their meanings. This information could be used by developers to interpret relationships and to determine future actions as a result of change. For example, if, in the OMS study, users' experience with telephones increases dramatically, this may no longer support the option to provide training. Hence the relationship may be (re)moved. However, it may not always be possible to determine the action to take in response to a change simply by examining the type of a relationship. For example, if users' experience with telephones only increases slightly, then the option to provide training is still supported.

Other types of relationships may need to be identified such that the action to take after some change can be determined by examining the type of a relationship. Further types can be identified by referring to the literature describing typed links in hypertext systems. For example, Conklin and Begeman (1988) described the gIBIS system, a graphical design rationale capture tool. They list the following typed links that users of gIBIS can employ when describing design rationale

Responds–to
Supports
Objects–to
Questions
Replaces
Generalizes
Specializes
Suggested–by
Other

It is clear that the second and third types described above match the *supports* and *contradicts* types identified during the case studies, but further work is required to identify which of the remaining types would be useful to users of LD and to identify ways in which these types would be used. Further case studies could be performed to investigate the ways in which users would use these types. The case studies could be used to identify what users expect to happen as a result of assigning types to different relationships. Do users expect automatic updates or constraints to be placed on the relationships as a result of the types assigned to the relationship?

However, assigning types to relationships may not always be straightforward. Conklin and Begeman (1988) motivate the 'Other' type of link in the gIBIS system by describing difficulties that people have in structuring 'new' information or information that has not yet been used. Being forced to assign a type to a link or a relationship forces people to think in one particular framework (the framework suggested by the available types). Such constraints are unreasonable in some situations, particularly when people are working on a new problem, with information that they do not fully understand. Conklin and Begeman (1988, p. 325) claim that "the early phase of consideration of a writing or design problem is critical and fragile and must be allowed to proceed in a vague, contradictory, and incomplete form for as long as necessary". While it may not always be necessary to be vague, contradictory and incomplete, the essence of Conklin and Begeman's argument is that users should not be constrained into describing aspects of a design or problem solution as being values of one particular type, unless they feel this is appropriate. Such constraints should not be placed upon users of LD.

It remains unclear whether or not assigning types to relationships is sufficient to enable developers to determine the action to take in response to some change in the design. Further work and investigation is required to answer this question. If such types can be identified, they must support consistent interpretation during all stages of the design process and during all manner of design change.

Another factor to consider when interpreting relationships and determining the action to take in response to some change is the *time* at which the type is assigned, in terms of the completion of the particular design problem that users are working on. The time at which relationships are created and inspected can be useful in generating a meaningful interpretation of a relationship. While working on a particular design problem most of the relationships related to that problem could be interpreted validly as design policies, since the relationships suggest that that one part of the relationship (the context for example) should have a particular influence over the other part of the relationship (the design for example). The relationship can be viewed as a policy, since it suggests that if either the context or the design change then the associated part should be updated appropriately with respect to the relationship. Thus the relationship is setting an agenda for designers to adhere to.

In contrast, if a relationship is created or inspected after completion of the design problem, then the relationship can be viewed as part of the historical record of the design development process. The relationship relates two parts of say, the context and the design, and provides reasons for the relationship. It describes how part of the context was used to influence part of the design. This raises another set of issues, namely those of defining a design problem (is it the whole design project, or component parts of the design project?), when a design problem has been solved and when it still remains a problem (how does the interaction between component parts affect the definition of a solved and unsolved design problem).

In order to choose between using relationships as design policies to define an agenda for designers for future development and maintenance of a design or as a historical record of the design process, users have to decide what the likely uses of the set of relationships will be. If they are likely to be used to maintain a system, then the relationships should perhaps be considered as design policies. If the relationships are to be used as a historical record of the design process, for example in order to learn about the design process to improve the process for future projects, then the relationships should perhaps be viewed as a historical record.

It is clear from this discussion that further work is required to investigate the different usage of relationships and to determine if the distinction between relationships as design policies and relationships as a historical record of the design process is appropriate. If such a distinction is appropriate, the effect of time on the relationships should be investigated. Does the length of time that a relationship has existed have an effect on the way that the relationship is interpreted? For example, when does a policy relationship become a historical relationship? In general, what effect does time have on relationships, whether they are policy relationships, historical relationships or any other type? The length of time since the last update of either the source or destination of a relationship may have an effect on the relationship.

7.2.5 Supporting different analyses over the relationships

One evaluation participant suggested that the relationships between context and design could be used to perform different analyses over the relationships. He suggested that it would be useful to be able to isolate features of a design and to see how these features were influenced by the context. In this way, different designs could be compared in terms of individual components. This would allow designers to isolate the well designed features of a design and incorporate these in other similar designs. Further investigations are required to identify other analyses that should be supported by LD. Such investigations could be carried out via focus groups but it would be preferable to have real designers use the tool and suggest the analyses they think would be useful in their work. Asking designers to use the tool gives them a deeper understanding of the tool and its capabilities than simply describing the tool and generating feedback in a focus group.

7.2.6 Implementing a multi–user version of LD

In order that a design team make successful use of LD, a multi–user version must be made available, so that a group of designers can inspect and update descriptions of the context and design for their particular project simultaneously. Currently, LD is single–user only, meaning that a group of designers have to take turns to inspect and update descriptions of the context and design. This can interfere with the work of a design team. Often, design teams will separate a design project into a number of smaller, more manageable projects. The group is split up into small sub–groups, with each sub–group tackling one of the smaller projects. Clearly it would be disruptive and inefficient to force one group to wait until another group is finished using LD, so that they can update the descriptions of context and design to reflect the portion of the design project that they have been working on. What is required is a version of the tool that allows multiple groups to access and update the descriptions of context and design simultaneously and securely, taking into account standard transaction management issues such as serializibility and deadlock control (see Korth and Silberschatz, 1988 Chapter 11, for a detailed discussion of these issues).

However, as well as these technical issues that must be addressed, there are other, more fundamental issues that must also be addressed. The evaluations described in the previous chapter indicated that different people make different interpretations of individual contextual factors. In the evaluations, five people used LD separately to perform a number of tasks. Each person interpreted some of the contextual factors differently. Clearly, this will have consequences in a multi–user version of the tool since different members of the same design team could potentially have different ideas about what the context means. As a consequence each member of the team could have different views about the importance of the context and its potential influence over the design. It was clear that the explicit description of individual contextual factors highlighted these different interpretations, but further work is required to investigate how consensus can be reached about the interpretation of these factors. One of the ways this might be done would be to change the organisational structure and working methods of a group. Before working on the design, the group could be encouraged to discuss and record their interpretations and understanding of the contextual information. GENEVA, described in Appendix F, was implemented to support such a process.

There are other issues related to multi–user contextual design teams that must be addressed. At the CHI 96 Doctoral Consortium, I presented a short paper about LD. Most of the discussions after the presentation concerned issues relating to group use of LD. Concerns were raised about 'ownership' of the design and of the contextual information. Many participants felt that designers might be less motivated to make use of contextual information that has been collected by another group. Designers may be more inclined not to believe the data presented to them since they have invested no time or effort in collecting the data, and may prefer to believe and use their own experiences or intuitions, gained from similar projects they have worked on in the past.

The effects of these group issues on the design of a multi–user version of LD need to be investigated. The format of a group that will make good use of LD needs to be determined so that the tool can be designed to reflect the format of the group. Issues to be investigated here concern who should participate in discussions about the influence of context over a design (data collectors/designers/users) what levels of participation are appropriate and how this participation should be achieved.

7.2.7 Performing an 'ecologically valid' evaluation of extended LD

The work described so far aims to extend LD such that it can be used successfully in a real setting, by real designers. The case studies and evaluations described earlier in the thesis motivate these efforts by indicating that making explicit records of the relationships between context and design is worthwhile, but that the tool support for the maintenance and creation of such relationships in the form of LD needs improvement.

Once LD is extended to address the needs identified above, it should be evaluated in a valid setting. By doing so, we will have a better idea of the advantages and disadvantages of an industrial strength LD tool. The evaluations described in the previous chapter can best be described as formative evaluations of a prototype version of LD. They identifed most of the major problems with the current version of LD and at the same time indicated that the approach was worthwhile. They also provided valuable input into the design of an evaluation of LD.

Performing an ecologically valid evaluation of LD will evaluate the difference that LD makes to a design team's effective use of contextual information. The evaluations that were described in Chapter 5 evaluated a prototype version of LD and could not determine if the tool made a significant impact on designer's use of contextual information. Thus, an ecologically valid evaluation of the improved LD tool will determine if explicit relationships between context and design can increase the effectiveness with which designers use contextual information. It may also identify further ways in which effective use of contextual information can be achieved.

7.3 Chapter Summary

This chapter has described further work that has been suggested by the work presented in this thesis. The thesis has demonstrated that making the relationships between context and design explicit can be beneficial to designers. The proposals presented in this chapter outline research through which these benefits can be better exploited, and tool support improved.

8. References

Apple Computer, Inc. (1992) *Human Interface Guidelines: The Apple Desktop Interface*, Reading: Addison Wesley

Apple Computer, (1995). *The Open Doc Framework Resource Compiler Guide*, Apple Computer (guide available on OpenDoc Component Software Toolkit for Macintosh CD–ROM)

Baddeley, A. D (1990). *Human Memory: Theory and Practice*. Lawrence Erlbaum Associates, Hove.

Bellamy, R. K. E., (1996) Designing Educational Technology: Computer Mediated Change, in B Nardi (ed), *Context and Consciousness: Activity Theory and Human Computer Interaction*, MIT Press, pp 123 – 146

Bellotti, V. M. E. (1993) Integrating Theoreticians' and Practitioners' Perspectives with Design Rationale in *Proceedings of InterCHI'93: Human Factors in Computing Systems*, Amsterdam, Holland, ACM Press: New York, pp 101 – 106

Bevan, N. and MacLeod, M. (1994) Usability measurement in context, in *Behaviour and Information Technology*, Vol 13, Nos 1 & 2, pp 132 – 145

Bignell V. and J. Fortune (1991), *Understanding Systems Failures*, Manchester University Press

Bjerknes, G., Ehn, P. & Kyng, M. (1987*) Computers and Democracy: A Scandinavian Challenge*. Brookfield, VT: Gower

Bødker, S. (1996) Applying Activity Theory to Video Analysis: How to Make Sense of Video Data in HCI, in B Nardi (ed), *Context and Consciousness: Activity Theory and Human Computer Interaction*, MIT Press, pp 147 – 174

Bødker, S., (1991) *Through the Interface – A Human Activity Approach to User Interface Design,* Hillsdale, NJ: Lawrence Erlbaum

Britton, D. and A. Reyes (1994) Discovering Usability Improvements for Mosaic: Application of the Contextual Inquiry Techniqe with an Expert User, in *Proceedings of the*

Second International Workshop on the World Wide Web, http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/HCI/britton/britton_reyes.html

Buckingham Shum, S. (1995) Analysing the Usability of a Design Rationale Notation, in Moran, T.P. and Carroll, J. M. (eds) (1995), *Design Rationale: Concepts, Techniques and Use*, Lawrence Erlbaum Associates: NJ

Campbell, R. L. (1992) Will the real scenario please stand up? *SIGCHI Bulletin,* Vol 24, No. 2, pp 6 – 8

Card, S. K., T. P. Moran, and A. Newell (1983). *The Psychology of Human Computer Interaction.* Lawrence Erlbaum Associates, New Jersey.

Carroll, J. M. (1992) Making Use A Design Representation, *Technical Report RC 19287 (79613),* IBM Research Division, T. J. Watson Research Center, Yorktown Heights, NY

Carroll, J. M., S. R. Alpert, J. Karat, M. Van Deusen, and M. B. Rosson (1994) Raison d'Etre: Capturing Design History and Rationale in Multimedia Narratives, in B. Adelson, S. Dumais & J. Olson (eds), *Proceedings of CHI'94: Human Factors in Computing Systems*, ACM Press, pp 192-197

Carroll, J. M., (ed) (1995) *Scenario Based Design: Envisioning Work and Technology in System Development*, John Wiley and Sons, Inc.

Chalmers, A. F. *What is this thing called Science?* Open University Press (2nd edition)

Clarke, S. (1996) Putting Context Into Design, in *Human Factors in Computing Systems CHI'96 Conference Companion, Doctoral Consortium*, ACM Press: New York, pp 33 – 34

Clement, A. and van den Besselaar, P. (1993). A Retrospective Look at PD Projects, *Communications of the ACM,* April 1993, vol 36, no. 4, pp 29 - 39

Cockton, G., Clarke, S. & Gray, P. (1995) Theories of Context Influence the System Abstractions Used to Design Interactive Systems, in Kirby, M. A. R., Dix, A. J. and Finlay, J. E. (eds.) *People and Computers X (Proceedings of HCI'95),* Cambridge University Press, pp 387 – 405

Cockton, G., Clarke, S., Gray, P. & Johnson, C. (1996) Literate Development: Weaving Human Context into Design Specifications, in D Benyon & P. Palanque (eds.), *Critical Issues in User Interface Systems Engineering*, Springer Verlag, pp 227 – 248

Cole, M., and Engeström, Y. (1991) A cultural historical approach to distributed cognition. In G. Salomon, (ed) *Distributed Cognition*, Cambridge University Press, pp 1 – 47.

Conklin, J. & Begeman, M.L. (1988) gIBIS: A Hypertext Tool for Exploratory Policy Discussion. *ACM Transactions on Office Information Systems*, Vol 6, No. 4, pp 303 – 331

Curtis, B., Krasner, H. and Iscoe, N. (1988) A Field Study of the Software Design Process for Large Systems, in *Communications of the ACM*, Novermber 1988, Vol 31, No 11, pp 1268 – 1286

Czerwinski, M. (1996) Personal communication, Microsoft Corporation, Seattle, WA. July 1996

Diaper, D. (1996) Task Analysis and Structured Design Methods, in Johnson, C. and Jones, S. (eds) (1996*), User Centred Requirements Engineering Workshop: Integrating Methods from Software Engineering and Human Computer Interaction,* available as GIST Technical Report G96–1, Glasgow University, Computing Science Department

Dix, A., J. Finlay, G. Abowd and R. Beale (1993). *Human Computer Interaction*, Prentice Hall International, Hemel Hempstead.

Downs, E. Clare, P. and Coe, I. (1991) *Structured Systems Analysis and Design Method – Application and Context* (2nd edition), Prentice Hall

Ehn, P. (1989) *Work Oriented Design of Computer Artifacts.* Erlbaum, Hillsdale, NJ, 1989 (Cited in Bødker, 1996).

Erickson, T. (1995), Notes on Design Practice: Stories and Prototypes as Catalysts for Communication, in Carroll, J. M., (ed) (1995) *Scenario Based Design: Envisioning Work and Technology in System Development*, John Wiley and Sons, Inc., pp 37 – 58

Eva, M. (1994) *SSADM Version 4: A User's Guide* (2nd edition), McGraw Hill

Floyd, C., Mehl, W. M., Reisin, F. M., Schmidt, G & Wolf, G. (1989) Out of Scandinavia: Alternative approaches to software design and system development, *Human Computer Interaction*, Vol 4, No. 4, pp 253 – 350

Friedman, B and P. H. Kahn Jr (1994), Educating Computer Scientists: Linking the Social and the Technical, *Communications of the ACM*, January 1994, Vol 37, No. 1, pp 65 – 70

Gould, J. D., Boies, S. J., Levy, S., Richards, J. T. and Schoonard, J. (1987) The 1984 Olympic Message System: A Test of Behavioral Principles of System Design, *in Communications of the ACM*, September 1987, Vol 30, No. 9, pp 758 – 769

Greenbaum, J., and Kyng, M. (1992) *Design at Work* Hillsdale, NJ: Lawrence Erlbaum

Gruhn, V. and Wolf, S. (1995) Software Process Improvement by Business Process Orientation, *Software Process – Improvement and Practice*, Pilot Issue, pp 49 – 56

Guindin, R. (1990) Designing the Design Process: Exploiting Opportunistic Thoughts, in *Human Computer Interaction*, Vol 5, pp 305 – 344

Hammond, N. V. (1991) Modelling user, system and design: Results of a scenario matrix exercise. In S. P. Robertson, G. M. Olson & J. S. Olson (Eds.) *CHI'91 Conference Proceedings: Human Factors in Computing Systems*, ACM Press, pp 377 – 380

Hartson, H. Rex, Antonio C. Siochi and Deborah Hix (1990), "The UAN: A User–Oriented Representation for Direct Manipulation Interface Designs," *in ACM Transactions on Information Systems*, 8(3), pp.181–203.

Herbert, D. M. (1993) *Architecural Study Drawings*, New York: Van Nostrand Rheinhold

Hoare, C. A. R. (1978). Communicating sequential processes. *Communications of the ACM*, Vol 21, pp 666–677

Holtzblatt, Karen and Hugh Beyer (1993), "Making Customer-Centred Design Work For Teams", *Communications of the ACM*, October 1993, Vol 36 No. 10, pp 93 - 103

Holtzblatt, Karen and Hugh Beyer (1996), Contextual Design: Principles and Practice, in *Field Methods for Software and System Design*, D Wixon & J Ramey (eds), NY, NY: John Wiley & Sons, Inc.

Hughes, J., V. King, T. Rodden and H. Andersen (1995), The Role of Ethnography in Interactive Systems Design, in *interactions,* Vol. 2, No. 2, pp 50 – 65

Jacobson, I. (1995) The Use–Case Construct in Object–Oriented Software Engineering, in Carroll, J. M., (ed) (1995) *Scenario Based Design: Envisioning Work and Technology in System Development*, John Wiley and Sons, Inc., pp 309 – 336
Johnson et. al (1984)

Johnson, C. (1996) Literate Specification: Using Design Rationale To Support Formal Methods In The Development of Human-Computer Interfaces, in *Human-Computer Interaction Journal*, Vol. 11, No. 4

Johnson, C. and Jones, S. (eds) (1996*), User Centred Requirements Engineering Workshop: Integrating Methods from Software Engineering and Human Computer Interaction*, available as GIST Technical Report G96–1, Glasgow University, Computing Science Department

Kaiya, H., Saeki, M and Ochimizu, K. (1995) Design of a Hyper Media Tool to support Requirements Elicitation Meetings, in Müller, H. A. and Norman R. J (Eds.) *Proceedings of the Seventh International Workshop on Computer Aided Software Engineering*, Toronto, Ontario, Canada, IEEE Computer Society Press, pp 250 – 259

Kaptelinin, V. (1996) Activity Theory: Implications for Human Computer Interaction, in B Nardi (ed), *Context and Consciousness: Activity Theory and Human Computer Interaction*, MIT Press, pp 45 – 68

Karsenty, L. (1996) An Empirical Evaluation of Design Rationale Documents, in *Human Factors in Computing Systems CHI'96 Conference Proceedings,* ACM Press: New York, pp 150 – 156

Kleimann, S. (1996) Response to "Using Contextual Inquiry", in *The Journal of Computer Documentation*, Vol. 20, No. 1, ACM Press: NY, pp 22 – 24

Kling, R. (ed) (1996) *Computerization and Controversy: Value, Conflicts and Social Choices*, Academic Press Inc.

Koehler, Saskia, (1995) *Support System for IT Applications*, MSc in Information Technology dissertation

Korth, Henry and Abraham Silberschatz, (1988) *Database System Concepts*, McGraw Hill Book Company

Kuutti, K. (1995) Work Processes: Scenarios as a Preliminary Activity, in Carroll, J. M., (ed) (1995) *Scenario Based Design: Envisioning Work and Technology in System Development, John Wiley and Sons, Inc.*, pp 19 – 36

Kuutti, K., (1996), Activity Theory as a Potential Framework for Human Computer Interaction Research, in B Nardi (ed), *Context and Consciousness: Activity Theory and Human Computer Interaction*, MIT Press, pp 17 – 44

Kyng, M. (1995) Creating Contexts for Design, in Carroll, J. M., (ed) (1995) *Scenario Based Design: Envisioning Work and Technology in System Development, John Wiley and Sons, Inc.*, pp 85 – 108

Landauer, T. (1995) *The Trouble With Computers*, MIT Press

Lee, J. (1990) SIBYL: A Tool for Managing Group Design Rationale. *CSCW'90, Proceedings of Computer Supported Cooperative Work*, Los Angeles, CA, ACM Press: NY, pp 79 – 92

Leont' ev, A. (1974) The problem of activity in psychology. *Soviet Psychology* Vol 13 No. 2 pp 4–33, cited in Nardi (1996)

MacLean A, and Diane McKerlie (1995), Design Space Analysis and Use–Representations, in Carroll, J. M., (ed) (1995) *Scenario Based Design: Envisioning Work and Technology in System Development*, John Wiley and Sons, Inc.

MacLean, A., Young, R. M., Bellotti, V. and Moran, T. (1991) Questions, Options and Criteria: Elements of Design Space Analysis. *Human Computer Interaction*, Vol 6, Nos. 3 & 4, pp 201 – 250

Maissel, Jonathan , Miles Macleod, Cathy Thomas, Ralph Renger, Richard Corcoran, Andrew Dillon, Martin Maguire and Marian Sweeney (1993), *Usability Context Analysis: A Practical Guide*, Version 3.0, National Physical Laboratory and HUSAT Research Institute, DITC, NPL, Teddington, TW11 0LW, UK.

Mateas, M., Salvador, T., Scholtz, J. & Sorensen, D. (1996), Engineering Ethnography in the Home, in *Human Factors in Computing Systems CHI'96 Conference Companion, Short Papers*, ACM Press: New York, pp 283 – 284

Microsoft (1992) *The Windows Interface: An Application Design Guide*. Microsoft Press

Moran, T. P. (1994) Introduction to This Special Issue on Context in Design, *Human Computer Interaction*, Volume 9, pp 1 – 2

Moran, T.P. and Carroll, J. M. (eds) (1995), *Design Rationale: Concepts, Techniques and Use*, Lawrence Erlbaum Associates: NJ

Morrison, R., Connor, R. C. H., Cutts, Q. I., Dunstan, V.S. & Kirby, G. N. C. (1995) Exploiting Persistent Linkage in Software Engineering Environments in *The Computer Journal*, Vol 38, No 1. 1995

Nardi, B. A. (1992), The use of scenarios in design, *SIGCHI Bulletin*, Vol 24, No. 4, pp 13 – 14

Nardi, B. (1995) Some Reflections on Scenarios, in Carroll, J. M., (ed) (1995) *Scenario Based Design: Envisioning Work and Technology in System Development, John Wiley and Sons, Inc.*, pp 387 – 399

Nardi, B. (ed) (1996), *Context and Consciousness: Activity Theory and Human Computer Interaction*, MIT Press

Nardi, B., (1996a), Some Reflections on the Application of Activity Theory, in B Nardi (ed), Context and Consciousness: Activity Theory and Human Computer Interaction, MIT Press, pp 235 – 246

Object Management Group, (1996). *The Common Object Request Broker: Architecture and Specification*. Available at ftp://ftp.omg.org/pub/docs/ptc/96-08-04.ps.gz

Orlikowski, W. J. (1992) Learning from Notes: Organizational Issues in Groupware Implementation. *CSCW'92: Proceedings of the Conference on Computer Supported Cooperative Work*, Toronto, Canada, ACM Press: New York, pp 362 – 369

Parker, H., Roast, C. & Siddiqi, J. (1996) Towards a Framework for Investigating Temporal Properties in Interaction, in Johnson, C. and Jones, S. (eds) (1996), *User*

Centred Requirements Engineering Workshop: Integrating Methods from Software Engineering and Human Computer Interaction, available as GIST Technical Report G96–1, Glasgow University, Computing Science Department

Payne, S. J. and T. R. G. Green (1989). Task--action grammar: the model and its developments, in *Task Analysis for Human Computer Interaction*, D Diaper (ed.) Chichester: Ellis Horwood

Potts, C., Takahashi, K. & Anton, A. I. (1994) Inquiry based requirements analysis, *IEEE Software* (March) pp 21 - 32

Powrie,S. E. and Siemieniuch, C. E. (1990), An Investigation of User Requirements for Broadband Communications in the Automotive Industry in *Human Computer Interaction – INTERACT '90*, D Diaper, D. Gilmore, G. Cockton and B. Shackel (eds), Elsevier Science Publishers BV, pp 233–238.

Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S. and Carey, T. (1994), *Human Computer Interaction*, Addison Wesley

Raven, M. E. and Flanders, A. (1996) Using Contextual Inquiry to Learn About Your Audiences, in *The Journal of Computer Documentation*, Vol. 20, No. 1, ACM Press: NY, pp 1 – 13

Reisig, W. (1986) *Petrinetze* (in German - cited in Gruhn and Wolf, 1995), Springer, Berlin, FRG

Robertson S. P. (1995) Generating Object–Oriented Design Representations via Scenario Queries, in Carroll, J. M., (ed) (1995) *Scenario Based Design: Envisioning Work and Technology in System Development*, John Wiley and Sons, Inc., pp 279 – 308

Rosson, M. B. and Carroll, J. M. (1995) Narrowing the Specification–Implementation Gap in Scenario Based Design, in Carroll, J. M., (ed) (1995) *Scenario Based Design: Envisioning Work and Technology in System Development*, John Wiley and Sons, Inc., pp 247 – 278

Rosson, M. B., Maass, S. and Kellogg, W. A. (1988) The Designer as User: Building Requirements for Design Tools from Design Practice, in *Communications of the ACM*, November 1988, Vol 31, No 11, pp 1288 – 1298

Sachs, P. (1995) Transforming Work: Collaboration, Learning, and Design, in *Communications of the ACM*, September 1995, Vol 38, No. 9, pp 36 – 44

Salomon, G. (1990) How the look affects the feel: Visual design and the creation of an information kiosk, in *Proceedings of the Human Factors Society 34th Annual Meeting*, Orlando, FL, October 8 - 12, Santa Monica, CA: Human Factors Society, pp 277 – 281

Salvador, T and Scholtz, J (1996) Systematic creativity: a methodology for integrating user, market and engineering requirements for product definition, design and usability testing, in Bass and Unger (eds), *Engineering for Human-Computer Interaction.* Chapman & Hall, pages 307-332

Salvador, T. and Mateas, M. (1996) Personal communication, Intel Corporation, Portland, OR. July 1996

Scharfstein, B. (1989) *The Dilemma of Context,* New York University Press

Schuler, D. and Namioka, A. (eds) (1993) *Participatory Design: Principles and Practices.* Erlbaum, Hillsdale, NJ.

Seely–Brown, J. and Duguid, P. (1994), Borderline Issues: Social & Material Aspects of Design, *Human Computer Interaction*, vol 9, no.1, pp 3 – 36

Simpson, M. (1996) A Commentary on "Using Contextual Inquiry", in *The Journal of Computer Documentation*, Vol. 20, No. 1, ACM Press, NY, pp 25 – 28

Smith, S. L. and Mosier, J. N. (1986), *Guidelines for Designing User Interface Software.* ESD-TR-86-278. Bedford, MA: MITRE Corporation.

Sommerville, I. (1992) *Software Engineering 4th Edition,* Wokingham, England: Addison Wesley

Spivey, J. M. (1988). *The Z Notation: A Reference Manual.* Prentice Hall International, Hemel Hempstead.

US General Accounting Office (1992). *Patriot Missile Defense: Software Problem Led to System Failure at Dhahran, Saudi Arabia* (GAO/IMTEC-92-26) Washington D.C van Aalst, J. W., Carey, T. T. and McKerlie, D. L. (1995) Design Space Analysis as "Training Wheels" in a Framework for Learning User Interface Design, *in Proceedings of CHI'95 Human Factors in Computing Systems*, Denver, CO. ACM Press: NY, pp 154 – 161

Vygotsky, L. S. (1925/1982) Consciousness as a problem in the psychology of behaviour. In *Collected Works: Questions of the Theory and History of Psychology.* Moscow: Pedagogika. cited in Nardi (1996)

Weaver, P. (1993) *Practical SSADM version 4 – A Complete Tutorial Guide*, Pitman

Whiteside, J., Bennett, J. & Holtzblatt, K. (1988) Usability engineering: our experience and evolution, in *Handbook of Human Computer Interaction*, Helander, M., (ed) Amsterdam: North–Holland

Williams, A. S. (1994). The OLE 2.0 object model. In *ACM OOPS Messenger, Addendum to the Proceedings of OOPSLA 1993,* pages 68-70. Published as ACM OOPS Messenger, Addendum to the Proceedings of OOPSLA 1993, volume 5, number 2.

Wirfs–Brock, R. (1995) Designing Objects and Their Interactions: A Brief Look at Responsibility Driven Design, in Carroll, J. M., (ed) (1995) *Scenario Based Design: Envisioning Work and Technology in System Development*, John Wiley and Sons, Inc., pp 337 – 360

Wixon, D. & M. E. Raven (1994), Contextual Inquiry: Grounding Your Design In Users Work, Tutorial Notes — *Conference on Human Factors In Computing Systems CHI '94,* Digital Equipment Corporation, MA, USA.

Wong, Y. Y (1992) Rough and ready prototypes: Lessons from graphic design, in *Human Factors in Computing Systems: CHI'92 Conference, Posters and Short Talks*, New York: ACM Press, pp 83–84

9. Appendix A Checklist of Contextual Factors

STAKEHOLDERS
- General
  - Multi or single users.
  - Frequency of use
  - Objectives
  - Task experience
  - Experience with the product or related products
  - Training received
  - Qualifications
  - Relevant input skills
  - Linguistic ability
  - Background knowledge
- Mental
  - Intelligence
  - Expertise
  - Individual characteristics
  - Problem representation and solution strategy
  - Cognitive Styles
  - Dogmatism
  - Intro/extro–version
  - Education
  - Motivations
  - Attitude
    - To the job and task
    - To the product
    - To information technology
    - To computers
    - To employing organisation.
  - Experience
    - Programming
    - Work
    - System
    - Computer
  - Representation techniques
  - Spatial ability
  - Antipathy
  - Knowledge
    - Of application domain
    - Of computer systems
  - General abilities
  - Anxiety
    - Computer
    - Trait
  - Attention span
- Physical
  - Visual ability
  - Age
  - Sex
  - Role
  - Work demands
  - Effector
    - Feet
    - Hands
    - Voice
  - Senses

- Auditory
  - Acuity
  - Biaural
  - Monaural
  - Tone perception
- Olfactory
- Tactual
- Vision
  - Accomodation
  - Acuity
  - Binocular
  - Colour perception
  - Convergence
  - Monocular

ACTIVITY

- Type
  - Communication
    - Type
      - Advise
      - Answer
      - Comprehend
      - Coordinate
      - Direct
      - Indicate
      - Inform
      - Instruct
      - Request
      - Supervise
      - Transmit
    - Attribute
      - Oral
      - Written
  - Meditation
    - Type
      - Information processing
      - Problem solving and decision making
      - Recall
    - Attribute
      - Complexity
      - Difficulty
  - Perceptual processing
    - Type
      - Searching for and receiving information
      - Identifying objects, actions, events
    - Attributes
      - Amount of labour required
      - Complexity
      - Degree of response chaining
      - Difficulty
      - Knowledge of results
      - Output
      - Pacing
      - Precision
      - Repetitiveness
      - Skill demands
      - Simultaneity of responses
      - Task autonomy

- General
    - Goals
    - Outputs
    - Side effects
    - Frequency
    - Duration
    - Flexibility
    - Physical and mental demands
    - Dependencies
    - Other activities
    - Safety
    - Criticality
    - Activity sequences
    - Information requirements per activity
- DOMAIN
    - General
        - Task frequency
        - Time to complete task
        - Vocabulary
        - Collaborative/non–collaborative work
        - Type
            - Allocation
            - Stuctural
            - Analytic
    - Area
        - Learning
            - Cognitive styles
            - Learning styles
        - Database manipulation
            - Querying
            - Validation
            - Elicitation
            - Conceptual representation
            - Query formualtion
            - Maintaining
            - Data definition
        - Reading
        - Decision Making
            - Characteristics of the knowledge base (structured/unstructured/well defined/ambiguous/qualitative/quantitative)
            - Source of knowledge (integration of knowedge from multiple experts)
        - Writing
        - Idea Processing.
- ENVIRONMENT
    - Security
        - User levels
        - Information access (readable/writable/executable etc.)
        - Access mechanisms
    - Organisation
        - Organisational policy
        - Organisational aims
        - Organisational culture
        - Organisational procedures
    - Physical
        - Type
            - Laboratory
            - Office

- Outer space
- Noise
  - Duration
  - Frequency
  - Intensity
  - Medium
  - Range
  - Spectrum
- Lighting
  - Fluorescent
  - Incandescent
  - Sunlight
- Thermal environment
- Layout
- Environmental stability
- Space and furniture
- User posture
- Health hazards
- Protective clothing and equipment
- Interruptibility
- Deliverables
- Context in which user will use application
  - Alone
  - With other software
  - With other hardware
  - Over a network
  - As part of a training class

Social
- Role allocation
- Sex
- Source of knowledge (integration of knowedge from multiple experts)
- Multi/single users
- User characteristics (varied etc.)
- Group working
- Assistance
- Interruptions
- Management structure
- Communications structure
- Culture

Attitudes and culture
- IT policy
- Standards
- Directives
- Expectations
- Industrial relations

Technological
- Technical set up.
- Number of commands/functions.
- Space (disk etc.)
- Hardware
- Software
- Reference materials

10. Appendix B Hypotheses pertaining to LD

The following hypotheses were developed while attempting to evaluate LD. The hypotheses relate to LD alone. They do not relate to the situation where an experienced designer makes use of LD. The hypotheses simply state what LD supports. It is up to individual designers how best to make use of LD. Hence the hypotheses are fairly straightforward.

Hypothesis 1: Users of LD can accurately identify those elements of contextual data that have influenced the design ideas represented in the LD tool;

Hypothesis 2: Users of LD can accurately identify those elements of contextual data that have not influenced the design ideas represented in LD.

Hypothesis 3: Users of LD can accurately identify those elements of contextual data that have been used to influence the design ideas more than once;

Hypothesis 4: Users of LD can accurately identify those elements of design ideas that have been influenced by the contextual data;

Hypothesis 5: Users of LD can accurately identify those elements of design ideas that have not been influenced by contextual data.

11. Appendix C Evaluation Materials

Participants were given one of two different sets of materials. The first set contained embedded links within the design and context specifications of the Olympic Messaging System. The other set contained identifal specifications except that the links were removed. Both sets of participants were also given a description of the OMS and a set of questions that they were asked to answer about the OMS. These questions were use to ensure that each participant had the same understanding of OMS. These questions are shown first, followed by the materials containing the embedded links, then the materials without the links and finally, the questions that each participant was asked to answer using the materials.

A description of the Olympic Messaging System

The Olympic Messaging System (OMS) was a system designed and implemented by IBM for the 1984 Los Angeles Olympic Games. It allowed Olympians (athletes) to send messages to other Olympians and to receive messages from other Olympians and friends and family from all over the world.

The OMS used a telephone network to send and receive messages. 25 kiosks from which Olympians could use the OMS were placed around the Olympic Village. Each kiosk contained a standard push button telephone connected to the OMS, a TV screen upon which a video of how to use the OMS was played and a written instruction guide. The video screen also showed a list of Olympians who had new messages waiting to be received plus any news items of interest for Olympians in general.

Olympians accessed the OMS by picking up the telephone and following the pre–recorded prompts that were spoken to them over the phone. The prompts told them which buttons on the telephone to press to record a message, which buttons to press to hear a message, how to identify an Olympian to send a message to, how to identify themselves and how to enter their password. Twelve buttons beside the telephone allowed the Olympian to select one of twelve different languages that they would like to hear the spoken instructions in.

To leave a message for an Olympian, the Olympian would follow the pre–recorded prompts and then speak the message into the telephone. Once complete, the OMS would send the recorded message to the other Olympian that the message was for. To hear a message, the Olympian could listen to the recording of the message by following the appropriae pre–recorded prompts.

Non Olympian users from all over the world could access the OMS by telephoning their own country's National Olympic Committe (NOC) who were based in Los Angeles for the duration of the Olympics. The NOC would access the OMS for non Olympian users so that they could leave messages for Olympians.

Questions

1. What was the Olympic Messaging System?
2. Who could Olympians leave messages for?
3. Who could Olympians receive messages from?
4. How did Olympians send messages?
5. How did Olympians receive messages?
6. How did non Olympians access the OMS?
7. How many languages did the OMS work in?

Context of use of OMS containing links to design issues

Level 1	Level 2	Level 3	Value	Refer to
STAKEHOLDERS	General	Multi or single users.	The system will be used by many people at one time	
		Frequency of use	Will vary depending on person. Some may not use it at all, others only rarely and others a lot.	
		Task experience	Most people will have some experience with using a phone	See Design Issues 6 & 8
		Experience with the product or related products	Most people will have very little experience with using voice	See Design Issues 6 & 8

		messaging systems	
	Training received	Training for non Olympians will be minimum	See Design Issues 3, 4, & 5
	Relevant input skills	Some Olympians and non Olympians will not have experience of a push button phone	See Design Issues 6 & 8
	Linguistic ability	Many Olympians and non Olympians will not speak the two official languages of the tournament, English and French	
Mental	Expertise	Many Olympians and non Olympians will have low levels of expertise with regards using telephones, computers and other technology	See Design Issues 6 & 8
	Motivations	Many Olympians and	See Design Issues 6 & 8

		non Olympians will have low levels of expertise with regards using telephones, computers and other technology	
	Attitude To the job and task .	Most people will be very motivated to use the system as Olympians like to receive 'Good luck' messages.	
	Computer Experience	Most Olympians and non Olympians will have very little experience with using computer systems	See Design Issues 6 & 8
	Knowledge of computer systems	Most Olympians and non Olympians will have very little knowledge about computer systems	See Design Issues 6 & 8
Physical	Hands	Users of the OMS will use their hands to pick up the phone, dial	

			numbers etc	
		Voice	They will use their voice to leave messages	
		Auditory	They will use their auditory senses to listen to messages and instructions	
ACTIVITY	Type	Communication	All users will use the OMS to communicate with other users	
	General	Duration	The length of time it takes to leave a message should be kept to a minimum so as to minimise the length of time some non Olympian users need to spend on a long distance call	See Design Issue 5
ENVIRONMENT	Security	Access mechanisms	Personal information is being stored so need to restrict access to those who messages are intended for	See Design Issue 11
	Physical	Located outside	OMS kiosks will be placed	

		outside in visible places around the Olympic Village	
	Noise - duration	Since the kiosks are outside it is likely that noise will be present for most of the day and some of the night.	
	Noise - frequency	Particularly before and after major events when athletes are going to or from venues.	
	Noise - intensity	Could be high as kiosks are in public places and there are many athletes.	
	Sunlight	Strong sunlight during the day	See Design Issue 9
	Thermal environment	Could be very hot during the day, reaching the 30s	See Design Issue 9
	Air quality	Can often be very smoggy in LA	

	User posture	Users will be standing up to use the OMS	
	Context in which user will use application	Olympians will use the OMS on their own, following spoken instructions Non olympians will interact with an operator who will operate the system for them	See Design Issue 9 See Design Issue 2

List of design issues with references to contextual information

Design Issues for the OMS

1. The OMS should work with both push button and dial telephones since some users may only have access to and experience of one type of phone.

2. Non Olympians would be able to phone someone in their own country's National Olympic Office in Los Angeles. This ensures that they will be able to speak to someone in their own language, who can operate the OMS for them.

3. When the National Olympic Office is not staffed, phone calls will be passed to a central group of telephone operators who will be able to take the call. This will allow all calls to be handled.

4. Non Olympian users will be given instruction on how to use OMS via small pamphlets or postcards which describe the steps that need to be taken to use the OMS. This option is chosen because it would be impossible to train all prospective users in using the system due to the number of users and their locations.

5. Trained operators will operate the OMS for non Olympians so as to minimize the cost of long distance calls.

6. Olympian and non Olympian users will hang up a call simply by replacing the receiver. Other options such as pressing the button '4' are rejected since they force the users to listen to and remember more instructions and may be unfamiliar to users with little experience of computers or voice messaging systems or similar technology.

7. Olympian users will be able to 'undo' any mistakes they may make while operating the OMS by pressing the * key on the telephone. This increases the amount of additional knowledge Olympians must remember but gives them a chance to recover from their mistakes.

8. The functionality offered by the OMS will be kept to a minimum since including more complex type of functions will be off putting for those users who have very little experience with voive messaging systems, computers etc. Plus, the more functionality offered by the system, the more prompts and instructions are required in order to tell people how to use the system.

9. Written instructions, video based material and spoken instructions will all be available to the Olympians at the OMS system. Classroom based instruction is infeasible due to the widespread layout of the Olympic Village and the difficulties Olympians may have getting to the class. Plus, since it could be hot in LA, giving Olympians instruction cards to carry around with them could lead to the cards disintegrating in hot, sweaty pockets. Glossy paper might suffer from glare in the sunlight.

10. If the user does not do anything within a specified length of time from the end of any spoken message or instruction, the system will 'time out' and reset to the normal 'unlogged' state. The time is specified from the end of the message since the same message in different languages may last different lengths of time. If the time out was specified from the start of the message, some users may get less time than others to figure out what to do before the system timed out.

11. Olympians will use the last three numbers on their Olympic Badge as their password. Olympians have too many other things on their minds to have to bother with remembering passwords.

Context of use of OMS without references to the design issues

Level 1	Level 2	Level 3	Value
STAKEHOLDERS	General	Multi or single users.	The system will be used by many people at one time
		Frequency of use	Will vary depending on person. Some may not use it at all, others only rarely and others a lot.
		Task experience	Most people will have some experience with using a phone
		Experience with the product or related products	Most people will have very little experience with using voice messaging systems
		Training received	Training for non Olympians will be minimum
		Relevant input skills	Some Olympians and non Olympians will not have experience of a push button phone
		Linguistic ability	Many Olympians and non Olympians will not speak the two official languages of the tournament,

		English and French
Mental	Expertise	Many Olympians and non Olympians will have low levels of expertise with regards using telephones, computers and other technology
	Motivations	Many Olympians and non Olympians will have low levels of expertise with regards using telephones, computers and other technology
	Attitude To the job and task .	Most people will be very motivated to use the system as Olympians like to receive 'Good luck' messages.
	Computer Experience	Most Olympians and non Olympians will have very little experience with using computer systems
	Knowledge of computer systems	Most Olympians and non Olympians will have very little knowledge about computer systems

	Physical	Hands	Users of the OMS will use their hands to pick up the phone, dial numbers etc
		Voice	They will use their voice to leave messages
		Auditory	They will use their auditory senses to listen to messages and instructions
ACTIVITY	Type	Communication	All users will use the OMS to communicate with other users
	General	Duration	The length of time it takes to leave a message should be kept to a minimum so as to minimise the length of time some non Olympian users need to spend on a long distance call
ENVIRONMENT	Security	Access mechanisms	Personal information is being stored so need to restrict access to those who messages are intended for
	Physical	Located outside	OMS kiosks will be placed outside in visible places around

		the Olympic Village
	Noise - duration	Since the kiosks are outside it is likely that noise will be present for most of the day and some of the night.
	Noise - frequency	Particularly before and after major events when athletes are going to or from venues.
	Noise - intensity	Could be high as kiosks are in public places and there are many athletes.
	Sunlight	Strong sunlight during the day
	Thermal environment	Could be very hot during the day, reaching the 30s
	Air quality	Can often be very smoggy in LA
	User posture	Users will be standing up to use the OMS
	Context in which user will use application	Olympians will use the OMS on their own, following spoken instructions

Non olympians will interact with an operator who will operate the system for them

Design Issues for the OMS without references to contextual information

1. The OMS should work with both push button and dial telephones.

2. Non Olympians will be able to phone someone in their own country's National Olympic Office in Los Angeles who will operate the OMS for them.

3. When the National Olympic Office is not staffed, phone calls will be passed to a central group of telephone operators who will be able to take the call.

4. Non Olympian users will be given instruction on how to use OMS via small pamphlets or postcards which describe the steps that need to be taken to use the OMS.

5. Trained operators will operate the OMS for non Olympians.

6. Olympian and non Olympian users will hang up a call simply by replacing the receiver.

7. Olympian users will be able to 'undo' any mistakes they may make while operating the OMS by pressing the * key on the telephone.

8. The functionality offered by the OMS will be kept to a minimum.

9. Written instructions, video based material and spoken instructions will all be available to the Olympians at the OMS system.

10. If the user does not do anything within a specified length of time from the end of any spoken message or instruction, the system will 'time out' and reset to the normal 'unlogged' state.

11. Olympians will use the last three numbers on their Olympic Badge as their password.

List of questions given to both sets of participants

1. Make a list of the contextual information described in the context of use of the OMS document which, as far as you can tell, have not been used in the design of the OMS. Please make a note of the time taken to compile the list for question 2. When compiling the list, use the names of the items as described in the level 3 column of the context of use of the OMS document. From this list, propose a design change that in your opinion will improve the OMS.
2. How long did it take you to compile the list in question 1? Please give an answer in minutes. If you were unable to compile a list, please answer N/A.
3. Please rate the task of compiling the list in question 1 on the following dimensions.

	extremely	quite	slightly	neutral	slightly	quite	extremely	
easy								difficult
simple								complicated

4. Make a list of the design features that, as far as you can tell, have not been influenced by the context of use as described in the context of use of the OMS document. Please make a note of the time taken to compile the list for question 5. When compiling your list, just list the numbers of each appropriate design feature. Can you think of any contextual factors that these design features should be influenced by? If so, please list them. The factors in your list need not be taken only from those factors described in the context of use of the OMS document.
5. How long did it take you to compile the list in question 4? Please give an answer in minutes. If you were unable to compile a list, please answer N/A.
6. Please rate the task of compiling the list in question 4 on the following dimensions.

	extremely	quite	slightly	neutral	slightly	quite	extremely	
easy								difficult
simple								complicated

7. How much of the contextual information described in the context of use of the OMS document do you think has been used to influence the design? Please answer on the following scale of 1 to 5 by circling the appropriate answer. In the scale, 1 represents the answer 'Between 0 and 19% of all of the contextual information has been used to influence the design', 2 represents the answer 'Between 20 and 39% of all of the contextual information has been used to influence the design' and so on.

1	2	3	4	5
0 - 19%	20 - 39%	40 - 59%	60 - 79%	80 - 100%

8. How many of the design features do you think have been influenced by the contextual factors described in the context of use of the OMS document? Please answer on the following scale of 1 to 5 by circling the appropriate answer. In the scale, 1 represents the answer 'Between 0 and 19% of the design features have been influenced by the contextual factors described in the context of use of the OMS document', 2 represents the answer 'Between 20 and 39% of the design features have been influenced by the contextual factors described in the context of use of the OMS document' and so on.

1	2	3	4	5
0 - 19%	20 - 39%	40 - 59%	60 - 79%	80 - 100%

12. Appendix D Log Files

Log files are shown for each participant who took part in the third evaluation described in Chapter 5.

Participant 1

```
Show unused contextual information menu option chosen 2:08:26 pm
Moved to group Communication in contextual focus view 2:27:20 pm
Moved to group Communication in contextual focus view 2:27:20 pm
Moved to group Stakeholders in contextual focus view 2:27:22 pm
Moved to group Stakeholders in contextual focus view 2:27:22 pm
Moved to group Environment in contextual focus view 2:27:25 pm
Moved to group Environment in contextual focus view 2:27:25 pm
Moved to group Physical in contextual focus view 2:27:28 pm
Moved to group Physical in contextual focus view 2:27:28 pm
Moved to group Security in contextual focus view 2:27:31 pm
Moved to group Security in contextual focus view 2:27:31 pm
Moved to factor Information access in contextual focus view 2:27:33 pm
Moved to factor Information access in contextual focus view 2:27:33 pm
Moved to factor Access mechanisms in contextual focus view 2:28:14 pm
Moved to factor Access mechanisms in contextual focus view 2:28:14 pm
Moved to factor Information access in contextual focus view 2:28:20 pm
Moved to factor Information access in contextual focus view 2:28:20 pm
Moved to factor Access mechanisms in contextual focus view 2:28:22 pm
Moved to factor Access mechanisms in contextual focus view 2:28:22 pm
Moved to factor Information access in contextual focus view 2:28:31 pm
Moved to factor Information access in contextual focus view 2:28:31 pm
Moved to factor Access mechanisms in contextual focus view 2:28:32 pm
Moved to factor Access mechanisms in contextual focus view 2:28:32 pm
Relationship paths from current part of current document menu option
chosen for document Contextual Focus and part Information load 2:31:51
pm
Contextual Focus:Information load
Objects reachable from Contextual Focus:Information load
          Olympic QOCs:Users recall abilities
          Objects reachable from Olympic QOCs:Users recall abilities
```

=====================================

2:31:51 pm

Show number of relationships menu option chosen 2:33:18 pm

inspecting new class Olympian 2:34:12 pm

inspecting new class Olympic Session 2:34:13 pm

inspecting new class Olympic Session 2:34:30 pm

Now inspecting command Connect 2:34:33 pm

Now inspecting command IdentifyOlympian 2:34:47 pm

Now inspecting command Connect 2:35:00 pm

Now inspecting command IdentifyOlympian 2:35:02 pm

Opened new link information needed as password 2:38:09 pm

New relationship information needed as password created 2:38:10 pm

inspecting new class Olympian 2:44:04 pm

Now inspecting attribute Password 2:44:09 pm

Participant 2

Show unused contextual information menu option chosen 4:29:22 pm

In Olympic QOCs new question selected What kinds of telephones to use? 4:42:33 pm

In Olympic QOCs new question selected How should non-Olympians interact with OMS? 4:42:42 pm

In Olympic QOCs new question selected How should Olympians recover from errors? 4:42:50 pm

In Olympic QOCs new question selected How will users make notes about messages? 4:43:03 pm

In Olympic QOCs new question selected How should instructions be provided to Olympians? 4:43:10 pm

In Olympic QOCs new question selected How will messages be accessed? 4:43:44 pm

Moved to factor Training received in contextual focus view 4:54:34 pm

Moved to factor Training received in contextual focus view 4:54:34 pm

Opened new link Using the available technology to constrain the technology that OMS is based on 5:02:20 pm

Opened new link Users experience constrains the technology used 5:03:28 pm

Show number of relationships menu option chosen 5:17:34 pm

Participant 3

Moved to group General in contextual focus view 8:32:02 pm
Moved to group General in contextual focus view 8:32:02 pm
Moved to group Stakeholders in contextual focus view 8:32:03 pm
Moved to group Stakeholders in contextual focus view 8:32:04 pm
Show number of relationships menu option chosen 8:32:28 pm
Show unused contextual information menu option chosen 8:32:55 pm
Opened new link Sending a message 8:35:05 pm
Opened new link Using the available technology to constrain the technology that OMS is based on 8:35:36 pm
Opened new link Users experience constrains the technology used 8:35:38 pm
Opened new link Working out which password to use 8:35:39 pm
Opened new link Relating work of an Olympian to type of password 8:35:41 pm
Opened new link Information load determines the appropriate password 8:35:42 pm
Opened new link Deciding on the appropriate type of telephone 8:35:43 pm
Opened new link Deciding how non Olympians should interact with the system 8:35:44 pm
Opened new link Processing information linearly 8:35:46 pm
Opened new link Experience with technology constrains the number of functions offered by the system 8:35:49 pm
Opened new link Using pen and paper to make notes 8:35:50 pm
Opened new link Determining which languages to support in the system 8:35:51 pm
Opened new link Different languages take different times 8:35:53 pm
Opened new link The environment constrains the kinds of instructions provided 8:35:58 pm
Opened new link Entering the password 8:36:00 pm
Opened new link Identifying an Olympian 8:36:02 pm
Opened new link Using the operator 8:36:04 pm
Opened new link information needed as password 8:36:08 pm
Opened new link Sending a message 8:36:09 pm
New scenario selected User sending a message 8:39:33 pm
New scenario selected User sending a message 8:39:37 pm

New anchor knowledge of numbers added to scenario User sending a message 8:42:07 pm

New anchor selected knowledge of numbers 8:42:11 pm

New anchor selected Sending a message 8:42:35 pm

Opened new link previous knowledge of message systems 8:45:10 pm

New relationship previous knowledge of message systems created 8:45:11 pm

Opened new link Experience with technology constrains the number of functions offered by the system 8:46:58 pm

Show number of relationships menu option chosen 8:47:43 pm

In Olympic QOCs new question selected How will messages be accessed? 8:49:25 pm

In Olympic QOCs new question selected What kinds of telephones to use? 8:50:09 pm

In Olympic QOCs new question selected How should Olympians recover from errors? 8:50:14 pm

In Olympic QOCs new question selected How will messages be accessed? 8:50:30 pm

Show number of relationships menu option chosen 8:52:58 pm

In Olympic QOCs new question selected How many commands at a time? 8:53:18 pm

In Olympic QOCs new question selected What password to use? 8:53:25 pm

In Olympic QOCs new question selected How should Olympians recover from errors? 8:53:30 pm

In Olympic QOCs new question selected How should instructions be provided to Olympians? 8:53:41 pm

Moved to group General in contextual focus view 8:55:28 pm

Moved to group General in contextual focus view 8:55:28 pm

Moved to factor Multi or single users in contextual focus view 8:55:29 pm

Moved to factor Multi or single users in contextual focus view 8:55:29 pm

Moved to factor Experience with (related) product in contextual focus view 8:55:31 pm

Moved to factor Experience with (related) product in contextual focus view 8:55:31 pm

Looking for links that Experience with (related) product is involved in 8:55:47 pm

Have found involved links. Displaying involved links and affected objects 8:55:47 pm

Involved links are 8:55:48 pm

previous knowledge of message systems 8:55:48 pm

Users experience constrains the technology used 8:55:48 pm

Reachable objects are 8:56:14 pm

Change made in Contextual Focus:Task Experience

Following link previous knowledge of message systems

Relationship path starting from Olympic scenarios:Sending a message

Olympic scenarios:Sending a message

Objects reachable from Olympic scenarios:Sending a message

Olympic NUF:Sending a message

Objects reachable from Olympic NUF:Sending a message

=====================================

=====================================

Change made in Contextual Focus:Description of experience with technology

Following link Users experience constrains the technology used

Relationship path starting from Olympic QOCs:Users experience criteria

Olympic QOCs:Users experience criteria

Objects reachable from Olympic QOCs:Users experience criteria

Olympic QOCs:Keeping commands to a minimum

Objects reachable from Olympic QOCs:Keeping commands to a minimum

Olympic QOCs:Using pen and paper to make notes

Objects reachable from Olympic QOCs:Using pen and paper to make notes

=====================================

=====================================

=====================================

8:56:14 pm

Updated Experience with (related) product in contextual focus view 9:03:07 pm

Looking for links that Experience with (related) product is involved in 9:03:37 pm

Have found involved links. Displaying involved links and affected objects 9:03:38 pm

Involved links are 9:03:38 pm

previous knowledge of message systems 9:03:38 pm

Users experience constrains the technology used 9:03:38 pm

Reachable objects are 9:03:42 pm

Change made in Contextual Focus:Task Experience

Following link previous knowledge of message systems

Relationship path starting from Olympic scenarios:Sending a message

Olympic scenarios:Sending a message

Objects reachable from Olympic scenarios:Sending a message

Olympic NUF:Sending a message

Objects reachable from Olympic NUF:Sending a message

====================================

====================================

Change made in Contextual Focus:Description of experience with technology

Following link Users experience constrains the technology used

Relationship path starting from Olympic QOCs:Users experience criteria

Olympic QOCs:Users experience criteria

Objects reachable from Olympic QOCs:Users experience criteria

Olympic QOCs:Keeping commands to a minimum

Objects reachable from Olympic QOCs:Keeping commands to a minimum

Olympic QOCs:Using pen and paper to make notes

Objects reachable from Olympic QOCs:Using pen and paper to make notes

====================================

====================================

====================================

9:03:42 pm

Updated Experience with (related) product in contextual focus view 9:03:46 pm

Participant 4

Show unused contextual information menu option chosen 10:23:37 am

Moved to group Communication in contextual focus view 10:26:01 am
Moved to group Communication in contextual focus view 10:26:01 am
Moved to group Stakeholders in contextual focus view 10:26:04 am
Moved to group Stakeholders in contextual focus view 10:26:04 am
Moved to group Stakeholders in contextual focus view 10:26:10 am
Moved to group Stakeholders in contextual focus view 10:26:10 am
Moved to group General in contextual focus view 10:26:13 am
Moved to group General in contextual focus view 10:26:13 am
Moved to factor Multi or single users in contextual focus view 10:26:18 am
Moved to factor Multi or single users in contextual focus view 10:26:19 am
Moved to factor Task experience in contextual focus view 10:26:22 am
Moved to factor Task experience in contextual focus view 10:26:22 am
New part Task Experience of document Contextual Focus created 10:29:18 am
Moved to factor Experience with (related) product in contextual focus view 10:30:03 am
Moved to factor Experience with (related) product in contextual focus view 10:30:03 am
Moved to factor Skill in contextual focus view 10:30:13 am
Moved to factor Skill in contextual focus view 10:30:13 am
Opened new link Experience related to method of system use 10:34:11 am
New relationship Experience related to method of system use created 10:34:14 am
Show unused contextual information menu option chosen 10:34:43 am
Moved to factor Experience with (related) product in contextual focus view 10:52:24 am
Moved to factor Experience with (related) product in contextual focus view 10:52:24 am
Show number of relationships menu option chosen 10:53:22 am
Relationship paths from current part of current document menu option chosen for document Contextual Focus and part Description of experience with technology 10:53:46 am
Show involved relationships menu option chosen for Contextual Focus 10:54:41 am
Relationship paths from current part of current document menu option chosen for document Contextual Focus and part Description of experience with technology 10:55:44 am

Relationship paths from current part of current document menu option chosen for document Contextual Focus and part Description of experience with technology 10:56:37 am
In Olympic QOCs new question selected How will messages be accessed? 11:00:05 am
In Olympic QOCs new option selected By entering a password for How will messages be accessed? 11:00:12 am
In Olympic QOCs new option selected Voice recognition for How will messages be accessed? 11:00:16 am
In Olympic QOCs new option selected By smart (swipe) card for How will messages be accessed? 11:00:17 am
In Olympic QOCs new option selected By entering a password for How will messages be accessed? 11:00:22 am
In Olympic QOCs new question selected How many commands at a time? 11:00:27 am
In Olympic QOCs new question selected How will users make notes about messages? 11:00:37 am

Participant 5

Opened new link Relating work of an Olympian to type of password 4:16:49 pm
New scenario selected Parent leaving a message 4:22:31 pm
New scenario selected User listening to message 4:22:32 pm
New scenario selected User sending a message 4:22:33 pm
New anchor selected Sending a message 4:23:25 pm
New scenario selected User listening to message 4:24:00 pm
New scenario selected Parent leaving a message 4:24:01 pm
New anchor selected Using the operator 4:24:03 pm
New scenario selected User listening to message 4:24:12 pm
New anchor selected Identifying an Olympian 4:24:14 pm
Moved to group Physical in contextual focus view 4:24:40 pm
Moved to group Physical in contextual focus view 4:24:40 pm
Moved to group Stakeholders in contextual focus view 4:24:41 pm
Moved to group Stakeholders in contextual focus view 4:24:41 pm
Moved to group Domain in contextual focus view 4:24:52 pm
Moved to group Domain in contextual focus view 4:24:52 pm

Moved to group General in contextual focus view 4:24:56 pm
Moved to group General in contextual focus view 4:24:56 pm
Moved to factor Time to complete task in contextual focus view 4:24:59 pm
Moved to factor Time to complete task in contextual focus view 4:24:59 pm
Moved to group General in contextual focus view 4:25:02 pm
Moved to group General in contextual focus view 4:25:02 pm
Moved to group Stakeholders in contextual focus view 4:25:03 pm
Moved to group Stakeholders in contextual focus view 4:25:03 pm
Moved to group Environment in contextual focus view 4:25:04 pm
Moved to group Environment in contextual focus view 4:25:04 pm
Moved to group Physical in contextual focus view 4:25:06 pm
Moved to group Physical in contextual focus view 4:25:06 pm
Moved to factor Type - outer space in contextual focus view 4:25:06 pm
Moved to factor Type - outer space in contextual focus view 4:25:07 pm
Moved to group Physical in contextual focus view 4:25:09 pm
Moved to group Physical in contextual focus view 4:25:09 pm
Moved to group Stakeholders in contextual focus view 4:25:10 pm
Moved to group Stakeholders in contextual focus view 4:25:10 pm
Moved to group Activity in contextual focus view 4:25:31 pm
Moved to group Activity in contextual focus view 4:25:31 pm
Moved to group Domain in contextual focus view 4:25:32 pm
Moved to group Domain in contextual focus view 4:25:32 pm
Moved to group Environment in contextual focus view 4:25:33 pm
Moved to group Environment in contextual focus view 4:25:33 pm
In Olympic QOCs new question selected How should non-Olympians interact with OMS? 4:25:52 pm
In Olympic QOCs new option selected Issuing commands via buttons on phone for How should non-Olympians interact with OMS? 4:25:54 pm
In Olympic QOCs new option selected Via an operator for How should non-Olympians interact with OMS? 4:26:10 pm
In Olympic QOCs new question selected How will messages be accessed? 4:26:29 pm
In Olympic QOCs new question selected What kinds of telephones to use? 4:26:40 pm
In Olympic QOCs new question selected How should non-Olympians interact with OMS? 4:27:03 pm

inspecting new class Olympic Session 4:27:21 pm
inspecting new class Olympian 4:27:22 pm
inspecting new class Message 4:27:23 pm
inspecting new class Olympic Session 4:27:24 pm
Now inspecting command DisplayNewMessages 4:27:37 pm
Now inspecting command IdentifyOlympian 4:27:58 pm
inspecting new class Olympian 4:28:05 pm
Now inspecting command LeaveMessage 4:28:11 pm
Moved to group Stakeholders in contextual focus view 4:32:07 pm
Moved to group Stakeholders in contextual focus view 4:32:08 pm
Moved to group General in contextual focus view 4:32:11 pm
Moved to group General in contextual focus view 4:32:11 pm
Show unused contextual information menu option chosen 4:33:15 pm
Moved to factor Multi or single users in contextual focus view 4:43:24 pm
Moved to factor Multi or single users in contextual focus view 4:43:25 pm
Moved to factor Experience with (related) product in contextual focus view 4:43:27 pm
Moved to factor Experience with (related) product in contextual focus view 4:43:27 pm
Moved to factor Task experience in contextual focus view 4:43:28 pm
Moved to factor Task experience in contextual focus view 4:43:28 pm
Moved to factor Multi or single users in contextual focus view 4:47:21 pm
Moved to factor Multi or single users in contextual focus view 4:47:22 pm
Opened new link Using the available technology to constrain the technology that CMS is based on 4:48:17 pm
Moved to group General in contextual focus view 4:48:50 pm
Moved to group General in contextual focus view 4:48:50 pm
Moved to group General in contextual focus view 4:48:53 pm
Moved to group General in contextual focus view 4:48:53 pm
Moved to group Physical in contextual focus view 4:50:10 pm
Moved to group Physical in contextual focus view 4:50:10 pm
Moved to group General in contextual focus view 4:50:11 pm
Moved to group General in contextual focus view 4:50:11 pm
Moved to factor Multi or single users in contextual focus view 4:50:12 pm

Moved to factor Multi or single users in contextual focus view 4:50:13 pm
Moved to group General in contextual focus view 4:50:17 pm
Moved to group General in contextual focus view 4:50:17 pm
Moved to group Stakeholders in contextual focus view 4:50:17 pm
Moved to group Stakeholders in contextual focus view 4:50:18 pm
Moved to group Environment in contextual focus view 4:50:19 pm
Moved to group Environment in contextual focus view 4:50:19 pm
Moved to group Physical in contextual focus view 4:50:21 pm
Moved to group Physical in contextual focus view 4:50:21 pm
Moved to group Technological in contextual focus view 4:50:23 pm
Moved to group Technological in contextual focus view 4:50:23 pm
Moved to factor Technical set up in contextual focus view 4:50:24 pm
Moved to factor Technical set up in contextual focus view 4:50:24 pm
Moved to group Physical in contextual focus view 4:51:45 pm
Moved to group Physical in contextual focus view 4:51:45 pm
Moved to group Stakeholders in contextual focus view 4:52:03 pm
Moved to group Stakeholders in contextual focus view 4:52:03 pm
Moved to group Environment in contextual focus view 4:52:26 pm
Moved to group Environment in contextual focus view 4:52:26 pm
Moved to group Physical in contextual focus view 4:52:29 pm
Moved to group Physical in contextual focus view 4:52:29 pm
Moved to group Stakeholders in contextual focus view 4:52:31 pm
Moved to group Stakeholders in contextual focus view 4:52:31 pm
Moved to group General in contextual focus view 4:52:35 pm
Moved to group General in contextual focus view 4:52:35 pm
Moved to group Stakeholders in contextual focus view 4:52:44 pm
Moved to group Stakeholders in contextual focus view 4:52:44 pm
Opened new link Experience with technology constrains the number of functions offered by the system 4:56:05 pm
Moved to group General in contextual focus view 5:02:00 pm
Moved to group General in contextual focus view 5:02:01 pm
Moved to factor Multi or single users in contextual focus view 5:02:02 pm
Moved to factor Multi or single users in contextual focus view 5:02:02 pm
Moved to group General in contextual focus view 5:02:03 pm
Moved to group General in contextual focus view 5:02:03 pm

Moved to factor Multi or single users in contextual focus view 5:02:05 pm

Moved to factor Multi or single users in contextual focus view 5:02:05 pm

Moved to factor Experience with (related) product in contextual focus view 5:02:08 pm

Moved to factor Experience with (related) product in contextual focus view 5:02:08 pm

Moved to group General in contextual focus view 5:03:10 pm

Moved to group General in contextual focus view 5:03:10 pm

Moved to group Stakeholders in contextual focus view 5:03:12 pm

Moved to group Stakeholders in contextual focus view 5:03:12 pm

Moved to group Activity in contextual focus view 5:03:16 pm

Moved to group Activity in contextual focus view 5:03:16 pm

Moved to group Communication in contextual focus view 5:03:17 pm

Moved to group Communication in contextual focus view 5:03:17 pm

Moved to factor Answer in contextual focus view 5:03:22 pm

Moved to factor Answer in contextual focus view 5:03:22 pm

Moved to group Communication in contextual focus view 5:03:25 pm

Moved to group Communication in contextual focus view 5:03:25 pm

Moved to group General in contextual focus view 5:03:26 pm

Moved to group General in contextual focus view 5:03:27 pm

Moved to factor Duration in contextual focus view 5:03:28 pm

Moved to factor Duration in contextual focus view 5:03:28 pm

Moved to group Communication in contextual focus view 5:03:30 pm

Moved to group Communication in contextual focus view 5:03:30 pm

Moved to group Stakeholders in contextual focus view 5:03:31 pm

Moved to group Stakeholders in contextual focus view 5:03:31 pm

Moved to group Environment in contextual focus view 5:03:39 pm

Moved to group Environment in contextual focus view 5:03:39 pm

Moved to group Physical in contextual focus view 5:03:41 pm

Moved to group Physical in contextual focus view 5:03:41 pm

Moved to group Security in contextual focus view 5:03:43 pm

Moved to group Security in contextual focus view 5:03:43 pm

Moved to group Technological in contextual focus view 5:03:44 pm

Moved to group Technological in contextual focus view 5:03:44 pm

Moved to factor Technical set up in contextual focus view 5:03:45 pm

Moved to factor Technical set up in contextual focus view 5:03:45 pm

Moved to group Physical in contextual focus view 5:04:02 pm
Moved to group Physical in contextual focus view 5:04:03 pm
Moved to group Stakeholders in contextual focus view 5:04:09 pm
Moved to group Stakeholders in contextual focus view 5:04:09 pm
Moved to group Activity in contextual focus view 5:04:16 pm
Moved to group Activity in contextual focus view 5:04:16 pm
Moved to group Domain in contextual focus view 5:04:17 pm
Moved to group Domain in contextual focus view 5:04:17 pm
Moved to group General in contextual focus view 5:04:18 pm
Moved to group General in contextual focus view 5:04:18 pm
Moved to factor Time to complete task in contextual focus view 5:04:20 pm
Moved to factor Time to complete task in contextual focus view 5:04:20 pm
Moved to group General in contextual focus view 5:04:47 pm
Moved to group General in contextual focus view 5:04:47 pm
Moved to group Stakeholders in contextual focus view 5:04:48 pm
Moved to group Stakeholders in contextual focus view 5:04:48 pm
Moved to group General in contextual focus view 5:04:51 pm
Moved to group General in contextual focus view 5:04:51 pm
Moved to group Physical in contextual focus view 5:04:54 pm
Moved to group Physical in contextual focus view 5:04:54 pm
Moved to factor Effector - hands in contextual focus view 5:04:55 pm
Moved to factor Effector - hands in contextual focus view 5:04:55 pm
Moved to factor Work demands in contextual focus view 5:04:58 pm
Moved to factor Work demands in contextual focus view 5:04:58 pm
Moved to group General in contextual focus view 5:05:08 pm
Moved to group General in contextual focus view 5:05:08 pm
Moved to group Stakeholders in contextual focus view 5:05:14 pm
Moved to group Stakeholders in contextual focus view 5:05:14 pm
Moved to group Activity in contextual focus view 5:05:16 pm
Moved to group Activity in contextual focus view 5:05:16 pm
Moved to group Communication in contextual focus view 5:05:17 pm
Moved to group Communication in contextual focus view 5:05:17 pm
Moved to factor Answer in contextual focus view 5:05:21 pm
Moved to factor Answer in contextual focus view 5:05:21 pm
Moved to factor Comprehend in contextual focus view 5:05:23 pm
Moved to factor Comprehend in contextual focus view 5:05:23 pm

Moved to group Communication in contextual focus view 5:05:27 pm
Moved to group Communication in contextual focus view 5:05:27 pm
Moved to group Stakeholders in contextual focus view 5:05:28 pm
Moved to group Stakeholders in contextual focus view 5:05:28 pm
Moved to group Environment in contextual focus view 5:05:30 pm
Moved to group Environment in contextual focus view 5:05:30 pm
Moved to group Physical in contextual focus view 5:05:31 pm
Moved to group Physical in contextual focus view 5:05:31 pm
Moved to group Organisation in contextual focus view 5:05:34 pm
Moved to group Organisation in contextual focus view 5:05:34 pm
Moved to factor Organisational policy in contextual focus view 5:05:35 pm
Moved to factor Organisational policy in contextual focus view 5:05:35 pm
Moved to group Physical in contextual focus view 5:05:39 pm
Moved to group Physical in contextual focus view 5:05:40 pm
In Olympic QOCs new question selected How many commands at a time? 5:06:17 pm
In Olympic QOCs new option selected Minimum for How many commands at a time? 5:06:21 pm
Opened new link Users experience constrains the technology used 5:07:17 pm
Relationship paths from current part of current document menu option chosen for document Contextual Focus and part Description of experience with technology 5:07:47 pm
In Olympic QOCs new question selected How should Olympians recover from errors? 5:08:54 pm
In Olympic QOCs new question selected How will messages be accessed? 5:09:00 pm
In Olympic QOCs new option selected By entering a password for How will messages be accessed? 5:09:16 pm
In Olympic QOCs new option selected By entering a password for How will messages be accessed? 5:09:18 pm
In Olympic QOCs new option selected By smart (swipe) card for How will messages be accessed? 5:09:33 pm
In Olympic QOCs new option selected Voice recognition for How will messages be accessed? 5:09:37 pm
In Olympic QOCs new option selected By smart (swipe) card for How will messages be accessed? 5:09:40 pm

In Olympic QOCs new option selected By entering a password for How will messages be accessed? 5:09:42 pm
In Olympic QOCs new question selected What kinds of telephones to use? 5:09:45 pm
In Olympic QOCs new option selected Push button for What kinds of telephones to use? 5:09:51 pm
In Olympic QOCs new option selected Dial for What kinds of telephones to use? 5:09:53 pm
In Olympic QOCs new option selected Both for What kinds of telephones to use? 5:09:55 pm
In Olympic QOCs new option selected Push button for What kinds of telephones to use? 5:10:08 pm
In Olympic QOCs new question selected How should non-Olympians interact with OMS? 5:10:15 pm
In Olympic QOCs new question selected How should Olympians recover from errors? 5:10:18 pm
In Olympic QOCs new option selected Press the * key for How should Olympians recover from errors? 5:10:24 pm
Opened new link Identifying an Olympian 5:14:14 pm
Opened new link Using the operator 5:14:25 pm
Opened new link Entering the password 5:14:31 pm
Moved to group Stakeholders in contextual focus view 5:14:49 pm
Moved to group Stakeholders in contextual focus view 5:14:49 pm
Moved to group General in contextual focus view 5:14:54 pm
Moved to group General in contextual focus view 5:14:54 pm

13. Appendix E List of Links

Olympic Links

LINK NAME	FROM	TO
Using the available technology to constrain the technology that OMS is based on	Contextual Focus:Description of available technology	Olympic QOCs:Available technology criteria
Users experience constrains the technology used	Contextual Focus:Description of experience with technology	Olympic QOCs:Users experience criteria
Working out which password to use	Olympic QOCs:Accessing by password	Olympic QOCs:Deciding on the password to use
Relating work of an Olympian to type of password	Contextual Focus:Olympians workload	Olympic QOCs:Information criteria
Information load determines the appropriate password	Olympic QOCs:Information criteria	Olympic QOCs:The appropriate password
Deciding on the appropriate type of telephone	Olympic QOCs:The appropriate telephone	Olympic QOCs:Available technology criteria
Deciding how non Olympians should interact with the system	Olympic QOCs:The appropriate telephone	Olympic QOCs:Interacting with OMS (Non-Olympians)
Processing information linearly	Contextual Focus:Information load	Olympic QOCs:Users recall abilities
Experience with technology constrains the number of functions offered by the system	Olympic QOCs:Users experience criteria	Olympic QOCs:Keeping commands to a minimum
Using pen and paper to make notes	Olympic QOCs:Keeping commands to a minimum	Olympic QOCs:Using pen and paper to make notes
Determining which languages to support in the system	Contextual Focus:Languages used	Olympic QOCs:Languages used in the system
Different languages take different times	Olympic QOCs:Languages used in the system	Olympic QOCs:Timing out

The environment constrains the kinds of instructions provided	Contextual Focus:Thermal environment	Olympic QOCs:Providing instructions
Entering the password	Olympic QOCs:The appropriate password	Olympic scenarios:Identifying an Olympian
Identifying an Olympian	Olympic scenarios:Identifying an Olympian	Olympic NUF:Identifying an Olympian
Using the operator	Olympic QOCs:Interacting with OMS (Non-Olympians)	Olympic scenarios:Using the operator
Sending a message	Olympic scenarios:Sending a message	Olympic NUF:Sending a message
information needed as password	Contextual Focus:Information load	Olympic NUF:Identifying an Olympian
previous knowledge of message systems	Contextual Focus:Task Experience	Olympic scenarios:Sending a message

IT Links

LINK NAME	FROM	TO
Application identification/specification	IT scenarios:Identified applications	IT NUF:Specified application
Automatically create letters	IT scenarios:Creating letters	IT QOCs:Automatic creatiion
Specification of letter creation	IT QOCs:Decision: creation of letters	IT NUF:Specified create letter
Why time is important	Contextual Focus:Other activities	IT QOCs:Decision: creation of letters
Academic culture	Contextual Focus:Academic culture	IT QOCs:Harmony
Who should use the system	IT QOCs:Who should use the system?	Contextual Focus:The users
Handling interruptions	Contextual Focus:Interruptions	IT QOCs:Handling interruptions
Shared office and effect on	Contextual Focus:Shared	IT QOCs:Control over

control	environment	creating letters
When mail is received	Contextual Focus:Receiving information	IT scenarios:Receiving mail/information
Structuring information	Contextual Focus:Receiving information	IT QOCs:Structuring infdrmation
Structuring information as a whole	IT QOCs:Structuring information as a whole	Contextual Focus:Receiving information
What details to show	IT scenarios:Summary details	Contextual Focus:Interruptions
Setting the summary details	IT QOCs:Summary details	IT NUF:Setting Summarcy Details
Current applicant	IT scenarios:Current applicant object	IT NUF:Specified Applicant
View	IT scenarios:View object	IT NUF:Specified View
Query	IT scenarios:Query object	IT NUF:Specified query
CurrentApplicant	IT scenarios:Current applicant object	IT NUF:Specified CurrentApplicant
Handling interruptions	IT QOCs:Decision on handling interruptions	IT scenarios:Handling interruptions
Accepting an offer	IT scenarios:Accepting an offer	IT NUF:Accepted Offer
Starting Up	IT NUF:Starting Up	IT UANs:Starting Up
Creating a query	Specify Query UAN:Starting the query	IT NUF:Starting Up
Creating a query (actual)	Specify Query UAN:Starting the query	IT NUF:Specifying a query
Choosing an access method	IT QOCs:Decision on accessing	Specify Query UAN:Starting the query
Choosing an access method (NUF)	IT QOCs:Decision on accessing	IT NUF:Specifying a query
Accessing applications	IT QOCs:Decision on accessing details	IT QOCs:Decision on accessing details
Test link	IT QOCs:Decision on	IT QOCs:Decision on

	accessing details	handling interruptions
Accessing applications (actual)	IT QOCs:Decision on handling interruptions	IT QOCs:Decision on accessing details
Waiting list and status	IT NUF:AdvancingStatus	IT scenarios:Waiting list

LD Links

LINK NAME	FROM	TO
Collaborative system issue	Lit development QOCs:Single user option	Lit development scenarios:Collaborative option
Available technology	Lit development QOCs:Technology criteria	Contextual Focus:Available technology
Link structure	Lit development QOCs:Link structure	Lit Dev NUF:Link specification
Generating criteria from requirements	Contextual Focus:Requirements	Lit development QOCs:Visualising links criteria
Accessing stats table	Lit development QOCs:Visualising links decision	Visualising links UAN:Display Table
Placing links table on screen	Lit development scenarios:Link creation	Lit development QOCs:Table placement
Table and placement	Lit development QOCs:Visualising links decision	Lit development QOCs:Table placement
Reasons for choosing particular editors	Lit development QOCs:App types	Lit Dev NUF:Chosen app types
Experience in notations	Contextual Focus:Background knowledge	Lit development QOCs:Expertise in notations
Handling changes	Lit development scenarios:Making a change	Lit Dev NUF:Handle change
Representing QOCs	Contextual Focus:Objectives	Lit development QOCs:QOC representation
Use of object lists	Lit development scenarios:ObjectList	Lit Dev NUF:ObjectList
Indicating the granularity of	Lit development	Contextual

a link by the size of the anchors	QOCs:Anchors for links	Focus:Requirements
Measuring the cost of a change	Lit development QOCs:Measuring impact of change	Contextual Focus:Evaluating the cost of a change
Identifying unused context	Lit development QOCs:Identfying unused context	Contextual Focus:Identifying unused context
Identifying the duplicate anchor points	Lit development QOCs:Identifying duplicate anchors	Contextual Focus:Identifying duplicate links
Calculating the effect of a change to an object	Lit development QOCs:Measuring impact of change	Lit Dev NUF:Measuring change to an object
Identifying unused context	Lit development QOCs:Identfying unused context	Lit Dev NUF:Displaying the unused context
Identifying duplicate anchors	Lit development QOCs:Identifying duplicate anchors	Lit Dev NUF:Displaying the duplicate links

14. Appendix F An Industrial Case Study at Intel Corporation

14.1 Introduction

This chapter describes a case study that was performed in an industrial context at Intel Corporation. Its aim was to see how industrial designers used contextual information in design and to identify any new problems with using context that were hitherto unsupported by current methods and techniques. By identifying and addressing such problems, increased knowledge about how context is used in design is gained. This knowledge can then be used to identify further ways to support designers in making efficient use of contextual information in design.

The study also provided an opportunity to see how and if LD would fit into an industrial context. This was a worthwhile study to perform since the linking tool had been designed in an academic context. The three studies described in the previous chapter were also carried out in an academic context, free from the different pressures and constraints that exist in an industrial setting (e.g., tight deadlines, developing a marketable product). This case study would validate the tool in an industrial setting and provide further requirements relating to its use.

The first section of this chapter briefly describes the design group at Intel. The second section then decribes a series of interviews that were carried out with members of the design group, during which they were asked to describe and talk about their experiences using contextual information in the form of ethnographic descriptions of particular environments. As a result of these interviews, the need for a different kind of support tool was identified, one that supports the understanding of contextual information. The second section justifies the need for such a tool while the third section describes the tool itself which I implemented at Intel. The fourth section of the chapter then describes some rough evaluations of LD that I asked the designers to perform by 'walking through' a paper mock up. The evaluation helped to determine if they thought such a tool would be useful and how they thought it would fit into the process. It also provided information relating to how both the tools (the one developed at Intel and LD) would fit into a design project that attempts to make effective use of contextual information in design. Such information is used to inform the development of a framework for contextual design which is described in the next chapter.

14.2 The Research Relations Group at Intel

I was lucky enough to secure a summer internship with the Research Relations group at Intel Corporation. This gave me the opportunity to study one example of how industrial designers attempt to use contextual information in design.

The Research Relations group was set up at Intel at the start of 1996, in order to collect and carry out research concerning future technologies such as multimedia communications devices, which would harness the power of the next generations of Pentium processors. Intel has grown into the most succesful computer chip manufacturer due to its speed in developing faster and more powerful processors. However, physical limitations inherent in the materials used to manufacture the processors will soon be reached and Intel will no longer be able to continually manufacture faster and more powerful processors. The Research Relations group was therefore set up to identify future technologies that would both harness the power of the Pentium processors and that would appeal to a wide range of different markets, so that Intel's success in the processor market would continue.

The Research Relations group consists of a number of sub–groups. The ethnography group consists of two ethnographers whose task is to investigate potential markets and describe these markets to the rest of the group in order that the group can identify potential products that both utilise the Pentium processor and that would be accepted in that market. While the whole Research Relations group gets to see and discuss the ethnographic descriptions, the technical concepts group, another sub–group within the Research Relations group, is the main target for the ethnographic descriptions. This group consists of two designers whose job it is to identify potential products or ideas in collaboration with the ethnographers and then to implement and demonstrate prototypes of these ideas to the Research Relations group and other groups within Intel. They use a number of prototyping tools such as Visual Basic and MacroMedia Director to demonstrate the ideas. Two other sub–groups support the ethnographic descriptions and research. The first of these support groups consists of one person who provides aggregate data in the form of surveys, opinion polls etc., to back up the findings in the ethnographic descriptions and to suggest further avenues for research. The second support group makes contacts with people in research labs and universities who are working on similar research areas that might be able to collaborate with the group in some way, and is made up of approximately eight staff (they also have the responsibility of attracting and arranging talks which do not have any relevance to the immediate needs of the rest of the Research Relations group but which may be relevant to other groups within Intel).

14.3 Interviewing the designers

I worked closely with the ethnographers and the technical concepts team to try to understand how product ideas were generated from the ethnographic descriptions and how the technical concepts team implemented demonstrations of these ideas. I interviewed the ethnographers, the technical concepts team and the person who supplies the groups with the aggregate data. When I arrived at Intel, the ethnographers were just completing an ethnographic analysis of American teenagers. The technical concepts team were however working on the previous ethnographic analysis that the ethnographers had performed, which concerned families with young children. In the interviews, I asked each person to focus on how they used, or were using, the descriptions of the families with young children to identify and demonstrate (when appropriate) product ideas.

Each person I interviewed gained their first exposure to the ethnographic descriptions via a talk given by the ethnographers. In the talk, the ethnographers explained to the rest of the group how they had collected the information. They then discussed their analysis of the information, which they presented in the form of a graphical model (see Figure 37), and then discussed briefly some product ideas or concepts that they felt were suggested by the analysis. After the talk, some discussion took place about the graphical model and what it meant. Each person then went back to their work after the talks and discussion and started trying to make sense of the ethnographic descriptions.

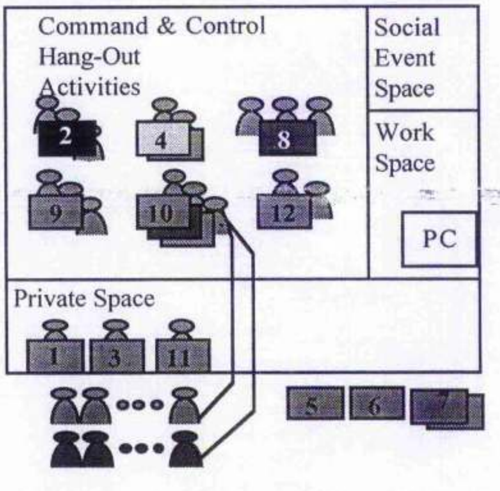


Figure 37. The graphical model of the families with young children analysis

It was clear from the interviews with the technical concepts team and the person who supplies the aggregate data that one of the most difficult aspects of identifying and demonstrating product ideas was in understanding the ethnographic descriptions. Even though each had been present at the talk, it was difficult for each person to start to make sense of the data when they started to use it to develop their own product ideas or to develop the ideas that had been suggested by the ethnographers. The product ideas or concepts that had been suggested by the ethnographers at the end of the talk were very abstract ideas which suggested for example that 'supporting effective communication is important' for families with young children. When the technical concepts team attempted to understand this in more detail, they experienced difficulties in understanding the details of the model. Some of the difficulties were in understanding the terminology used in the model. For example, knowing what sorts of activities constituted 'Hang–out activities' and what exactly happened in the 'Command and control centre' (see Figure 37). Some other difficulties concerned details that weren't described in detail in the model, such as the different time periods (the numbered and shaded boxes in Figure 37).

To address these difficulties, communication with the ethnographers was attempted but this was often difficult since the ethnographers were at this stage carrying out another ethnographic analysis (the American teenagers analysis) and were often off–site and not readily available to discuss details of the ethnographic model. When communications with the ethnographers could be established, it commonly took the form of short phone calls, email messages or quick conversations in the corridor. The teams were not always able to organise formal meetings to sit down and discuss the model in detail. Each member of the technical concepts team and the aggregate survey person was also involved in other projects which meant that they couldn't meet regularly to discuss their own understanding of the model. One member of the technical concepts team made five video clips of a video recording of the original talk given by the ethnographers. These video clips captured what this person felt were the important points from the talk. He used these clips to try to explain the model to other people who had not attended the talk and to explain his views of the model to those who had attended the talk. He realised that this was important to do since he was developing an understanding of the model that probably differed from other people's understanding, due to the lack of regular contact between the technical concepts team, the ethnographers and the aggregate survey person. The danger here was that each person may go away and develop their own understanding of the model. Hence by giving the video clips to others, he could in some way share his understanding with the other members of the group.

The team expressed other concerns about the process of generating an understanding of the ethnographic model. They wanted more detailed artefacts or descriptions to use in improving their understanding of the model but they did not want this increased detail up front. They did not want to be bombarded with large amounts of detailed information that would have to be absorbed in one go. Instead, they wanted the detail to be revealed as and when it was required. The detail would have to emerge as the understanding progressed.

It was clear that it would not be possible for the technical concepts team to gain a full understanding of the model in the time that they had available. They needed to start demonstrating some ideas within five or six weeks. Instead of trying to understand the whole model, the team decided to focus on one part of the model and produce and demonstrate a design idea related to that part. They decided to focus on the communication aspect of the model, since this was suggested by the ethnographers as one area of the model that could potentially be used to develop a useful product that would fit in to the families with young children market. This caused some concern on the part of the ethnographers since they felt that the model presented a rich set of concepts that could be combined to develop a product that would have a great impact across all areas of the model. However, due to the communication difficulties experienced by the different teams, it was not possible for the ethnographers to engage in detailed discussions about the model with the technical concepts team to suggest other product ideas.

The technical concepts team focused on the types of communication support that was required, where and when this communication would take place and how it would work. Since they were focussing on a particular aspect of the model, they found it easier to ask detailed questions about one part of the model, since they did not have to think about how this communication aspect of the model related to the rest of the model. They were therefore able to get more detailed answers to their questions, since their questions were focused and specific. Eventually, the technical concepts team were able to start developing a product idea into a prototype. The product[12] would support all the necessary and important communication activities that occur between family members. It would be easily accessible and fit in with the pattern of their daily lives.

It took approximately two weeks for the technical concepts team to develop the prototype compared with the four weeks that it took them to develop an understanding of the ethnographic description. They produced both a physical prototype which demonstrated the physical appearance of the appliance, and a functional prototype, using Visual Basic, which demonstrated the various features.

The technical concepts team described how they attempted to validate their design ideas against the model. Each time they thought of another feature, they inspected the ethnographic model in an attempt to identify aspects of the model that would either support or reject this idea. They felt that they needed to validate the ideas so as to ensure that ideas

[12] Unfortunately, precise details of the product cannot be given as Intel are currently developing it for general market release.

were not being pursued just because they were interesting. At times it was difficult to keep within the constraints set by the model. They needed to maintain some amount of discipline to constrain themselves to stay within the knowledge base represented by the ethnographic model.

The model was not always perfectly suited to validating design ideas. While the model suggests the need for some kind of scheduling feature that lets members of the family know the plans of each other family member, the model does not indicate that the important point of such a feature is not to record the fact that something happens regularly , but to deal with exceptions, when something breaks down (such as someone having to work late). The team felt that the model didn't capture or represent this well so there was some translation involved for the team, in going from the model to their ideas. The difficulties experienced by the team echo the difficulties I experienced using Customer Centred Design models to model single attributes of context such as breakdowns. Here, the designers were using a model that necessarily abstracted over some of the details of families with young children. The checklist (see Appendix A) that I used to alleviate some of the difficulties with modelling attributes of context may also have helped the designers at Intel.

The team members also claimed during the interview that every feature they came up with for the prototype was not necessarily suggested by the model. They used their own experiences of being in a family with young children to develop some of the features contained in the prototype. Also, other factors unrelated to the ethnographic model were used to reject some design ideas. Hardware considerations and time pressures were two factors which weren't related to the ethnographic model but which were used to reject some design decisions.

14.4 Analysing the interviews

Once I had carried out all the interviews, I began to think about the conclusions that could be reached from the interviews and the implications that these conclusions would have for LD. It seemed to me that there were two main conclusions.

1. Validating design ideas against the model is a useful practice but it can also be difficult due to the use of factors that have been incompletely specified in the model or factors unrelated to the model being used to validate design ideas. Some aspects of the model also require some amount of reinterpretation or translation in order to validate design ideas.

2. Thinking about and understanding ethnographic descriptions of a culture or domain is difficult. Many of the comments given above relate to the difficulties that the team experienced in understanding the ethnographic model.

The first conclusion implied that LD may be useful for validating design ideas. The second conclusion suggested that some kind of support may be required to enable developing an understanding of ethnographic models. The next section describes my attempts at demonstrating LD to the team to see if they felt it could be used to perform the validations they require. Following this section, a tool is described that was designed and implemented to support the team in understanding ethnographic models.

14.4.1 Demonstrating the linking tool

The technical concepts team, the team responsible for using the ethnographic data to implement and demonstrate prototypes, were shown paper mock-ups of LD[13]. I walked through the main features of the tool with them, showing how designs, contextual information and relationships between context and design could all be specified. By displaying the appropriate screen representation at the appropriate time, I was able to demonstrate how the linking tool might support the validation of design ideas. I demonstrated how a designer would create a link between some contextual information and the design. I demonstrated how the links would be used to indicate parts of the design that have not been influenced by the context and parts of the context that have not influenced the design. I also demonstrated how the links could be used to maintain a system, by indicating the factors that are potentially affected by a change to the design or the context. I then asked both team members for their comments on the tool, whether they felt it would be useful in their work and what they thought might be some of the problems with the tool.

In general, both team members felt that the tool would be useful in their attempts to validate design decisions and to determine what aspects of the context or ethnographic descriptions were used to influence a design. They agreed that by making the links between context and design explicit, many different kinds of assessments about the use of context in design could be made, and that these assessments could be useful.

However, they did raise a number of concerns about the representation of the links. These are listed below:

- Both members of the technical concepts team felt that it would be important for any team that attempts to represent the links between context and design to agree on what the links

[13] Paper mock-ups were used as a version of Smalltalk was not available with which to demonstrate LD properly.

mean. For example, they felt that there was a danger that the linking tool could be used simply as a means to record design discussions and not to provoke them. A team could use the links simply to record what was said at a design meeting. In this case, the links may or may not be completely valid, since they would only indicate that these relationships were considered at the design meeting. The relationships would simply be a record of what the design team had considered not of actual relationships between context and design;

- In order that maximum benefit could be gained from the tool, both members suggested that policies of use for the tool would have to be defined by each design group using the tool. The policy would indicate the appropriate links, design and context representations that should be created in the tool. These policies would have to be enforced by the design team, a requirement that increases the effort of the design team. They did not think that a policy of use could or should be enforced by the tool, since such policies should be flexible and allow some amount of leeway, something which might be difficult to define for a computer based tool;
- They also suggested that the linking tool may only be fully appreciated by a large design team, or one in which communication between team members is difficult. In the case where team members are in constant communication it may be simpler to have a team member explain the relationships between context and design rather than rely on the tool;
- Lastly, they stressed that it would be important for the tool to allow the designers to use whatever notation or format they feel is appropriate to record the design and context. They did not want to be constrained to using particular notations or techniques to describe the design and contextual information.

The informal walk through of the linking tool and the discussions raised by the walk through indicated that the technical concepts team felt that the tool was potentially useful. They raised some concerns relating to the actual usage of the tool but felt that the goals that it was aiming to support (validating design ideas against contextual information and making assesments of the usage of contextual information) were worthwhile and indeed, they had aimed for these goals but had encountered difficulties.

14.4.2 Designing support for understanding ethnographic models

However, it appeared that most of the difficulties relating to using the ethnographic models to develop product ideas concerned generating an understanding of the models. Developing tool support for this process, such that these difficulties might be alleviated became the main focus of the summer internship at Intel. The next section describes the design of GENEVA, a tool that I implemented to support the development of a shared understanding of ethnographic models.

14.5 GENEVA: A tool to support understanding of contextual models

Geneva is a multi–user communications tool that lets users participate in and initiate discussions about some common area of interest. This common area of interest is represented as a graphical model. Discussions are tied explicitly to parts of the graphical model thus providing discussions with a common context and a simple means to refer to graphical features of the model.

Geneva can be used to support discussions about any model that can be represented in a GIF image. Hence, with respect to supporting designers in generating an understanding of contextual data, Geneva does not force one particular representation to be used over another. As long as the representational medium can be displayed as a graphical image, then Geneva can support discussions about the image. Hence textual models can be used, since these can easily be represented in a GIF image.

14.5.1 Motivation

Geneva was developed in response to the problems that the technical concepts team encountered in attempting to understand the ethnographic models produced by the ethnography group of the Research Relations group at Intel Corporation. Communication difficulties between two groups, one group responsible for collecting contextual data and the other group who were responsible for using that data in design, meant that a good, shared understanding could not be reached. This resulted in the contextual data not being used as efficiently as it could have been and meant that a lot of the data simply wasn't put to use in the resulting design. The technical concepts team focused on the communications support for families with young children. The ethnographers felt that potentially more useful product ideas could have been developed had the whole model been considered but the communication problems between the different teams meant that the technical concepts team were unable to take such an approach.

To try to address these problems, a project was initiated to investigate and implement a possible solution to the problem. After discussions with members of the design teams, which took place individually, a set of goals were derived which a computer based tool designed to support the development of a shared understanding would have to support. These goals are that the tool should:

- *Support communication and understanding*. Group members are busy and work on different projects making it difficult to synchronise communications. The tool should support active participation in all relevant communication for design team members at appropriate times;

- *Make knowledge public.* Providing access for all members of the design team to all discussions that take place about the model, supports the creation of a common picture or understanding of the model;
- *Encourage discussions.* The tool should encourage active participation in discussions since only through participation in discussions by all members of the design team can a shared and deep understanding of the model be created. Having access to the discussions that are taking place about the model makes people aware of the different issues that are being discussed and can trigger other questions that may be raised;
- *Focus on concepts.* The tool should not be viewed as an archival system which simply keeps track of all the discussions that have taken place, providing a variety of discussion maintenance facilities. Instead, it should focus on developing the understanding such that design concepts or ideas based on the understanding start to develop. In other words, the people using the tool should focus on developing an understanding of the model and not on keeping the discussions well maintained;
- *Be lightweight.* The tool should be lightweight and simple to use since users are busy and do not have time to learn how to use a new tool. Keeping the tool lightweight, helps focus users on design ideas and not on the tool.

14.5.2 Geneva

Geneva supports the above goals by providing simultaneous multi–user access to the different conversations that take place regarding a graphical model. The discussions are modelled in a similar way to Usenet newsgroups. Users can see the list of individual messages within individual discussions and can view particular messages. Users can respond to messages in a new message that 'quotes' the original. This is fairly typical and standard functionality which will be familiar to users who have browsed Usenet newsgroups, but is simple enough for users unfamiliar to Usenet to understand and pick up.

One problem with Usenet newsgroups however is that there are hundreds of different groups, within which there are potentially hundreds of different discussions taking place. The only way that users can access these discussions is to scan the list of newsgroups, find one of interest, open it, scan the list of discussions and then select one of these to read. Users can be overwhelmed by the sheer number of newsgroups and may be put off reading any of the discussions due to the effort required to find a newsgroup of interest. In supporting discussions about contextual models or ethnographic data, the potential for a similar situation occuring is high, since there could be a large number of discussions generated about the model. In order to avoid this situation and to encourage users to take part in discussions, Geneva provides access to the different discussions through the graphical model itself. Different discussions about the model are differentiated by different 'hot-spots' which are placed over the model. Users can click inside the hot-spot to access the discussions. The hot–spots provide a certain amount of contextual information in that they suggest that the discussions that can be accessed from this hot-spot relate to the area of the model that the hot-spot is drawn over. Hence, within discussions, users can ask things such as "What is the significance of this aspect of the model being separate from the rest of the model?" without having to describe what they mean by 'this' since this is defined by the area that the hot-spot covers. The hot-spots also identify areas of the model for which discussions have been created. Any parts of the model which do not have hot-spots drawn over them are either irrelevant for the present purposes or have been overlooked.

Users can access discussions whenever they like, and hence can read and send messages as they wish. Since messages are accessible by every member of the design team, everyone can share in the discussions and hence a common understanding of the models can be reached. The discussions can be accessed over a suitable network, such as the Internet, so members of the design team need not be in the same place to access the discussions. Therefore the tool alleviates the problems discussed earlier. The next section now describes how users interact with Geneva and how it appears to the end user.

14.5.3 Interacting with Geneva

After connecting to the system, users are presented with a window (see Figure 38), within which is displayed the graphical model from which discussions are accessed. Above the graphical model, at the top of the window, is a menu bar from which the user can access different functions such as closing the connection to Geneva. The diagram below shows the window that users will see when they connect to a discussion about families with young children.

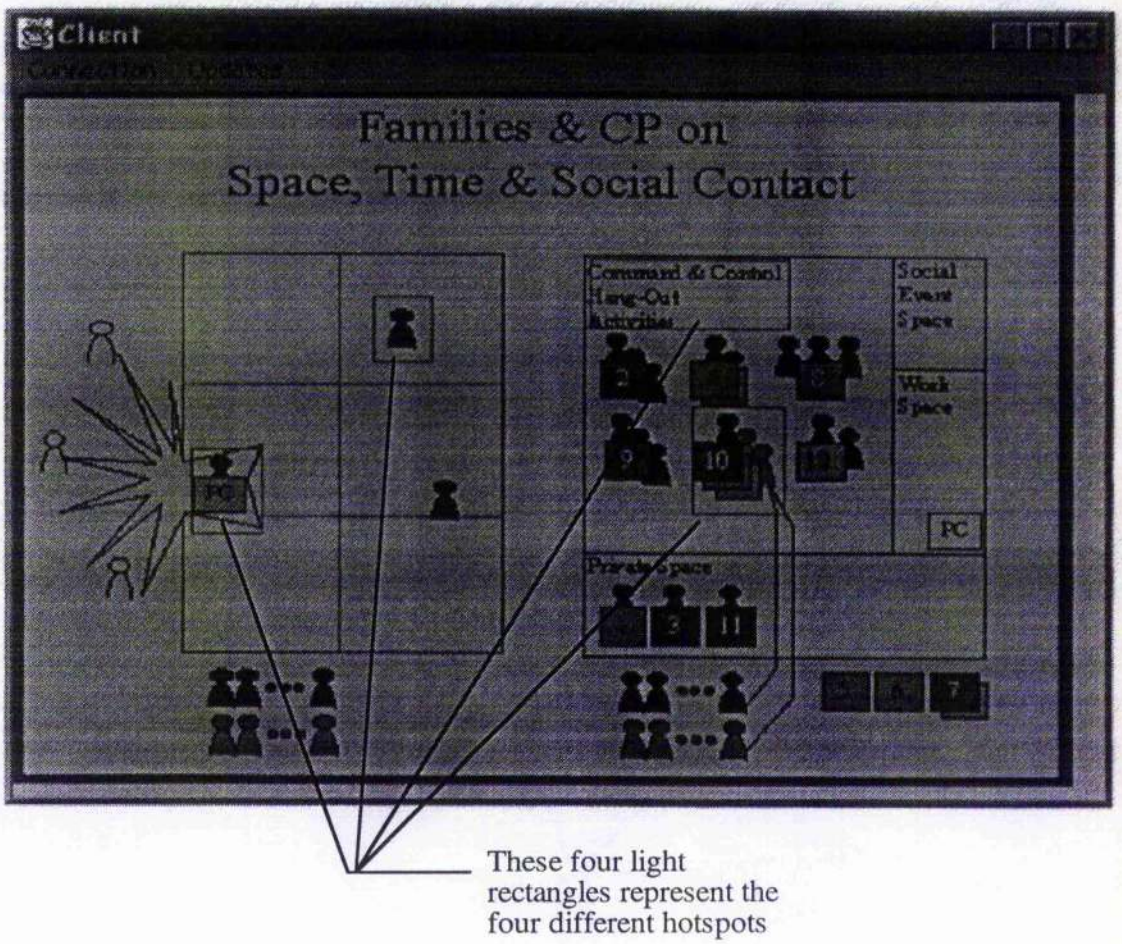


Figure 38. An example of a graphical model in GENEVA.

Four hotspots have been drawn over the main model. Users can click anywhere inside these hotspots to access the discussions. If the user clicks inside the rectangle surrounding the text 'Command & Control/hang-out/Activities', a window appears on top of the main model, showing the discussions that are taking place related to this hotspot (Figure 39).

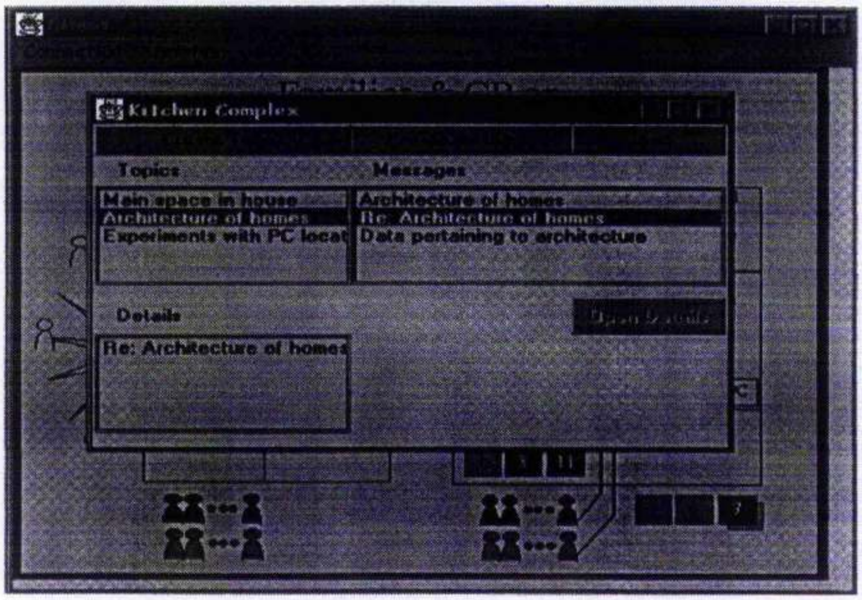


Figure 39. View of discussions that are relevant to area of model.

Discussions are placed into different topics. The list at the top left of the window displays all the topics that are accessible from the 'Kitchen Complex' hotspot. Each topic contains a number of messages. They are listed in the list at the top right. Users can click on individual topics to see its messages and can then open a message by selecting one and clicking the 'Open' button. At the bottom left hand corner of the window, a list displays the "details" for this hotspot. Details are important messages or items related to this hotspot. They may summarise a long discussion and capture new understanding generated through discussions. Summarising the discussion and placing the summary into the details section means that users do not have to read all the messages to reach the same level of understanding. Instead they can read the summary, which should provide the same amount of information as the sequence of different messages that the detail summarises. Figure 40 shows a typical message window that users would see if they opened a message from the message list or if they opened a detail from the details list.

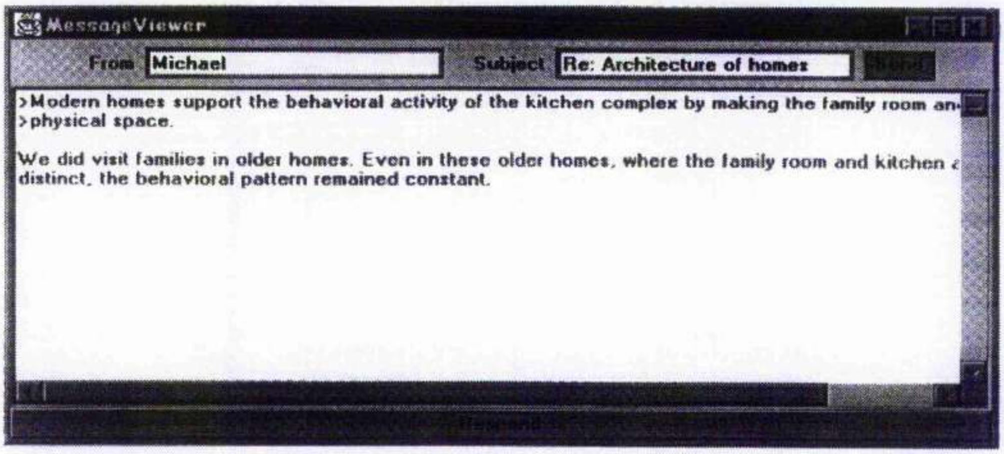


Figure 40. Users can read individual messages and respond to them.

Users can read the message and then simply close the window in the usual way. They can choose to respond to the message by clicking on the large 'Respond' button at the bottom of the window. This opens a new window with the original message copied and 'quoted' at the start of the new message. Users can then type in their response and send the new message by clicking on the send button at the top of the window. The message is then posted to the system. If the same hotspot is open on any other machine, it will be updated to show the new message.

14.6 Chapter Summary

Interviews with the designers about their experiences in attempting to make use of contextual information (in the form of ethnographic desriptions) in design identified two main problems:

- Understanding the contextual information. It was clear that the assumption made in methods such as Contextual Inquiry (Wixon & Raven, 1994) and Customer Centred Design (Holtzblatt & Beyer, 1993) that the people who collect contextual data are the same people as those who use it does not always hold. At Intel, the people that collected contextual data and the people that used it were different people. This led to some serious difficulties using the data efficiently, as well as raising doubts about the data in the minds of the people attempting to use it. Since they were not involved in collecting the data and did not have 'ownership' of the data, they were less inclined to believe the data as presented to them. It is vital therefore to support the understanding of the data and to increase the confidence in the data. The chapter described Geneva, a tool to support multi–user discussions about contextual data such that a common understanding and consensus about the data can be reached.

- Validating design ideas against contextual information. LD was identified as a potentially useful tool by the designers to support their goals of validating design ideas.

Understanding contextual data has been shown to be a non trivial task. Effort needs to be spent on understanding the contextual data, since to do otherwise would at best, result in only a subset of the data being used to influence the design (as was the case at Intel) or in the worst case, could lead to contextual data being used inappropriately. Appendix G takes the work presented here and in Chapter 3 to describe a framework for making effective use of contextual information in design, which draws together the stages of collecting contextual information, developing an understanding of it (identified in this chapter), using it to influence abstract design ideas (identified in this chapter and in Chapter 3) and using it to turn abstract design ideas into concrete systems (identified in Chapter 3).

15. Appendix G Understanding and Using Contextual Information in Design

15.1 Introduction

In Chapter 1, it was argued that in order to design a product succesfully that fully supports users in their work, designers must make more effective use of contextual information. It was suggested that this is no easy task, since there are many contextual factors which can influence the way a product is used. Chapter 2 highlighted some of the problems that exist for designers when attempting to use existing methods (understanding context descriptions, representing relationships between context and design, maintaining the system, defining an appopriate scope of contextual information). Chapters 3 and 4 and Appendix F described a number of case studies which identified a number of distinct tasks that need to be supported in order that designers can make effective use of contextual information.

Addressing the problems described in Chapter 2 and taking into account the tasks identified in Chapters 3 and 5, this chapter describes and defends a four stage framework for using contextual information in design:

- **Stage One.** Defining the *scope* of relevant contextual information and then collecting it.
- **Stage Two.** *Generating an understanding* of the contextual information. Since context, and the interaction between different components of context is so complex, designers need a deep understanding of the context and the influence that the context has on users before they can start thinking about the design of a product to fit into that context.
- **Stage Three.** *Comparing alternative designs* or prototypes and evaluating them to see how they fit into the context. Designers need to be able to validate their ideas against the contextual information, so that they can predict how succesful an idea may be. They also need to compare alternative designs with respect to the context so that they can choose the most appropriate one with good reason.
- **Stage Four.** *Specifying the design* that was chosen in the previous stage and using the contextual information to inform the concrete design of the product.

Figure 41 illustrates the four stage framework

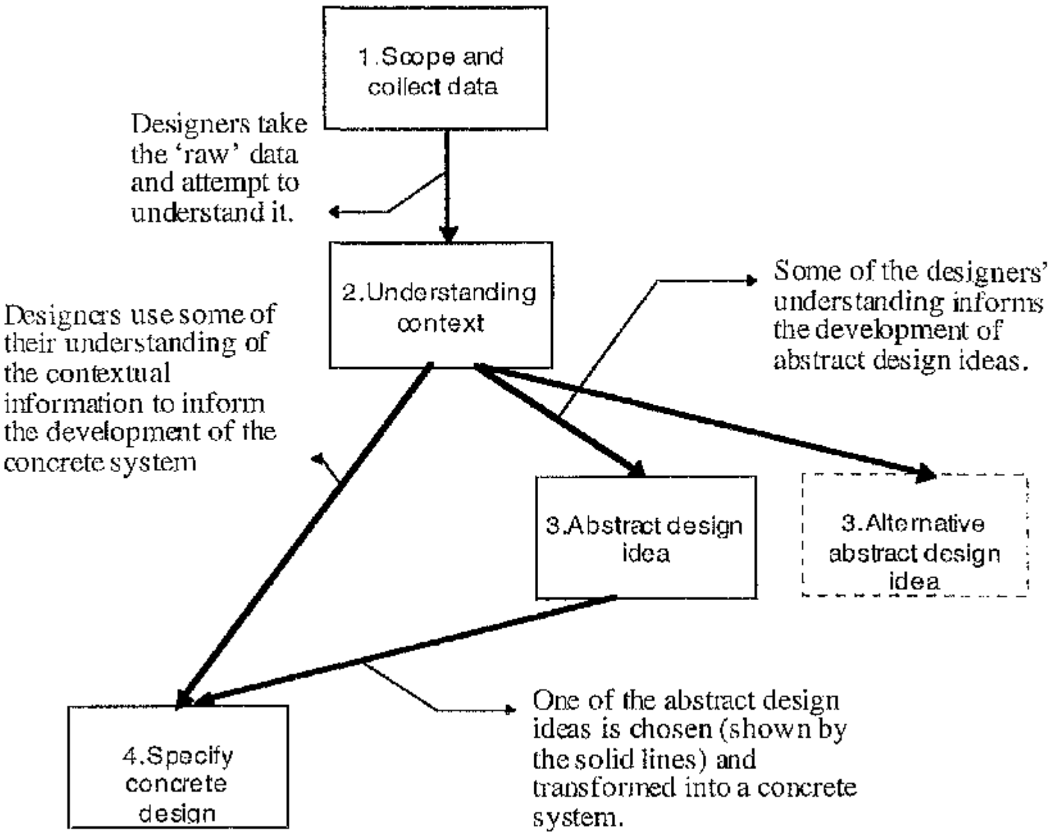


Figure 41 Four stage framework for contextual design

15.2 A framework for using contextual information in design

In order to make efficient and effective use of contextual information in design, this thesis argues that it is essential to define the scope of relevant contextual information, to collect this information, understand it and then use it to define and validate product ideas and actual products. Furthermore, the thesis argues that explicit links between contextual information and design ideas or designs are also necessary to facilitate maintenance and evaluation of the influence of contextual information on design ideas or designs. By supporting maintenance, designers can be sure that changes in the context or the design are reflected by appropriate changes in the design or context. By supporting evaluation, designers can determine if more work is required on any aspects of the design or contextual data collection/understanding by evaluating how well the contextual information has been used in the design. These stages and the issues involved in each are described below.

15.2.1 Defining the Scope of Relevant Contextual Information and Collecting It

In Chapter 2, a key difference between methods for using contextual information within the software development lifecycle was the scope of contextual information implied by each method. Differences in focus, elicitation methods, post processing and actual usage of the data resulted in differences of scope. In some methods, the scope centred on those aspects of context that could be articulated. In others, it centred on those aspects that were relevant to individual consciousness. No method was designed such that the scope implied by the method covered all possible aspects of contextual information. While a method might focus on some aspects of context, it ignores others. For example, Chapter 3 described how different models and notations were used to model the contextual information (Customer Centred Design models and the Contextual Checklist). It was clear that the Customer Centred Design models were not suited to recording quantitative information such as the number of times a day the applications secretary is interrupted but that the contextual checklist was well suited to recording these kinds of data.

The scope of a method has a direct influence on the ability of that method to make efficient use of contextual information in design. If the method ignores aspects that are crucial to developing a full understanding of the work to be supported, then it is likely that the support produced will be poor. However, it is unlikely that one method alone will be sufficient to collect and model all the relevant contextual information within the scope defined for a particular project. Each method described in Chapter 2 was shown to collect and use different kinds of contextual information due to the different foci, elicitation methods and usage of the data employed by each method. It is difficult to see how the factors that play a part in shaping the scope implied by a method can be reconciled such that a single method could cover all aspects of context.

For example, interviews only collect those elements of data that users can articulate. Using prototypes only collects data about the prototype and its level of support for users' work. It does not collect data about other aspects of work which are unrelated to the prototype.

As another example, it is difficult to see how a method could be designed that described a way of using the data collected to identify objects that should be implemented in an object oriented system or that suggested screen designs (in the way that Holtzblatt and Beyer use contextual information to design User Environment Designs). Both requirements (identifying objects; suggesting screen designs) imply different information requirements and different levels of analysis which may not be compatible with one another.

Designers should be made aware of the different scope that each method implies, so that they are aware of the data they are collecting and the data they are ignoring. Designers must be allowed flexibility in the methods they use, so that for example, when they recognise that some important data is being ignored, they can use another method to collect the data. Forcing designers to use a particular method forces them to ignore certain aspects of contextual data.

The above discussion implies that designers have an idea about the contextual information they are interested in. This must be the case, since if designers do not specify the kind of data they want to collect, how will they know when they have collected it? Even ethnographers define a purpose to their data collection, even if it is very broad and vague, for example "To identify the structures that define teenage culture". Defining the scope of relevant contextual information defines what is important and allows priorities to be made about the order in which data should be collected. A scope sets boundaries and limits on what data is collected, so that time is not wasted collecting irrelevant information. A scope is not rigid however, and can be changed as a result of increased understanding, certain design choices etc. Defining what is and is not relevant to a particular project is difficult and depends in part on the project undertaken. For example, the height of a person is not very relevant to the design of a hand-held computer but is very relevant to the design of a cash machine. As the project proceeds, and designers become aware of more of the issues involved in the project, the scope of the relevant contextual information will evolve and change as the issues that the designers are grappling with change. Design choices will make some aspects of context relevant and others irrelevant. For example, Chapter 3 described how the means by which the applications secretary most easily identifies individual applicants became relevant after a design decision was made to display a dialog box (to display all those applications which have not been updated within a certain period of time). The dialog box was limited in size and so only a small subset of the application details could be displayed in the dialog box. It was therefore important to know how the applications secretary identifies applications so that this information could be displayed in the dialog box. As the scope changes, methods used to collect and analyse the data may need to be changed also, since it is unlikely that any one method will be able to collect all the different kinds of data that designers need. It may also be useful to keep a history of changes in scope, including *why* the changes occurred. Carroll *et al.* (1994) describe the Raison d'Etre tool that records the history of a design project as the project proceeds. They suggest that access to such historical records of a design project can be useful to designers in learning about design and periodically reflecting on the design project.

Defining the scope of relevant contextual information is important in any project since it gives a goal for the data collection. This is important since it gives a means to evaluate what has been collected and what should be collected. Once enough data has been collected to satisfy the goal, or the scope, data collection can stop and members of the project team can start understanding and using the data. The scope itself can be evaluated and should probably be revised, as data collection and understanding proceeds. Revisions in the scope can help identify what extra data should be collected. However, revising the scope must be a manageable process. Designers should not be expected to fully revise the scope (i.e., throw away the previously defined scope and define a completely new scope) at regular time intervals throughout the project. Rather, they should revise the scope in response to certain design decisions that have suggested new items of relevant context. Designers should focus on these new items of context when revising the scope. There are other research issues involved such as exactly when it is appropriate to revise the scope (after identifying one, ten or a hundred new items of context?) and when to stop revising the scope (after considering only those new items of context that were identified as a result of certain design decisions or after no new context can be identified?). These questions and others are beyond the scope of this thesis. However, it is important to stress that the scope should be revised as the design project proceeds. The scope is not static and designers must be aware that new elements of context may become apparent after the initial scope has been identified and that they must be taken account of.

In summary, the first stage of the framework necessitates an awareness on the part of the designers of the differing scopes of contextual information that each contextual method implies. It also demands that designers specify the scope of the data they wish to collect so that they know when it has been collected and so that they can choose the appropriate method to collect the required data. Since no single method is ever likely to collect all kinds of contextual information, designers should not be forced into using a particular method, but should rather be given the freedom to choose whichever method suits the purposes of their project.

15.2.2 Understanding Contextual Information

In Appendix F, a case study at Intel Corporation was described. Designers were interviewed about their experiences in using contextual information. One major problem was trying to understand the data. The data had been collected and modelled by another group and it was the designers job to use this data to develop product concepts or ideas. They found it extremely difficult to understand all of the data presented to them however and decided just to focus on understanding a subset of the data. As a result they were able to produce a prototype that demonstrated a product concept based on part of the data, but to the disappointment of those who had collected the data, large parts of the data were unused. These difficulties in understanding data and the resulting consequences suggest that understanding contextual data is vital, but difficult. This is especially so in the case where those who have to use the data are not the same people as those who collect the data. In this common situation, one group of people have to understand data that has been collected by another group. The group that collect the data do not know what the group that are going to use the data need to know and the group that are going to use the data do not know what the group that collect the data know. Hence, some form of detailed communication needs to take place between the two groups to generate an understanding of the data in the group who are to use the data.[14]

It is important to ensure that a design team fully understands the contextual data that has been collected. Many diverse representations of the data may be used such as scenarios, graphical models or even prototypes. In order that the whole team can participate effectively in discussions about the data, the whole team must understand the data and its particular representations. Without such an understanding, confusion about major issues, priorities, important aspects of users' work etc., is likely. At Intel, the technical concepts team focussed on one particular aspect, communications, but did not consider temporal aspects or many social aspects that were also represented in the model. As a result they developed a product that addressed few needs of families with young children as identified by the data collectors (the ethnographers). This could potentially reduce the chances of the product succeeding in the families with young children market.

Confusion regarding the contextual information will at best slow down the design team down, and at worse, will lead to mistaken assumptions being made. This will result in a system being designed that does not fit its context or support the work that users need to do. Efficient use of contextual information in design has not been achieved.

Understanding must be shared and common across all members of the team so that there is little ambiguity about the current topic of discussion. Creating a deeper

[14] According to Kyng (1995) this cannot be done. He suggests that the people involved in collecting the data should also use the data in design, but as Nardi (1995) points out, many commercial organisations do not have the resources to permit such a situation.

understanding of contextual information makes designers more acutely aware of what would (not) work in the context. Therefore more feasible design ideas are likely to be pursued. With a greater understanding of the context, members of the design team are more likely to be able to make valuable contributions to discussions.

Those responsible for collecting the information should not be under the impression that they understand the data fully and that the rest of the design team simply have to come up to speed with them. Rather, sharing the process of understanding the contextual data across the whole team has the effect of increasing the insight into the data, since many pertinent questions will be asked about the data, questions that those who collected the data may have thought they knew the answers to but in reality didn't. By asking questions and sharing in conversations about the data, all members of the group share in the understanding created through the conversations. Once conversations begin to discuss how to use the contextual information and there are discussions about what the contextual information means, it can be assumed that a good understanding of the data may be achieved. Discussions about the understanding of the contextual information should not stop as soon as design begins however. As work on design ideas proceed, more understanding will be gained and this must be fed back into the discussions.

15.2.3 Demonstrating and Evaluating Design Ideas

Rosson and Carroll (1995) make full use of claims analysis in their scenario based design method. They argue that claims analysis ensures a deeper understanding of the objects within scenarios and the relationships between design features and other scenarios or design features. For example, the consequences of one claim can lead to the identification of other scenarios which illustrate each consequence. Claims can rely on other claims and design features can be evaluated with respect to their claims and their relationships with other design features and other claims. This process is termed claims analysis by Rosson and Carroll and is related to design space analysis and work on design rationale (Maclean *et al.*, 1991).

Interestingly, Karsenty (1996) claims that, while recording the design rationale of a system is useful for system designers and maintainers, it is not sufficient. Designers cannot generate a full understanding of a design from the design rationale, nor can they fully explore the whole design space through an analysis of the contextual information and the design rationale alone. Design does not simply proceed from an analysis of the contextual information to design space analysis. Often, designers will have a design idea before analysing the relevant design space or contextual information. Indeed, this design idea may identify some aspect of context as relevant, whereas beforehand it was considered irrelevant (see Chapter 3 – the dialog box presented to the applications secretary to identify applications which have not recently been updated). In this case, the design idea is driving the context analysis. Hence, design ideas need to be recorded, not just through the design rationale notations, but also through other design specification notations, such as NUF (Cockton *et al.*, 1996) since some design ideas will be created before any contextual or design space analysis has taken place. Thus designers need a certain amount of flexibility in recording their design ideas. They should be able to record their ideas using whatever notation is appropriate and they should be able to record a design idea both after analysis of the contextual information and design space and before such analysis.

At this stage of the framework, designers should just be concerned with developing design ideas. Design ideas can be abstract, high level descriptions of the kinds of tasks and design features that the design will support. As has been discussed, developing a number of different design ideas can open up areas of context which were previously not considered relevant, hence generating a richer contextual analysis. The different design ideas can be judged against this contextual analysis and the judgements can help suggest better design ideas. Design ideas can be specified in any particular way that suits the designers. A design idea may have been suggested by an analysis of the design space and so a design space analysis notation such as QOC may be appropriate. Another idea may simply have been suggested without analysis of the context or design space and so a simple design specification method or notation may be more appropriate. However, crucial at this stage is that design ideas are specified. For example, in the IT case study (see Chapter 3) one of the design ideas that was suggested was that the system should provide better feedback about the status of different applications to each applicant by automatically creating letters at appropriate stages in the application process. This was specified using NUF. NUF permits incomplete specifications of designs and does not force designers to specify accurately how a design feature or idea will operate. These features were most useful, since at that stage, it was not known how or at what stage letters would be created. But it was possible to specify the 'automatic creation of letters' feature in the NUF and then use this feature to drive further analysis of the context such that questions could be answered.

Design ideas suggest features and tasks that a design based on the design idea should support and in turn identify relevant contextual factors. For example, in the study of the process of handling applications for the IT course, identifying the design idea of providing better feedback about applications to applicants highlighted contextual factors such as the status of an application, the last date for which an application was updated etc., as relevant. It is unlikely that these factors would have been considered had this design idea not been specified. Specifying design ideas in this way limits the amount of extraneous information collected, since most contextual information collected will result from design ideas arising during initial data collection. Ideally, several ideas should have emerged during previous conversations about the contextual data. These ideas need not be specified in detail, but as the design team requires.

However, as has been described previously, the way in which the scope of relevant contextual factors changes needs to be managed well. It is too costly to revise the scope each time a designer in the team believes that they have found a new element of relevant context, since this would entail reviewing all the design decisions and design documents that have been created so far to see how they relate to the revised scope. Instead, it would be less expensive to review the current scope to see if it covers the new element of context (perhaps the element of context has been described differently in the scope). Only when a body of new contextual factors that the whole design team agree have not been included in the scope, does it make sense to revise the scope. Then the cost of revising the scope may be more affordable since the extent of the change to the scope is similar to the amount of work that will have to be done reviewing the design decisions etc. The relationship between design ideas and the scope of relevant contextual information has similarities with Kuhn's notion of *Scientific Paradigms* (Chalmers, 1994). Kuhn suggests that scientists work within a particular paradigm, which is a structure of scientific theories and axioms which sets the standards for legitimate work within the science it governs. Scientists will always come up against problems or anomalies in their work but it is too costly for the scientist to adopt a new paradigm since this would involve a re–appraisal of all aspects of the scientists work. Rather, the scientist attempts to resolve the problem (perhaps by re–calibrating their apparatus, taking measurements again) before attempting to modify the paradigm. Only under special circumstances should unresolved problems undermine confidence in the paradigm. For example, if a problem strikes at the core structure of a paradigm and yet cannot be resolved by any scientist working in the paradigm it may then be preferable to adopt a new paradigm, one in which the problem can be solved. In using context in design, one can view the scope of relevant context as a scientific paradigm and the designers in the design team as the scientists. As each designer uncovers a piece of relevant context they believe is not included in the scope they should not first of all abandon the current scope and attempt to define a new one such that it includes the new context. Instead, they should attempt to fit it into the current scope. Only when the whole design team cannot identify a way in which some context can be included in the scope does it make sense to revise the scope. Kuhn's notion of paradigm shifts makes clear that shifts can only occur with the consensus of all scientists working within that paradigm. Likewise, consensus across all designers in a design team must be reached before the scope will be revised.

Contextual information and design ideas are very closely linked. Design ideas may have been suggested by analysis of the context or they may have been validated during the design space analysis, with contextual information providing criteria with which to judge different design ideas. Design ideas may also have been specified before analysis of the context or design space. Hence design ideas can identify relevant aspects of the context and design as well as vice versa. These relationships between context and design ideas are important since they show the ways that context has been used in the design. This thesis argues that explicit links between the contextual data and the design ideas are required to make these relationships explicit. As was described at the end of the Chapter 2, Rosson and Carroll (1995) and Kyng (1995) provide some justification for including links between different specifications of the system and of its context but only use these links to demonstrate the pervasiveness of a claim, to illustrate all the consequences of a claim or to provide further explanatory material about various scenarios. Other uses for links between context and design specifications were identified during the IT case study described in Chapter 3 and in the case studies described in Chapter 4 and at Intel (Appendix F). Such uses were

- Identification of parts of the contextual information that have been used in influencing the design more than once (with potential inconsistencies);
- Identification of contextual information that has not been used in design (which can be used to identify relevant and irrelevant context);
- Calculation of the number of links between different design and context specifications (which can be used to identify areas where further work is required);
- Identification of parts of the design that have not been influenced by the context.

These assessments can all increase the efficiency with which contextual information is used in design. Increasing the influence that context has had on a design will increase the chances of the design fitting the intended context of usage.

The links also support maintenance of the design, since as the design project develops, things will change, both in the context and the design. These links can support propagation of change to all affected parts of both the design and context descriptions if the semantics of the link are made clear. If a piece of context changes, the links will identify all those elements of the design that require further investigation. Then by examining each link, understanding the consequences of the link and taking the appropriate action, designers can be sure that the design they are creating is kept up to date with the evolving contextual information.

In summary, this stage of the framework suggests that designers create and specify abstract and/or high level design ideas and that these ideas should be linked to the contextual information in some way. Specifying a design using a notation or method that allows incomplete specification lets designers focus on higher level details such as goals and tasks rather than on lower level details such as the placement of buttons, command names etc. At this stage, designers should be concerned with designing the right product. They need to ensure that they design a product that users will find useful, and not just easy to use. There is evidence to suggest that creating abstract or rough designs, is a valuable aid to the whole design process. Often, rough or abstract descriptions of a design can be ambiguous. Herbert (1993) has shown that this ambiguity can be a rich source of ideas, since designers are forced to think about what is being represented when attempting to resolve the ambiguity. Salomon (1990) and Wong (1992) have both shown that rough prototypes which are obviously incompletely specified provoke discussions about what is missing in the specification. When combined with links between context and design ideas, designers can determine if refinements to the specification will produce the 'right product'.

Since designers will be creating incomplete design ideas and using these to drive further contextual analysis, the whole context analysis and idea specification process should be iterative. Designers should not worry about specifying an idea completely in the first pass. Rather they should specify what they can from their knowledge of context, then use the gaps in the specification to identify what more they need to know about the context. This new information gained can then be used to fill out the specifications. Even though the first stage in this process is concerned with developing a deep understanding, designers cannot be expected to understand fully the contextual information before thinking about possible designs. The very act of designing makes some aspects of contextual information more important to understand than others (e.g., the information that supports the particular design idea). It cannot be possible to predict all the important aspects of the contextual information before designing begins (e.g., the startup dialog box described in Chapter 3 made the way that the secretary identifies applications relevant, *after* the decision had been taken to design the dialog box.)

15.2.4 Making the Abstract Concrete

In Chapter 3, there were difficulties described when Saskia Koehler, an M.Sc student, attempted to develop the abstract design ideas and specifications into a concrete design.

Some problems related to the lack of contextual information that could be used to inform the layout of screens, names of commands, position of controls etc. These problems identified the need for a stage of design that uses contextual information to inform *how* a design will work as opposed to informing *what* the design will do and support (i.e., previous stage of this framework).

In this last stage of the framework, designers fill in details. In the previous stage, designers specified a number of ideas in a fairly high level, possibly abstract manner. These will have specified the ways that users' work is to be supported. For example, in Chapter 3 one of the design ideas that was specified was that of automatically creating letters for applicants at appropriate stages in the application process. This feature supports that aspect of the users' work that is concerned with providing applicants feedback about their application.

In this section of the framework, designers take such ideas and make them concrete, describing *how* that functionality will be accessed. This will involve designers deciding upon issues such as screen layout, command names, icons etc.

Designers should make good use of contextual information at this stage to decide how the various functions will be accessed. Hence contextual information may be used to suggest command names, to validate particular screen layouts etc. Just as in the previous level, QOC diagrams or some other form of design space analysis may be appropriate. For example, in deciding how to access a function, designers may have to choose between providing access to that function via a pull-down menu or via a button on the screen. Designers can validate their choice through consideration of various contextual factors, such as the number of times users will access that functionality, the number of likely interruptions while accessing the function etc. Each factor will influence the way that the function is accessed. And, just as in the previous stage, these relationships between context and the design should be made explicit, since whenever any changes in the context occur, designers (or maintainers) will need to know what elements of the design were influenced by that piece of context.

Every aspect of a design need not be developed through a design space analysis. For example, there may be certain features of a design for which there can be no acceptable alternatives, such as the particular menu names and order for a Macintosh or Windows' 95 based application (i.e., File, Edit, etc.). If the contextual information indicates that the design will be implemented on a Macintosh then there is no use in deciding where the 'Edit' menu should be placed, and indeed if there should even be an edit menu, since the Apple Interface Guidelines (Apple Computer Inc., 1992) explicitly state the required position for such a menu. Hence, particular aspects of a design may be directly linked to the contextual information, and need not always be linked via a particular design rationale or design space analysis.

The linking tool described in Chapter 3 could be used at this stage as well as in the previous stage, as long as the context and design representations are sufficient to represent the detailed descriptions of context and the low level design decisions being taken. Other tools could also be used to support this stage of the framework. For example, Systematic Creativity, a method developed by Salvador and Scholtz (1996) can make use of contextual information in the form of user goals and product goals to inform the layout of various interface controls.

Both this and the preceding stage of the framework are very similar and it is likely that in practice, the third stage will merge into the fourth and the separation between them will be blurred. However, it is important to be aware of the distinction. In the third stage, designers are identifying *what* the system should do. They are investigating the context to design a system that fits into this context in terms of the support it provides for the work that users want to do. In the fourth stage, designers are investigating the context to identify *how* the system should support the functions identified in the previous section. Most other methods for using contextual information in design tend to focus on one of these two aspects. For example, Rosson and Carroll (1995) tend to focus designers on the lower level aspects, the how rather than the what, since they use scenarios to identify objects that should be implemented in an object–oriented system. The method does not suggest means of identifying whether or not the objects are the objects that are important to the users and whether or not they represent important aspects of users' work. On the other hand, the Contextual Inquiry and Customer Centred Design methods tend to focus on the higher level aspects by producing quite detailed models of the users' work but very little in the way of descriptions of how that work will be supported. Hence it is important that designers are aware that contextual information can be used, and indeed should be used, to determine both *what* the system will support and *how* it will support it.

15.2.5 Summary

Figure 42 summarises the framework.

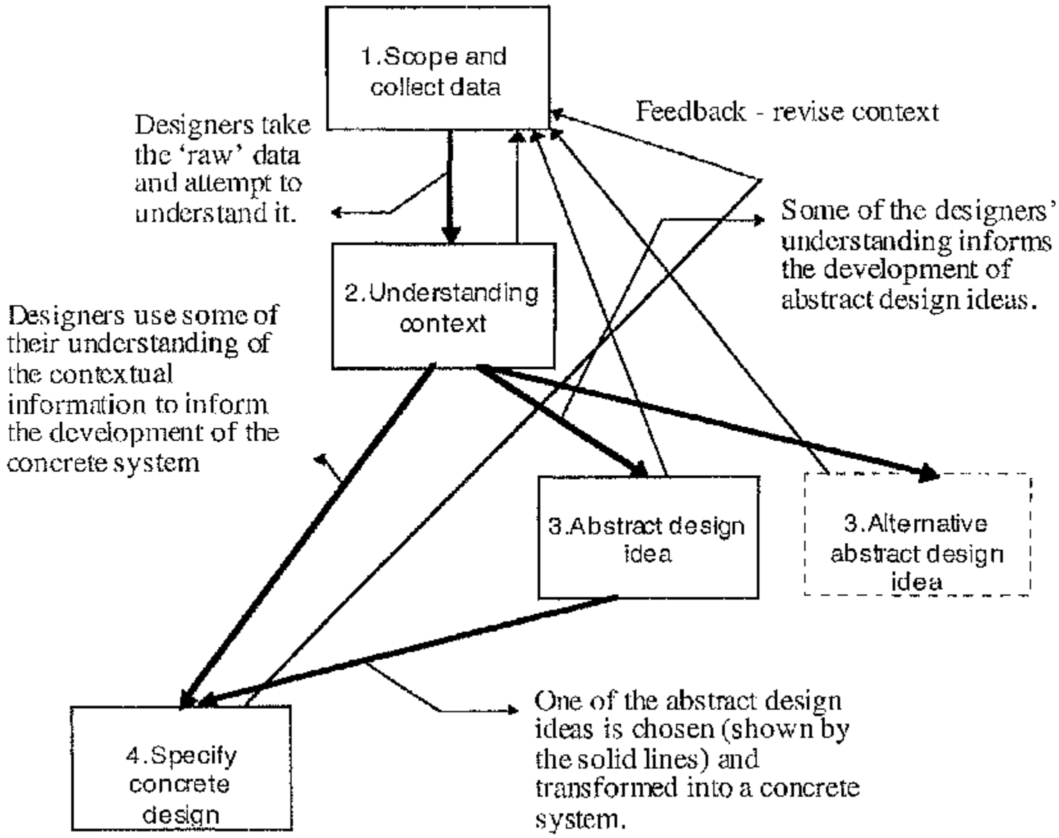


Figure 42. Four stage framework for contextual design

The process begins at the top of the diagram, where the scope of the relevant contextual information is decided upon. Designers start to develop their understanding of the contextual data in the second box. Once a good understanding has been achieved, the process then moves on to the third stage, specifying design ideas abstractly or roughly. In the diagram above, two ideas are shown in the boxes on the right hand side of the diagram. Lines from understanding the context to these design ideas represent the ways in which the context has influenced the ideas. Lastly, one idea is chosen (in the diagram above this idea is drawn with a solid line) and specified concretely. The concrete specification is represented by the lower box. Note that as well as input from the abstract design idea, the concrete specification also takes input directly from the contextual information, since there will be information here that can be used at the concrete level rather than the abstract (e.g., the platform upon which a system will be implemented.)

Note that boundaries between stages are not definite. In most cases, designers will gradually move from one stage to the next. The differences between the stages will not be as marked in practice as they sound in theory. This is because in design, there is no definite end to activities. Designers could spend forever understanding the contextual information or specifying abstract design ideas. In practice, the constraints of the particular project will determine when designers move on. Even so, designers will often need to return to a previous stage, to develop more understanding of a particular area of the contextual information, or to return to a part of a design idea.

15.3 Illuminating the Framework

This section provides an imaginary example of how the framework could be of benefit to designers. It uses the situation described at Intel in Appendix F and attempts to show how the framework and the tool support described in Chapters 3 and 5 could have helped the designers avoid some of the problems that were described in Appendix F.

The reason that the prototype development group addressed only a small subset of the issues modelled by the ethnographers was that they were unable to fully understand the model and appreciate how different parts of the model interacted with each other (e.g., social aspects of families with young children interact with the communication aspects). Had this understanding been better, the developers may have found it easier to relate what they were doing, not just to certain parts of the model, but to the whole model. They would be able to predict how the prototype would fit into the model as a whole. However, generating a deep understanding of the model is not sufficient. The other problem that the prototype developers faced in attempting to relate their prototype to the model was that it was difficult to validate their design ideas against the model. They tried to validate the prototype and see if it conflicted with the model or if anything more could be gained from the model, but this was difficult since the relationships between the model and the prototype were not clear and their understanding of the model was incomplete. One of the main contributions that the framework offers designers is its focus on making explicit the relationships between contextual information and design. By doing so, designers can see how their design fits in with the context.

The framework described above suggests that after data collection members of the design team *and* those involved in data collection spend some time to generate a common, shared understanding of the data. Consensus must be reached amongst members of the team about the important aspects of the data that should be addressed by prospective designs and design ideas. The framework stresses that the data collectors must be actively involved in this shared understanding process since it is unlikely that they fully understand the data or its implications. Much of this understanding will be developed through conversations with the designers, when they start to question the data. By actively taking part in conversations about the model through GENEVA, the technical concepts team and the ethnographers would have been able to discuss what they felt were the important aspects of the model in their own time. They would not have had to rely on chance meetings, short irregular phone calls, etc. Everyone involved would be able to take part in the discussions since the discussions are public instead of private (i.e., phone calls, chance conversations). Hence individuals would not have had to take time to explain their own understanding of the model to others as was the case with one member of the technical concepts team who compiled five video clips to show what he felt were the important points from the initial talk given by the ethnographers. Detailed discussions about aspects of the model would progress as the understanding of the models progressed. The technical concepts team would not be faced with a detailed description of the model to begin with. The details would emerge as the conversations about the model progressed.

There are other ways to try to improve the understanding of contextual information. Both the ethnographers and the technical concepts team could have been merged into one group forcing this new group to perform both data collection and idea development together. In this case communication difficulties would not be as severe, since everyone would be involved in all aspects of data collection and idea development and would have intimate knowledge of both the data and the ideas. However, as Nardi (1995) points out, most commercial organisations do not have the resources to allow such a situation. Forming a larger group in this way would have meant that investigation into further markets would be delayed until the investigation and idea development from the first market was completed. This increases the amount of time it would take the company to develop their ideas and the time it would take for them to get a product developed and into the market, giving the company's competitors an advantage. Another solution might have been to assign investigation and idea development for each different market to separate groups. However, it is unlikely that a group of people could be collected together such that they all had the same high level of skills in performing ethnographic analyses and developing prototypes as well as the necessary time. Hence, in this situation, the solution offered by the framework, that of utilising tool support to develop a shared understanding of the model would seem the most appropriate.

Once a sufficient understanding of the data and associated models has been reached, the framework suggests that designers identify design ideas and relate these ideas explicitly to the contextual data or models. This can only be done when a common understanding has been reached amongst all team members and when that understanding is developed enough for everyone to identify and agree upon the important aspects of the data or models. When these conditions hold, it is easier for designers to relate their designs to the models and to validate their designs with respect to the models.

By making the relationships between context and design explicit, as the framework stresses, designers are also in a better position to identify potential areas of further work. At Intel, the technical concepts team were only able to focus on the communications aspect of the model which resulted in them developing a prototype of a system that supported various kinds of communication that the ethnographic description of families with young children had suggested would be useful, but which did not support other aspects of the model such as the temporal sequencing of activities for a typical family with young children. The ethnographers were dissappointed that more of the model had not been used. Had the relationships between the prototype and the model been made explicit, it may have been clearer to the developers that there were large parts of the model that had not been used. Identifying these unused parts of the model might have encouraged the developers to think about incorporating these parts of the model in the prototype.

This example illustrates the potential benefits that the framework offers designers. It encourages a shared understanding of contextual data that makes the important points about the model clear and could indicate how product ideas could be developed that take account of all the important aspects of the model. By making the relationships between context and design explicit, designers can validate their design ideas and determine how well they fit the context. They are also able to determine which parts of the model have been used to inform the design of the product idea(s)

However, it is clear that simply stressing the importance of understanding and explicitly recording context/design relationships is not sufficient to enable designers to make more efficient use of contextual information in design. For example, without adequate tool support it would still be difficult to generate a deep and shared understanding of the model amongst all members of the design team, since the group involved in the data collection spent most of the time after presenting the results of the first analysis off–site. The main difficulty here in developing an understanding was in getting all the team members together to share in the discussions. This had a knock–on effect on the abilities of the designers to relate their prototype to the model, since their understanding of the model was not complete enough to enable them to do so. However, even if the ethnographers had not been off–site and had been constantly available to help develop the necessary understanding, maintaining the record of the relationships between the model and the prototype would be difficult. Many relationships could be created between the model and the prototype and it would be difficult to keep track of them as the model and prototype evolved. Hence some form of support for the framework is necessary, one that enables members of a design group to share in developing an understanding of a model, even if they cannot physically be in the same location at the same time, and one that maintains the set of relationships between context and design. Two such forms of computer based tool support were described in Chapters 3 and Appendix F.

15.4 Chapter Summary

This chapter has described a four stage framework for making effective use of contextual information in design. The first stage involves defining the scope of relevant contextual information and collecting the information. The next stage involves generating a sufficient understanding of the data, an understanding that is shared and common across all members of the design team. Once such an understanding has been developed, the third stage involves developing a set of design ideas based upon the contextual data and the understandings gained in the previous stage. The crucial point in this stage is that the relationships between design ideas and contextual information should be made explicit such that assessments of the usage of contextual information can then be made, e.g., in terms of consistency, appropriateness etc. The final stage involves developing one of these ideas into a concrete design, again making the relationships between the contextual information and the design explicit.

The framework was used to discover potential reasons for the lack of successful use of contextual information in a real life design project. It also offered suggestions as to how the process could be changed to make more efficient use of contextual information. The framework alone is not sufficient to increase the efficiency with which designers make use of contextual information. Computer based tool support is required. Two tools were described in Chapters 3 and Appendix F which support both the generation of an understanding of contextual information and the recording of the relationships between context and design. Chapter 4 described three studies of the usage of LD in which the tool was informally validated and shown to provide some benefit. Appendix F also indicated that LD was potentially useful by describing the reactions of designers at Intel to paper mock–ups of the tool which were demonstrated to them.

16. Appendix H Further Work on Geneva and the Framework

16.1 Evaluating GENEVA

GENEVA, described in Appendix F, has not been used or evaluated in any way, since the first stage of development of the tool was completed towards the end of my internship at Intel. There was no time to perform any kind of evaluation with the designers at Intel who were the prospective users of the tool, other than demonstrating the tool to them.

As with the evaluation process of LD, the first stage of evaluation should be a formative evaluation of GENEVA. Such an evaluation will identify parts of GENEVA that need to be improved, in the same way that the formative evaluation of LD identified areas of improvement for it. Due to the nature of GENEVA (it is a multi–user tool, operating over a network) evaluations of the tool will be more complex than the evaluations of LD. For example, users' experiences with GENEVA in a multi–user setting will have to be recorded. This entails recruiting a group of participants for the evaluations, and asking them to use GENEVA as a group, over a certain period of time and recording their experiences. This is in contrast to the single–user evaluations that were performed using LD.

As a result of the formative evaluations, improvements will be identified for, and made to, GENEVA. A detailed investigation should then be carried out on the way that GENEVA is used. Again, the improved version of GENEVA should be made available to a group of designers. They should be asked to use the tool throughout a design project to discuss various design issues related to their project. Some of the issues of interest for the investigation are

- *Who should have rights to create hotspots*? In the current version of GENEVA, only a subset of people in the design group are able to create hotspots over the graphical model under discussion. The consequences of this constraint should be investigated. Perhaps such constraints are only feasible when a subsection of the design group were responsible for data collection and creation of the graphical model (presumably they know where the areas of interest are in the model, at least to begin with). Perhaps over time, as all members of the group become engaged in discussions, the constraint should be lifted, since all members of the group may understand the model enough to be able to point to areas of interest in it.

- *What are the attitudes towards recording discussions in GENEVA?* The investigations need to discover if designers are willing to record discussions using GENEVA, and what the implications are for such records. *What types of group will be more willing to record their discussions in this manner? What will the discussions be used for?* Designers may be reluctant to put their thoughts down in writing, for fear of exposing their lack of understanding (or for fear of being held responsible for saying something). By recording discussions relating to understanding, designers and managers now have access to the roots of some design decisions, particularly if discussions are related to subsequent design decisions. For example, one designer may make some false claim during a discussion which leads to a wrong design decision being made later on. If the working culture is such that the group or management need to find someone to blame for such a mistake, designers will most probably be very careful about what they say, if it can be recorded and used against them at a later date.
- *At what point do discussions turn from understanding the graphical model to using that understanding in design?* There have been a number of investigations into the design process (Guindon, 1990; Curtis *et al.*, 1988; Rosson *et al.*, 1988). Opportunistic design strategies have been described whereby designers attempt to use their understanding of some particular aspect of the design problem to develop part of the design solution. Developing part of the solution can help designers understand more of the whole design problem, since they have to relate the solution back to the original problem. Hence, designers do not typically attempt to understand all of the design problem at once and then go on to using this understanding in developing a solution. Rather, the process is iterative, going back and forth between developing an understanding and developing a solution. It is important to identify the consequences for GENEVA of such a design process. Perhaps discussion topics need to be linked to design solutions, or highlighted, to show that they have been used to influence part of the design solution.

16.2 Developing and evaluating the four stage framework of contextual design

The four stage framework of contextual design described in Chapter 6 has not been used in a design project. The framework was developed as a result of the studies of designers at Intel and as a result of the IT case study described in Chapter 3. There are therefore a number of fundamental and practical issues that need to be addressed. These are listed below.

- *Does the framework cover all aspects of the design process?* If designers carry out some tasks which do not easily fit into the four stage framework but which the designers feel is necessary, the framework should be extended or modified to account for these tasks. For example, the framework does not explicitly describe a stage at which the design produced by the design team should be evaluated. This is primarily because the third and fourth stages of the framework (developing an abstract design idea and developing a concrete design) incorporate aspects of testing and evaluation within them. However, the testing performed at these stages (i.e., with respect to fit to context) may not be adequate for a commercial design project. User testing needs to be performed, but it is not clear if this should be considered a separate stage in the framework, or if it should be incorporated into one or more of the existing stages of the framework.
- *Are some stages of the framework unneccessary? Do designers see the need to carry out each of the four stages of the framework?* In particular, the third stage of the framework requires attention, since developing multiple abstract ideas in order to compare each idea in terms of their fit to context, is likely to be fairly novel to most designers. Some designers may consider this a waste of time, and indeed in some cases, particularly when designers are working under tight deadlines, it may well be.
- *Can some of the stages be better supported?* Following on from the previous issue, one way that designers could gain more benefit from each stage is by providing better support for each stage. For example, in the third stage, providing libraries of abstract design solutions could help designers to choose and describe alternative design ideas in a shorter space of time.

How successful is the tool support for the framework? Currently, LD and GENEVA are completely separate. It is likely that some kind of communication between both tools is required, since discussions recorded in GENEVA are likely to feed into design solutions described in LD. Some form of linking between the tools could be required, or both tools could be integrated into one tool. The merits of each approach, and whether or not such an approach is necessary, requires investigation.