**Published in: Byerley, P.F., Barnard, P.J., & May, J. (eds)** ***Computers, Communication and Usability: Design issues, research and methods for integrated services, pp. 197-219.***
**(North Holland Series in Telecommunication) Elsevier: Amsterdam (1993)**

*Chapter 2.4*

# Developing the Design Space with Design Space Analysis

Allan MacLean†,Victoria Bellotti† and Simon Shum*

†Rank Xerox EuroPARC, 61 Regent Street, Cambridge, UK — CB2 1AB

* Department of Psychology, University of York, Heslington, York, YO1 5DD, UK

## 1. Introduction

This chapter reports work carried out as part of the AMODEUS project (Esprit BRA 3066). The goal of the project is to develop interdisciplinary approaches to studying human-computer interaction and to move towards applying the results to the practicalities of design. We describe one of the approaches taken to represent design — Design Space Analysis. One of its goals is to help us bridge from relatively theoretical concerns to the practicalities of design. Design Space Analysis is a central component of a framework for representing the design rationale for designed artifacts. Our current work focuses more specifically on the design of user interfaces. A Design Space Analysis is represented using the QOC notation, which consists of Questions identifying key design issues, Options providing possible answers to the Questions, and Criteria for assessing and comparing the Options. In this chapter, we give an overview of the Design Space Analysis approach and describe a model of the design process which helps guide its use.

Amodeus (Assimilating Models of Design, Users and Systems) is a project which brings together teams from different disciplines with three general objectives (1) to extend the scope of modelling techniques to provide analytic leverage on the problems of user-system interaction: (2) to bridge the conceptual gaps between behavioural and computing disciplines: and (3) to bridge from theory to the practicalities of designing software artifacts. The first two objectives have been addressed by the previous chapters in this section; this chapter reports work aimed at supporting the third objective. It has two goals. The first is to develop a technique for representing design decisions which will, even on its own, support and augment design practice. The second goal is to use the framework as a vehicle for communicating and contextualising more analytic approaches to user-system interaction into the practicalities of design. We have made significant conceptual progress towards both of these goals in the course of the Amodeus project. However, an important pre-requisite for their application is to understand how the basic approach can actually be used in practical settings. That is the main focus of this chapter. We describe a model of the design process which acts as the basis for a simple method to support the systematic development of a design space representation. We continue by first describing the Design Space Analysis framework and the QOC notation. We go on to describe the process model. We then give a more concrete illustration of the process model in use by showing how a design space is developed starting from a description of a design problem. We conclude by discussing how the broader range of approaches represented within Amodeus are likely to fit into this model and the further benefits which that promises.

## 1.1 Design Space Analysis

Design Space Analysis is an approach to representing design rationale (MacLean et al., 1989, 1991a). It is essentially an argumentation-based approach to design (see Shum and Hammond, 1993). It is a central part of a long term project in which we are interested in helping software designers to reason about design (individually and in groups) and to produce an output which can help others understand why the resulting design is the way it is. Although our intent is to develop a framework for design applicable to a variety of domains, much of our work to date has been aimed at HCI audiences for three main reasons. First, the design domain in which we are most actively working is user interface design. Secondly, we believe that HCI expertise is a crucial ingredient in developing the approach itself as the usability of the tools and techniques we produce will ultimately be critical in determining the acceptability of our work to its users—i.e. the designers of computer systems. Third, and perhaps most importantly, HCI design is much more informal, more open-ended and more poorly understood than some other domains. Reasoning is not always clear and developing the 'knowledge base' by cumulating design rationale promises to be a powerful way to compile contextualised design knowledge.

A key characteristic of our approach is that the output of design is conceived of as a design space rather than a single artifact. The approach therefore contrasts with the traditional conception of design which assumes that the end product is simply a specification or artifact. The final artifact, although embodying the designer's decisions, does not preserve any of the thinking and reasoning which went into its creation. We use a semi-formal notation (called QOC, for Questions, Options & Criteria) to represent the design space around an artifact being produced. This design space is an explicit representation of alternative design options, and the reasons for choosing among those options.

Figure 1 illustrates the relationships between the main concepts we use to represent the design space. Questions highlight key issues in the design space. Options can be thought of as 'answers' to Questions. Criteria are the reasons that argue for or against possible Options. We find that considerable power can be gained simply from regarding relationships between Options and Criteria as relatively positive (solid line between Option and Criterion) or negative (dotted line) — i.e. arguing for or against the Option. Options may spawn off consequent Questions which allow more detailed aspects of the design to be addressed.

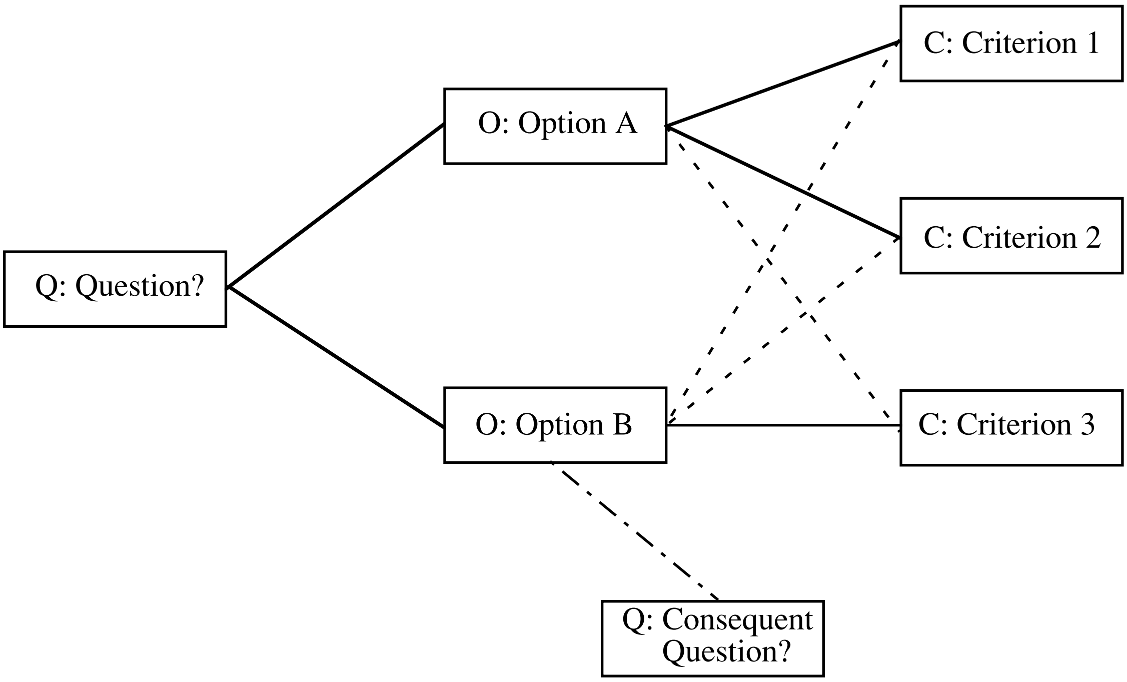

Figure 1. The components of a design space using the QOC notation

This representation of a design space provides a succinct rationale for the final design by placing it in a broader context which highlights how it might be different and why it is the way it is. Such a representation should be able to support communication between people with different backgrounds and goals, for example between members of a design team working on an initial design (see McKerlie and MacLean, 1993a, 1993b), between the original designers and designers of a later generation system who want to re-use parts of the original design, and even between the designers and users of a computer system.

Exploring some of these claims is still a future part of our research strategy. However, Design Space Analysis provides a framework for design which we have already found useful to help better understand design issues (MacLean et al., 1989, 1990, 1991a, b). This theoretical approach drives such activities as understanding the design process and how it can be improved (MacLean et al., 1990); requirements for tools to support the creation of a QOC representation (and thus the design process) (MacLean et al., 1991a); and the integration of other approaches, such as cognitive and system HCI modelling techniques, into software design (MacLean et al., 1991c).

Such motivations, arguments and examples are important for developing the kind of approach we are taking. However, to get practical advantage we need to understand how to actually *produce* a design space representation.

## 1.2 A model of the design process

As a way of articulating the process of carrying out Design Space Analysis, we have developed a model of the design process which focuses on gathering, organising and reasoning with design information. The model has two components. First, there is a sequential model which consists of five phases ranging from gathering and organising given information, through structuring it into a QOC design space, augmenting the design space and finally making decisions (see figure 2). The second component is a declarative model based on the use of design heuristics which focus on subparts of the design representation to assist the designer in developing the space (see figure 3).

*Phase 1:* Identify relevant information

Activities:
- Get a feel for the main issues
- Work out what information provided is relevant (& classify as Q, O, and C if possible)

*Phase 2:* Structure material into rough QOC

Activities:
- Structure and make sense of the information available
- Find good Questions

*Phase 3:* Flesh out design space

Activities:
- Use current understanding of design to help generate new ideas
- Generate new Options
- Generate new Criteria

*Phase 4:* Reformulate design space to tidy it up.

Activities:
- Tidy up description and make it more coherent
- Reword Q, O, C if necessary
- Reformulate Questions (and reorganise O, C) to improve decomposition

*Phase 5:* Make design decisions

Activities:
- Evaluate and select Options
- Use Criteria to evaluate Options
- Represent decisions by drawing a box around selected Options

Figure 2. The five phase sequential process model and the activities underlying each phase.

The goal of the sequential model in figure 2 is to structure and highlight the main activities. Clearly design is not carried out in the strictly sequential way implied by this model. This is taken account of in two ways. First, the phases in figure 2 should not be interpreted as a strictly serial model which requires one phase to be finished before the next one is started. For example, it may be that some decisions are clear well before phase 5. Nevertheless, there is some logic to the ordering of the phases. For example, beyond the obvious point that you have to have information before you can work with it, we would like to encourage (though not enforce) designers to delay making decisions until they have an adequate basis for making a good decision.

Secondly, we complement the sequential phases with a declarative model based on heuristics derived from the QOC framework. Figure 3 shows some of these heuristics. They focus on subpatterns within a QOC representation and suggest design "moves" which can be used to explore the design space. They encapsulate ways in which the QOC notation itself can provide leverage or 'cognitive scaffolding' for reasoning within a design space. Being declarative, these heuristics can be applied during any of the sequential phases and avoid the need for postulating extensive iteration between the phases. They therefore provide a mechanism for coping with the opportunistic aspects which we know are a characteristic feature of design behaviour (e.g. see Guindon, 1990). Their main role is to provide a mechanism for developing the design space in a systematic way. For example, *Use Options to Generate Questions* highlights focusing in on an explicit identification of the issue being addressed, which in turn helps to encourage the generation of alternative Options for its solution (*Use Questions to Generate Options*). *Represent Both Positive And Negative Criteria* encourages a balanced view of potential solutions and helps counter confirmation bias (Wason, 1968). If a pattern is observed of all Criteria which have been considered supporting a given Option, this heuristic encourages the designer to think about possible problems with the solution.

*Use Questions to Generate Options*

*Use Options to Generate Questions*

*Use Options to Generate Criteria*

*Consider Distinctive Options*

*Look for Novel Combinations of Options*

*Represent Both Positive And Negative Criteria*

*Overcome Negative, But Maintain Positive, Criteria*

*Design to a Set of Criteria*

*Search for Generic Questions*

Figure 3. Examples of heuristics which form the basis of the declarative component of the design model. (Based on MacLean et al., 1991a)

## 2. An example: Distributed multimedia services

Let us now look in more detail at the way in which a QOC design space is developed using the process model just presented. The example is taken from a 'design scenario' used in the Amodeus project. A design scenario is a description of a design situation and some of its surrounding context. The kinds of scenario we used could take a number of forms, corresponding to typical design situations. For example, they may highlight a particular problem to be solved, may ask for advice on a particular problem, may ask for assistance in choosing the best of a number of possible solutions, or may simply present a design solution to be critiqued. Such scenarios served as a common focus to allow the variety of approaches represented in Amodeus to be exercised on the same material to enable comparisons to be made among them. The set of scenarios ensured reasonable sampling over a variety of different situations.

### 2.1 The initial design problem

The particular example used here comes from the 'production interface' developed as part of the RACE project GUIDANCE (further details about the nature and development of this design can be found in chapter 3.3). The starting point is from a situation in which two different designs for providing information about users and work objects in a distributed multi-media CSCW system are being considered. The objects are displayed in a desktop style in both alternative designs, providing information about their type and usage by other people in the system. The main contrast between the designs is that one has people represented in the main display along with the various work objects, and the other has people displayed in a separate window from the work objects. We will refer to these as design A and design B respectively. An important issue which has been identified is the need for users to be able to find out what objects other people are currently working on. The design scenario produced from this situation is reproduced in Appendix A. It provides details about the nature of the design project, its current state, and some of the surrounding context.

The designers are looking for advice on choosing between the alternative designs or suggestions for improving on them. They are also concerned with the implications of dealing with a larger number of work objects or people than can comfortably fit on the screen and would like to know how scrollable displays would affect the situation. They are open to the possibility that the issue identified may not be the best way of looking at the problem.

## 2.2 Developing the design space

The approach taken by Design Space Analysis in this kind of situation is first of all to identify the relevant information which is available and try to understand its implications by organising it into a design space using the QOC notation. This then acts as the basis for further reasoning over the available information to expand the design space. The models presented in figure 2 and figure 3 above help guide this process. The first step is to identify the relevant information which is known, not necessarily worrying about classifying it in terms of QOC at this stage.

### *2.2.1 Phase 1: Organise initial information*

The analysis was begun by looking at the people, objects and tasks involved in the scenario. The work objects, users and their tasks are the same for both design alternatives in the scenario, but some of the more detailed system operations are different. They can be summarised as follows (see Appendix A, Figure 8 and Figure 9 for more details):

*People and Objects from the Scenario (the same for designs A and B)*

Users are represented by name-labelled icons showing self (in black) current users (in white) and non-current users (in grey).

System/work objects are represented by icons which show their type including: documents, records, text, video snippets, audio, stills, and music.

The purpose of the system is to facilitate collaboration among users working on the preparation of multi-media documents. A variety of specific tasks can be identified from the description given.

*Users' Tasks from the scenario (the same for designs A and B*

Co-operative document planning (several users contribute to the same document(s) over time).

Editing and assembling of existing materials using integrated editor functions.

Communication directly between system users.

We summarise these three classes of tasks as:

An individual or group working on multiple objects to try to pull them together
Several people working on one object
Users communicating with each other directly

This may be viewed as communication between people, mediated through more than one, one, or no work objects.

When we look in more detail at how these tasks are carried out with the alternative interfaces, we now see differences between a and b. We can identify three specific activities which are necessary for carrying out the classes of task highlighted in the previous section and which are instantiated differently. These are:

*Finding another user*
Design A displays all users in the same area with the work objects.
Design B displays all users in a separate list (the ordering scheme is not clear).

*Finding out what work objects a person is using*
Design A displays the user inside rounded “busy-boxes.”
Design B does not do this directly. The viewer has to select the user from the list to see what objects that person is using.

*Identifying who is using a “busy” work object*
Design A displays an icon representing the person inside the busy-box of each object they are using.
Design B requires you to select a user in the list and then puts busy-boxes around each work object that person is using.

## *2.2.2 Phase 2: Produce rough QOC*

The next step is to represent the information, which phase 1 identified as being important, in terms of QOC. It should be emphasised that the choice of what to represent in QOC is strongly driven by the nature of the design problem. In the present case, the combination of the collaborative tasks to be supported and the contrasting design solutions between the two proposed designs define the focus we take. The pairs of solutions are contrasting Options in QOC terms. The different activities which we identified highlight dimensions along which the Options differ, and therefore form the basis of Questions. The resulting QOC representation is shown in figure 4. In this case, suitable Questions are fairly easy to derive, at least in part because having directly comparable alternatives clarifies the nature of the dimension along which they relate. The reason for not asking Questions about how to support the users’ tasks in a more general sense is that this would expand the design space away from the focus which best contrasts the alternative designs.

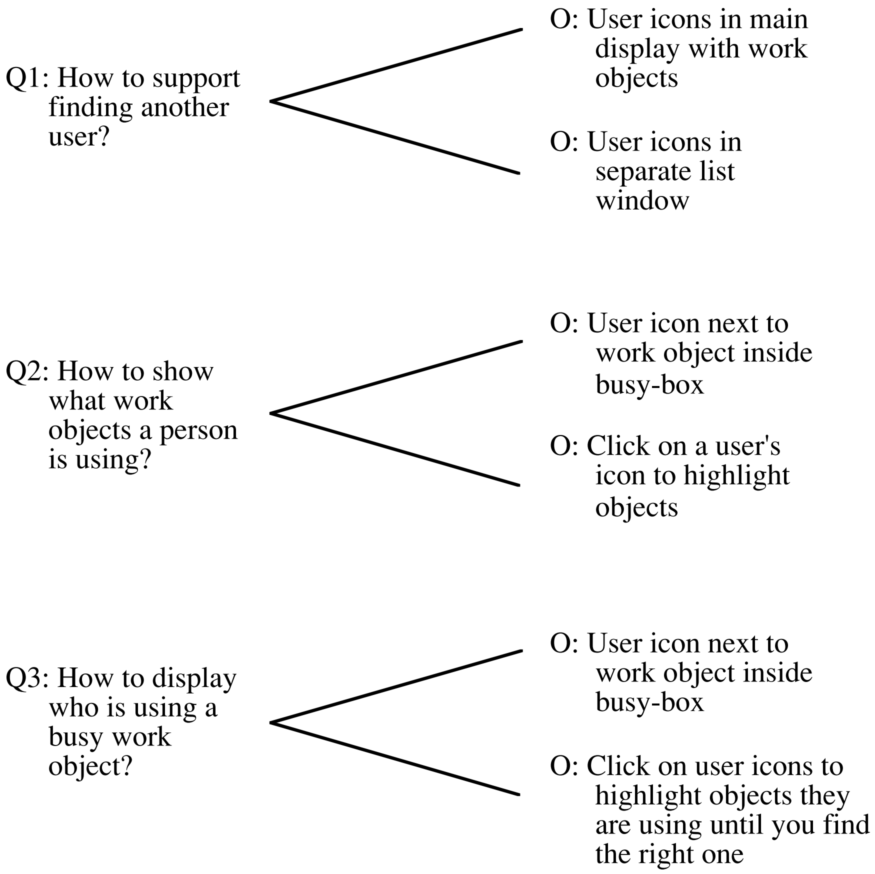
Q1: How to support finding another user?
O: User icons in main display with work objects
O: User icons in separate list window
Q2: How to show what work objects a person is using?
O: User icon next to work object inside busy-box
O: Click on a user's icon to highlight objects
Q3: How to display who is using a busy work object?
O: User icon next to work object inside busy-box
O: Click on user icons to highlight objects they are using until you find the right one

Figure 4. Initial QOC Questions and Options derived directly from the information given in the scenario. The upper Option for each Question is the solution found in design A and the lower Option is from design B.

It should be noted that in general identifying Questions can be one of the most difficult activities in producing QOC (see Bellotti et al. 1991; McKerlie and MacLean, 1993a; Shum (in press). The heuristic *use Options to generate Questions* is aimed towards helping in this situation. The idea is to encourage the designer to identify what is important about a particular Option—for example an aspect which is thought to be problematic for some reason.

### *2.2.3 Phase 3: Expand the design space*

Now that we have a basic design space, the next step is to use the focus it gives us to reason beyond the information available in the description of the design problem. It was interesting to note that no explicit Criteria were given in the scenario description to help us understand why particular options might be good or bad. Finding some of the relevant Criteria is therefore the first thing we need to do for evaluating the alternative solutions.

### 2.2.3.1 Finding Criteria

Our analysis in phases 1 and 2, which began structuring some of the important issues at stake, leads to the generation of Criteria which, in this case, basically involves a process of going through the scenario designs and critiquing them. Three heuristics are relevant for this activity: that of *using Options to generate Criteria*, i.e. looking for advantages and disadvantages of Options which are being examined; *representing positive and negative Criteria* which aims to ensure that both advantages and disadvantages of Options are considered; and *designing to a set of Criteria* which encourages Criteria relevant for one Option to be considered for others (More generally it is used to help identify a set of Criteria which are globally important for a given design, but that will not emerge with a small example like this one). We will now consider each pair of Options and their implications to see what Criteria we can identify.

**Q1: How to support finding another user?**

The Option of putting *user icons in main display with work objects*, which represents design A, has *high information content permanently in the display*. This could be useful, for example, if the person looking for the other user also knows which object that person is working on. However, the alternative Option, *user icons in separate window*, representing design B, gets more support because this display structure better *supports an organised serial search* (because it is clear where to look for user icons—although it is not clear exactly how they are organised), there is *less display clutter* with only one icon per user (it is assumed that users of multiple files have multiple icons in A). In addition, a dedicated display of user icons means that the user does not have to search through other icons as well which makes this more *supportive of task relevant information for finding another user*.

**Q2: How to display what work objects a person is using?**
(Assuming person has been found)

The Option *user icon next to work object inside busy-box* (design A) is supported by a Criterion of *few steps to get information from the display*. Even this may be called into doubt if the person is using many objects (hence having multiple icons) and the display is very cluttered since visual search would have to include reading the name label of every active user icon, and would be prone to omission and repetition of inspection. In contrast, the Option *click on a user's icon to highlight objects* (design B) *supports task relevant information about a particular user*, i.e., what objects they are using. It also generates *less display clutter* than design A, which means that more can be fitted on the computer screen so less scrolling should be necessary. If we were to refine our analysis further in this direction, we might suggest ways of getting all the objects a person is using on to the screen when the user clicks on their name. This could require redisplaying the objects in another window, or making them move onto the existing desktop display.

**Q3: How to display who is using a busy work object?**
(Assuming object has been found)

In this case the design A Option, *user icon next to work object inside busy-box*, is supported by three Criteria of *low user memory demands* (because the user need not remember the name of the work object), *few steps to get information from the display* (because they only have to look, unless they need to scroll the desktop display) and *support task relevant information on object's status* (because all the information they need is immediately accessible). The design B Option, *click on user icons to highlight objects they are using until you find the right one*, comes off very badly in this assessment because the user has to remember the object while clicking on each of the user icons, which requires many mouse keypresses. Task relevant information is not supported at all. The only Criterion favouring this Option is that of *low display clutter*. Note that the same Option from A appears under Q2 and Q3 and the Option from B is also similar in both cases. This shows that the same solution may be a candidate for different design Questions, which, everything else being equal, might be a reason for favouring it. However, the analysis presented here suggests that everything is not equal, and that design B which was preferable for the other tasks is a failure here. Trying to overload the solution to cover a variety of tasks does not work in this case.

These Criteria and their assessments against the Options are summarised in the QOC diagram in Figure 5. The overall pattern is that design B is claimed to be better on consideration of screen clutter. It also wins on the main user-oriented task of displaying what work objects a person is using because the person is easier to find, and because a person may be using more than one work object. Although the structure of the display does not actively support the search for the objects, it is likely to be easy to pick out the rounded boxes which provide at a glance the task relevant information required. Design A, on the other hand, is much better on the work object oriented task of identifying an object's current user because, once the object has been found, the relevant user information is immediately visible, whereas Design B loses heavily because it requires a highly time consuming search. This failing seems more serious than the two failures with design A, as long as the desktop display area is not very large. If the display area does become large then design A begins to fare much worse on both of its weak points.

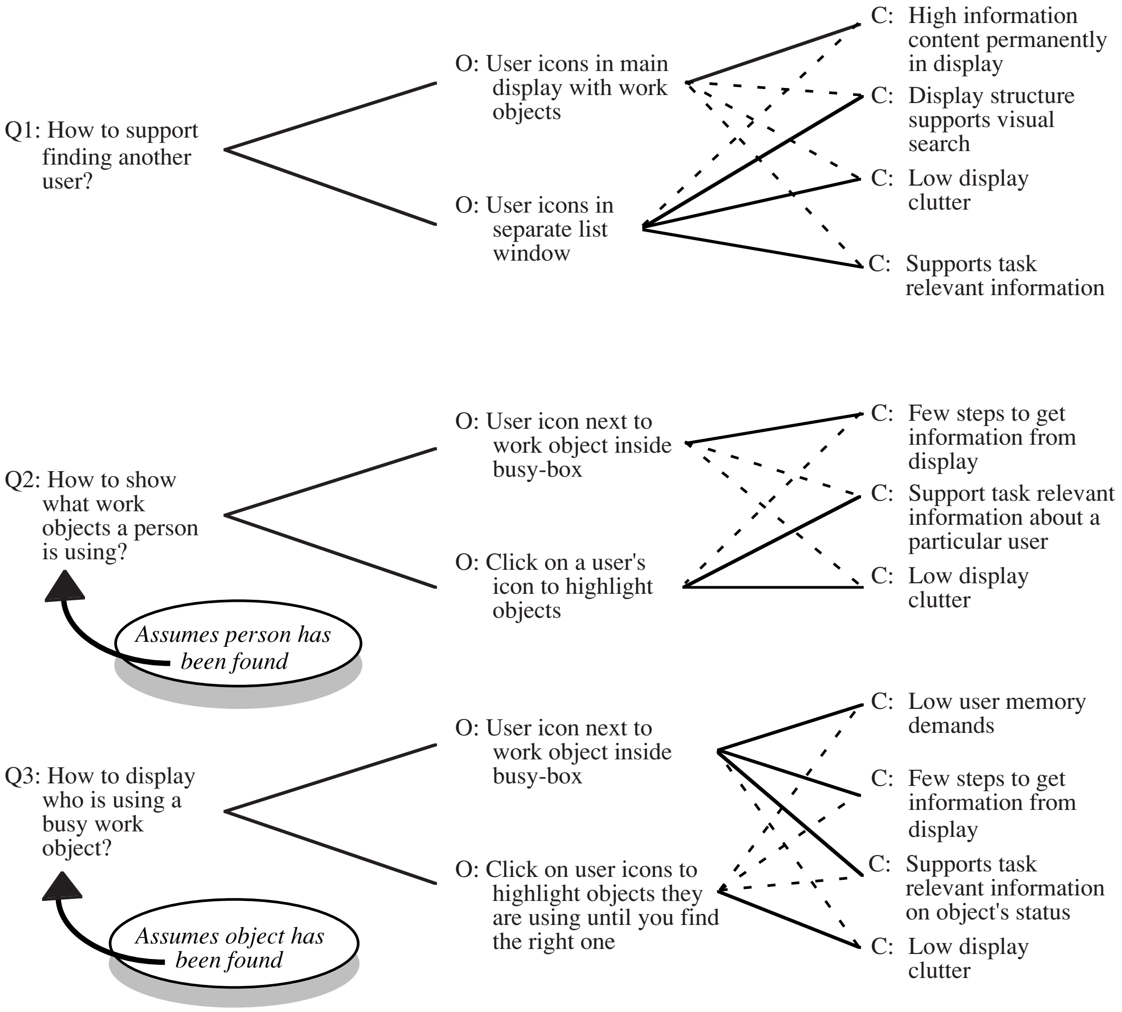
Q1: How to support finding another user?
O: User icons in main display with work objects
O: User icons in separate list window
C: High information content permanently in display
C: Display structure supports visual search
C: Low display clutter
C: Supports task relevant information
Q2: How to show what work objects a person is using?
Assumes person has been found
O: User icon next to work object inside busy-box
O: Click on a user's icon to highlight objects
C: Few steps to get information from display
C: Support task relevant information about a particular user
C: Low display clutter
Q3: How to display who is using a busy work object?
Assumes object has been found
O: User icon next to work object inside busy-box
O: Click on user icons to highlight objects they are using until you find the right one
C: Low user memory demands
C: Few steps to get information from display
C: Supports task relevant information on object's status
C: Low display clutter

Figure 5. QOC design space with Criteria for and against the Options.

### *2.2.3.2 Finding New Options*

It is clear that neither design provides a satisfactory solution for all three Questions, particularly if the desktop display area becomes larger and more cluttered. The next stage is to look for further Options which might lead to a better design. The Criteria just identified are important in this process. The heuristics which help here are *overcome negative, but maintain positive, Criteria* and *use Questions to generate Options*.

For the Question *how to support finding another user*, the *icons in a separate window* was the favoured Option, but although it was better than the alternative it was fairly weak on the Criterion about the *display structure supporting visual search*. One way of improving the Option would be to impose more structure on the user icons—for example placing them in alphabetical order. This new solution is represented in figure 6.

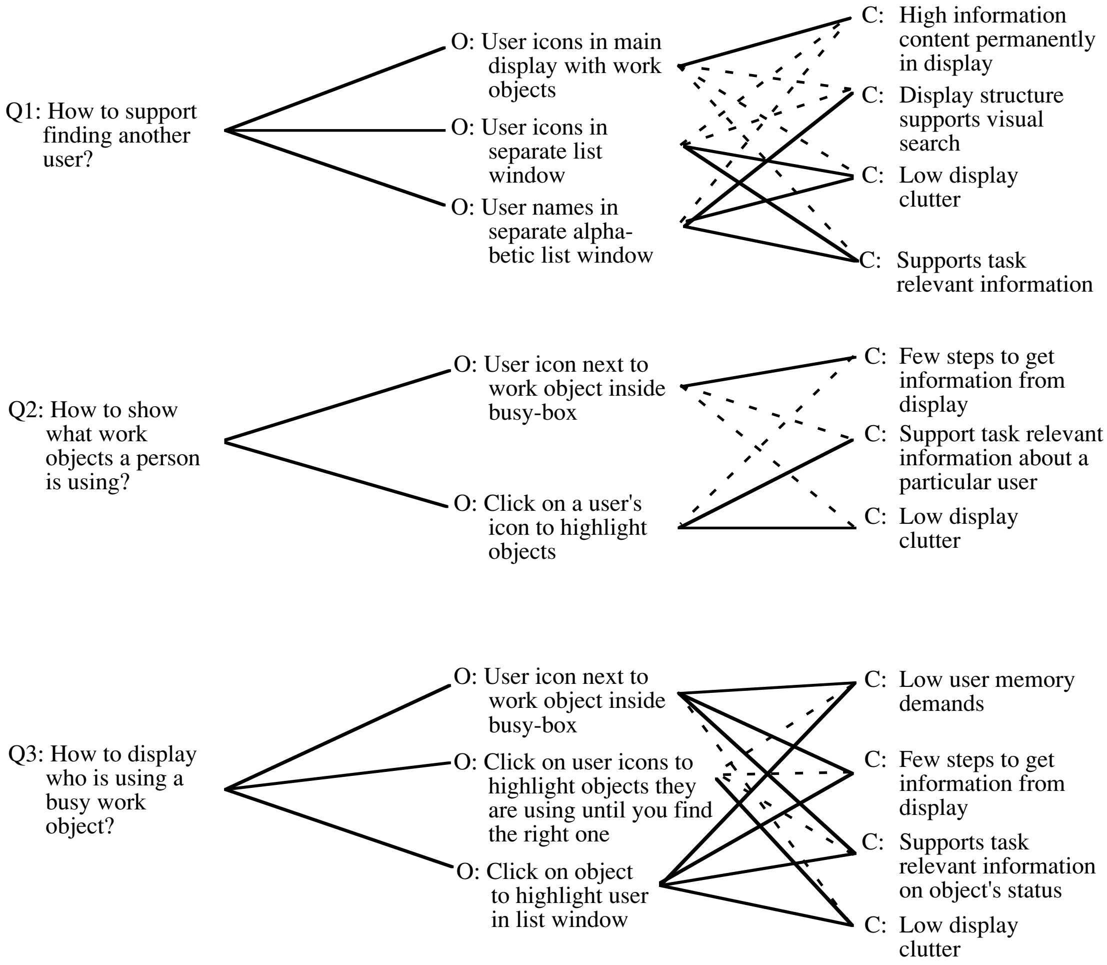
Q1: How to support finding another user?
O: User icons in main display with work objects
O: User icons in separate list window
O: User names in separate alphabetic list window
C: High information content permanently in display
C: Display structure supports visual search
C: Low display clutter
C: Supports task relevant information
Q2: How to show what work objects a person is using?
O: User icon next to work object inside busy-box
O: Click on a user's icon to highlight objects
C: Few steps to get information from display
C: Support task relevant information about a particular user
C: Low display clutter
Q3: How to display who is using a busy work object?
O: User icon next to work object inside busy-box
O: Click on user icons to highlight objects they are using until you find the right one
O: Click on object to highlight user in list window
C: Low user memory demands
C: Few steps to get information from display
C: Supports task relevant information on object's status
C: Low display clutter

Figure 6. Expanded QOC design space. The new Options are the bottom ones for Q1 and Q3

Note that the assessment against the *display structure supporting visual search* Criterion has now been altered to reflect the identification of this better solution. Given the Criterion of *low display clutter* it is not clear that the icons make the best use of screen real estate. This is increasingly important when the number of users increases and may not all fit within the window. Removing the icons and having a list of user names only would improve the Option. Doing away with the icons here does not hurt the design as they are only really useful in the main display to distinguish people from objects (we have not represented this in QOC, but it could easily be done if the claim was felt to be contentious). In the list, status information could be displayed by the colour (or shade) of labels as easily as with icons. The main advantage is that the names alone are far more compact, thus making it more likely that the user you are looking for is on screen. Having an alphabetical ordering scheme means that, if you know the name you will find them, no matter how many there are. This can be summarised by a proposed Option to put *user names in separate alphabetical list window*. The above argument claims that such an Option would be assessed better than the original design B Option. The result is shown in figure 6.

A major difficulty highlighted by the QOC analysis was that neither original design was better overall in the part of design space examined. They support different activities reasonably well, but all of the activities are important ones for the final system. For Q3: *how to display who is using a busy work object*? an alternative Option might be to *click on the object to highlight name in a list* *window*. This could also enable you to go on and find out what else that person is using by clicking on the name, which might be useful task relevant information. Such a solution is supported by all of the Criteria for this Question, and is therefore a good example of overcoming the negative assessments against the original Options while maintaining the positive ones.

Further examination might reveal flaws in our new design proposals, but this would draw us away from the scenario issues. Therefore we will not seek to refine the analysis further, although we would hope that the design suggested is an improvement on the original proposals. In this particular case, further refinement of the QOC in phase 4—tidying up the QOC representation is not necessary. This is in part because of the nature of the design problem we have been addressing. Other kinds of problem where it is more relevant will be mentioned shortly. We will now move to the final phase in which we make decisions about the most appropriate design solutions based on our new understanding of the problem and of the design space.

### *2.2.4 Phase 5: Make decisions*

Figure 7 highlights the design recommendations resulting from the analysis. It should be clear from the previous section how the argument has built up to support these choices. The Design Space Analysis helped highlight weaknesses in the original proposals, thereby allowing these to be overcome. In each case we have selected the Option which is supported by the largest number of Criteria. Things are not always this simple, since we may have chosen a solution which had little support from several Criteria but strong support from a very important one. We would then have designed ways to overcome its negative assessments. However, in this case we were able to find solutions by recombining the basic ideas already in the two designs A and B. Note that we do not typically make detailed distinctions about the weighting to be attached to Criteria or assessments in the notation. We typically rely on the supporting argument for that (i.e. the textual description in the present case) — see MacLean et al., (1991a) for further discussion of this point.

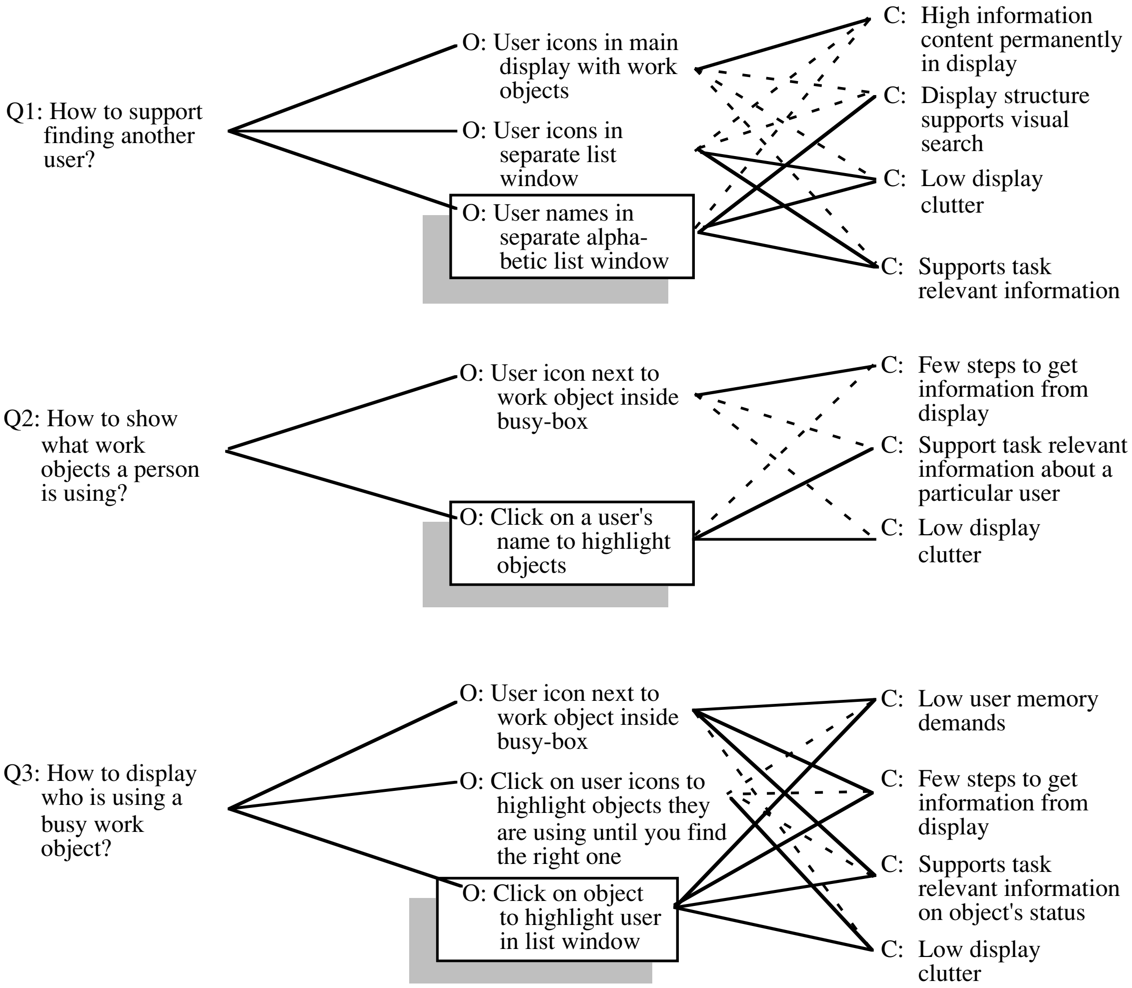
Q1: How to support finding another user?
O: User icons in main display with work objects
O: User icons in separate list window
O: User names in separate alpha-betic list window
C: High information content permanently in display
C: Display structure supports visual search
C: Low display clutter
C: Supports task relevant information
Q2: How to show what work objects a person is using?
O: User icon next to work object inside busy-box
O: Click on a user's name to highlight objects
C: Few steps to get information from display
C: Support task relevant information about a particular user
C: Low display clutter
Q3: How to display who is using a busy work object?
O: User icon next to work object inside busy-box
O: Click on user icons to highlight objects they are using until you find the right one
O: Click on object to highlight user in list window
C: Low user memory demands
C: Few steps to get information from display
C: Supports task relevant information on object's status
C: Low display clutter

Figure 7. Phase 5; QOC Design Space Analysis and Design Recommendations (shown as boxed Options).

To summarise the proposed solution shown in figure 7, we suggest the provision of multiple ways of accessing both people and objects to allow the user to choose the means to suit their task. The user should then be able to access further information via the most salient object in the current task. Our conclusion favours a separate, compact and alphabetical list of users' names (which could be colour coded as are the icons in both A and B for status information). You should be able to click on a user's name to display the objects they are using, and you should be able to cause the list to highlight the user when you click on a busy work object. This design supports both user-oriented and work object-oriented tasks equally and reduces display clutter considerably, making any visual search that much easier. This resolves the trade-off between these two types of tasks which would have had to be made if one or other of the original designs was accepted.

## 3. Discussion

The reader should not over-generalise from this example. There are many different kinds of design problem which emphasise different kinds of design activities and are therefore supported by Design Space Analysis in different ways from that illustrated here. The starting point for the current problem was two alternative designs. In such a case, starting a design space is relatively easy as competing Options can be identified fairly readily to give a good start to the QOC representation. This allowed us to present the analysis in quite a lot of detail. Although this is not an uncommon kind of problem, other design problems can be much more open-ended. For example, if we start with a single solution, it can be harder to find an appropriate Question to identify how best to look at the problem. Sometimes in early design there may be no existing solution to improve upon, so still more open-ended exploration would be necessary.

Such situations will have implications for the way in which the process model is used. For example, phase 2 may have much rougher QOC than shown here. When exploring the design space it may be much less clear what are candidates for Questions, Options or Criteria, and fairly extensive re-working may be necessary to produce a coherent QOC. Indeed, for some kinds of problem a rough QOC may be sufficient to get a feel for the issues, and the later phases may be ignored. Similarly, phase 4, 'tidying up' the QOC, may be much more problematic than the example described here. Sometimes the Questions may not be giving adequate coverage of the relevant issues, or may not be organising the design space in an appropriate way (see Bellotti et al., 1991). Fairly significant re-shaping of the design space may be necessary in such circumstances.

Some more discussion of rough QOC can be found in Shum (1991a, 1991b), and some examples of rough QOC and re-organising design spaces can be found in McKerlie and MacLean (1993a). Neither of these references put so much emphasis on the systematic development of the design space as we have done here. However, McKerlie and MacLean (1993a) go into considerably more detail about uses of Design Space Analysis over a period of time in a collaborative project. This highlights benefits we have not discussed at any length here, such as the role the QOC notation can play in assisting problem solving, in providing a basis for sharing and collaboration within a design team and in preserving a record of design deliberations over time. The present chapter has presented the development of a design space in a fairly clinical way in an attempt to communicate how the process models work. The reality within a design project is rather more subtle, both because of the range of different problems which may have to be tackled, as discussed above, and because of the dynamics of the design process itself.

We should emphasise that the intention is not to produce a 'complete' description of a design space using QOC. Rather its value comes from supporting exploration of fairly focused parts of the design space. Some of the situations where producing QOC might be encouraged include poorly understood issues, critical issues, areas where there is disagreement among the design team or where there is an expectation that the solution chosen will not be obvious in the future (e.g. because of a particularly clever solution, or because a short-cut had to be taken). There are many pragmatic reasons for choosing to produce QOC, but as McKerlie and MacLean (1993a) point out, the goal of design is doing design — not producing QOC. It is therefore important to place effort in areas where there is likely to be benefit, either for the immediate design requirements or for future comprehensibility of the design produced.

Another point worth emphasising here is that the QOC representation is not supposed to be a stand-alone representation. Rather, it summarises the structure of a design space in ways which linear text, diagrams of possible solutions or even prototypes cannot. The terse wording within the various elements is seldom comprehensible on its own. It generally requires further detail to be fully understood. For example, the text in this chapter has served to describe the various QOC figures presented. The value of the notation is to show alternative design possibilities, how they relate to each other and what determines their suitability, in a form which allows these relationships to be examined simultaneously. Compare this to a more traditional representation (e.g. the screen shots in figures 8 and 9). These show more clearly how the components of possible solutions fit together, but do not highlight the differences between these solutions, far less any reasons for these differences. A variety of complementary representations are essential for describing designs adequately (see also McKerlie and MacLean, 1993b).

The key to the technique we describe here for carrying out Design Space Analysis is to focus on identifying and organising relevant information. Design Space Analysis in and of itself has nothing to say about what domain information is relevant or should be represented. It simply provides a framework and a notation within which it can be represented. In the example described here all the initial information came from the description in the design scenario. In other situations it might come from a requirements document or the experience of the designers. Once some initial design information has been assembled, the approach uses that to 'bootstrap' the design space. The distinctions between Questions, Options and Criteria and the design process models presented in Section 1 help focus the designer on parts of the space which might fruitfully be augmented and give some guidance on the form that the augmentation might take — for example, in looking for better Criteria to evaluate the current state of the design, or in looking for new Options which meet the Criteria better.

The way in which Design Space Analysis assists with organising design information is valuable. However, it would be even more valuable to provide support for identifying what information *should* be present and for identifying the implications of information provided. That is the role of other contributions to Amodeus. The system modelling component is centred around work on formal specifications from the computer science tradition (e.g. chapters 2.2 and 2.3). The user modelling component comes from the psychological tradition and two approaches are represented in the project (e.g., chapter 2.1). Finally, CO-SITUE is a framework which provides a systematic basis for the categorisation of the factors that affect user performance (see chapter 2.5). In each case, these approaches emphasise a particular kind of information (to do with the system, the user, or the overall coverage which is required).

In a number of scenarios which have been analysed during the course of the Amodeus project, we have attempted to produce a QOC design space representation which combines this variety of perspectives within a single design space (see Bellotti, 1993 for an example). To oversimplify the conclusions a little, the major initial impact of the other Amodeus approaches on Design Space Analysis is seen in the Criteria they generate. For example, user modelling tends to emphasise Criteria to do with the cognitive processing of information or the knowledge required to use the system. In general, our attempts so far to combine these analyses have been post-hoc, taking the separate results of each individual analysis and relating them to each other using the QOC notation. It would clearly be more powerful to have the contribution of this variety of approaches embedded within the design process. If we consider how this might be done within the process model presented here, there are implications for both the sequential and declarative components. For example, in phase 1 of the sequential model, rather than rely only on the initial design information provided, we could suggest specific kinds of information which should be considered, and perhaps techniques for generating it. In the declarative model, specific kinds of information could perhaps be included. For example, the heuristic *design to a set of Criteria* could be expanded with guidance on how to develop an appropriate set of Criteria using the other Amodeus modelling approaches.

The approaches included in Amodeus promise to help select what available design information is important, and to help identify what other information is necessary but is not given. We believe that they can contribute to a solid information base on which to build a design space. The challenge now facing us is to take the next step beyond the process model presented in this chapter and make that a practical reality.

## Acknowledgements

The work reported here was partly funded by the European Commission under the Esprit Basic Research Actions 3066 (AMODEUS) and 7040 (AMODEUS II), and by SERC CASE Award 88504176. We would like to thank Phil Barnard, Jon May, Diane McKerlie and Tim O'Shea for helpful feedback on an earlier draft of this chapter.

## References

Bellotti, V., MacLean, A. & Moran, T. (1991) *Structuring the Design Space: The role of Questions in QOC Design Rationale.* AMODEUS working paper, RP6/WP4.1.

Bellotti, V. (1993) Integrating theoreticians' and practitioners' perspectives with Design Rationale. In Proceedings *of INTERCHI'93: Human Factors in Computing Systems, CHI'93 and INTERACT'93, April 24 —29, Amsterdam, Netherlands.* New York: ACM.

Guindon, R. (1990) Knowledge exploited by experts during software system design. *International Journal of Man-Machine Studies,* Vol. 33, 279-304.

McKerlie, D. & MacLean, A. (1993a) *Reasoning with rationale: Experience with Design Space Analysis*. AMODEUS II working paper ID/WP7.

McKerlie, D. & MacLean, A. (1993b) QOC in Action: Using Design Rationale to Support Design. In*Video Program, INTERCHI'93: Human Factors in Computing Systems, CHI'93 and INTERACT'93, April 24 —29, Amsterdam, Netherlands.* (Available in ACM SIGGRAPH Video Review Series).

MacLean, A., Young, R., & Moran, T. (1989). Design Rationale: The argument behind the artifact. In *Proceedings of CHI'89: Human Factors in Computing Systems, April 30—May 4, Austin, Texas,* 247-252. New York: ACM.

MacLean, A., Bellotti, V. & Young, R. (1990). What rationale is there in design? In Diaper, D., Gilmore, D., Cockton, G. and Shackel, B. (Eds). In *Proceedings of INTERACT'90: Third IFIP Conference on Human-Computer Interaction*. (Aug. 27-31, 1990, Cambridge, England). Amsterdam: Elsevier North-Holland, 207-212.

MacLean, A., Young, R., Bellotti, V. & Moran, T. (1991a). Questions, Options, and Criteria: Elements of Design Space Analysis. *Human-Computer Interaction*, Vol. 6 (3&4). [Special Issue on Design Rationale: John M. Carroll and Thomas P. Moran (Eds.)], 201-250.

MacLean, A., Bellotti, V., Young, R. & Moran, T. (1991b). Reaching through analogy: A design rationale perspective on roles of analogy. *Proceedings of CHI '91 Conference on Human Factors in Computer Systems,* New York: ACM, 167—172.

MacLean, A., Young, R., Bellotti, V. & Moran, T. (1991c). Design Space Analysis: Bridging from Theory to Practice via Design Rationale. In *Proceedings of Esprit'91, Brussels, 25-29 November, 1991*, Office for Official Publications of the European Communities, Luxembourg, 720-730.

Shum, S. (1991a). Cognitive dimensions of design rationale. In D. Diaper & N. V. Hammond (Eds.), *People and Computers VI.* Cambridge, England: Cambridge University Press, 331-344.

Shum, S. (1991b). A Cognitive Analysis of Design Rationale Representation. *Doctoral Dissertation,* Department of Psychology, University of York, UK.

Shum, S., and Hammond, N. V. (1993). Argumentation-Based Design Rationale: Conceptual Roots and Current Use. *Technical Report EPC-93-106,*: Rank Xerox EuroPARC.

Shum, S. (in press). Analysing the Usability of a Design Rationale Notation, and its Interaction with Modes of Designing. In T. P. Moran and J. M. Carroll (Eds.), *Design Rationale.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Wason, P.C. (1968) Reasoning about a Rule. *Quarterly Journal of Experimental Psychology*, Vol. 20, 273-281.

Young R. and Barnard, P. (1987) The Use of Scenarios in Human-Computer Interaction Research: Turbocharging the Tortoise of Cumulative Science. In *Proceedings of CHI+GI'87: Human Factors in Computing Systems*, Toronto, Canada. ACM, New York, 21-26.

# *Appendix A*

## The design scenario: Distributed multimedia services

A prototype CSCW (Computer supported co-operative work) environment is being developed by a manufacturer which wants to create a market for its integrated multimedia workstations. Among the applications being targeted is the production of multimedia travel brochures by the publicity and marketing departments of international holiday companies.

Task analyses have been conducted on the targeted users. The system should allow users access to a shared database of 'work objects' (elements of brochures, plans about brochures, and comments on both of these), and provide users with both real-time communication and a mail system. The task analyses have differentiated between the 'entities' in the worksystem (i.e. the people who can use it) and the 'objects' (the contents of the database that they work upon), and identified the need for users to have an overview of the worksystem so that they can get information about both entities and objects.

To provide this overview, the designers have decided to use a desktop-style presentation of the objects, with icons representing types of object in the worksystem. Only one person can access an object at a time, and so if an object is being used it is surrounded by an additional rounded border. People in the worksystem are to be similarly represented by stylised head and shoulder icons (that do not attempt to resemble the individual they represent), and which are coloured black if they are the user of the workstation themselves, white if they are other people currently using the system, and grey if they are people known to the worksystem but who are not currently using it. Users can individually arrange the objects on their desktop as they please, but the entities are positioned by the system (see system section below).

## The Problem

The task analysis identified the need for users to be able to find out what objects other people were currently working on.

A) *People in object overview (Figure 8).* The person icons could be represented in the object overview, placed next to the icon of the object that they are working on (inside the rounded ' busy' border). When a user stops working on one object and starts working on another, their entity icon would be moved (by the system) next to the object and enclosed with it in a box. (Specifying which object to work on can be achieved with a single mouse click). Entities without proprietary boxes remain adjacent to the object they were last using, even when the user is represented by a white icon, indicating that they are now using some other facility such as mail or the videophone not represented as objects.

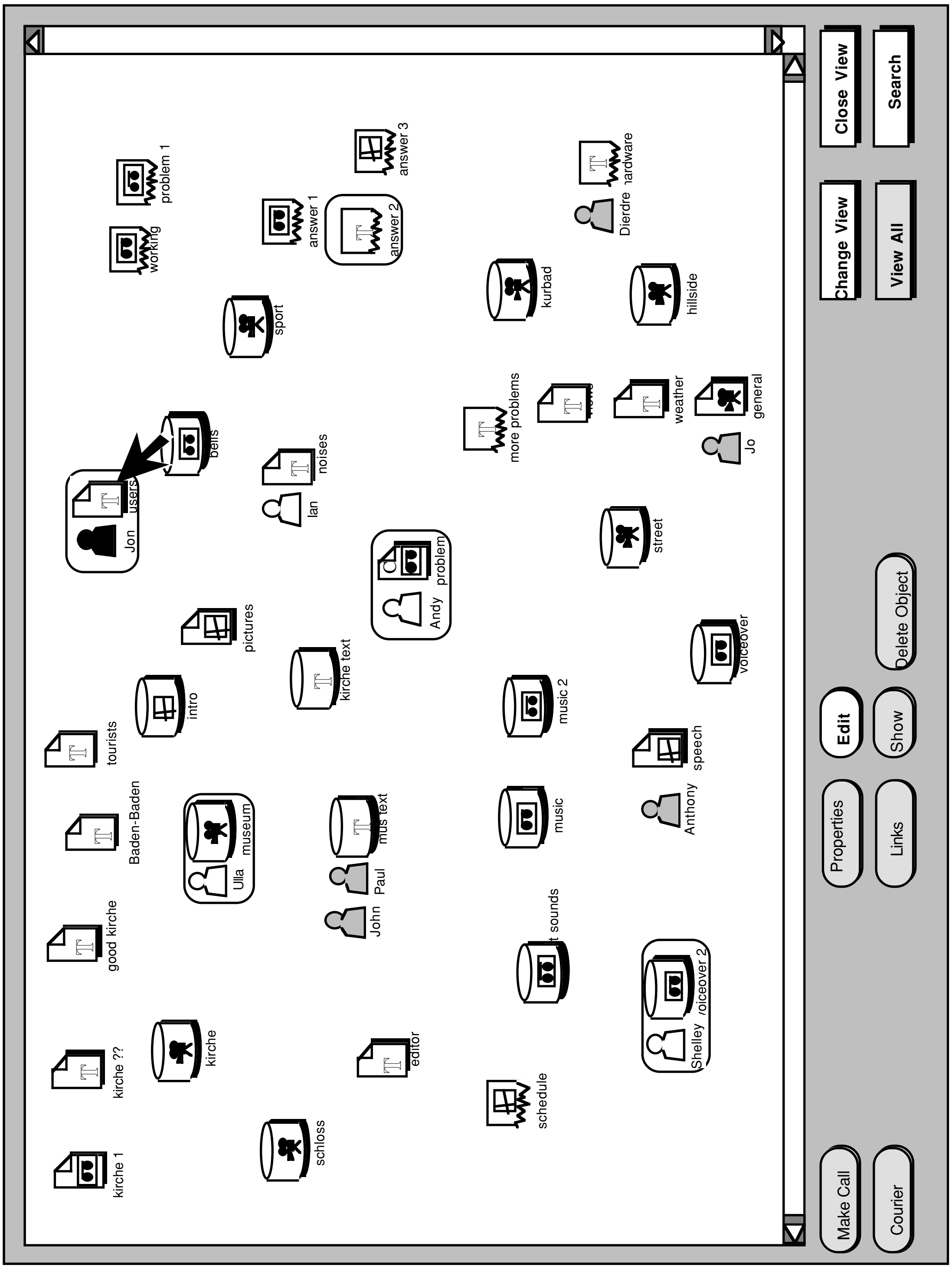

Figure 8 The suggested design with person icons included in the object window.

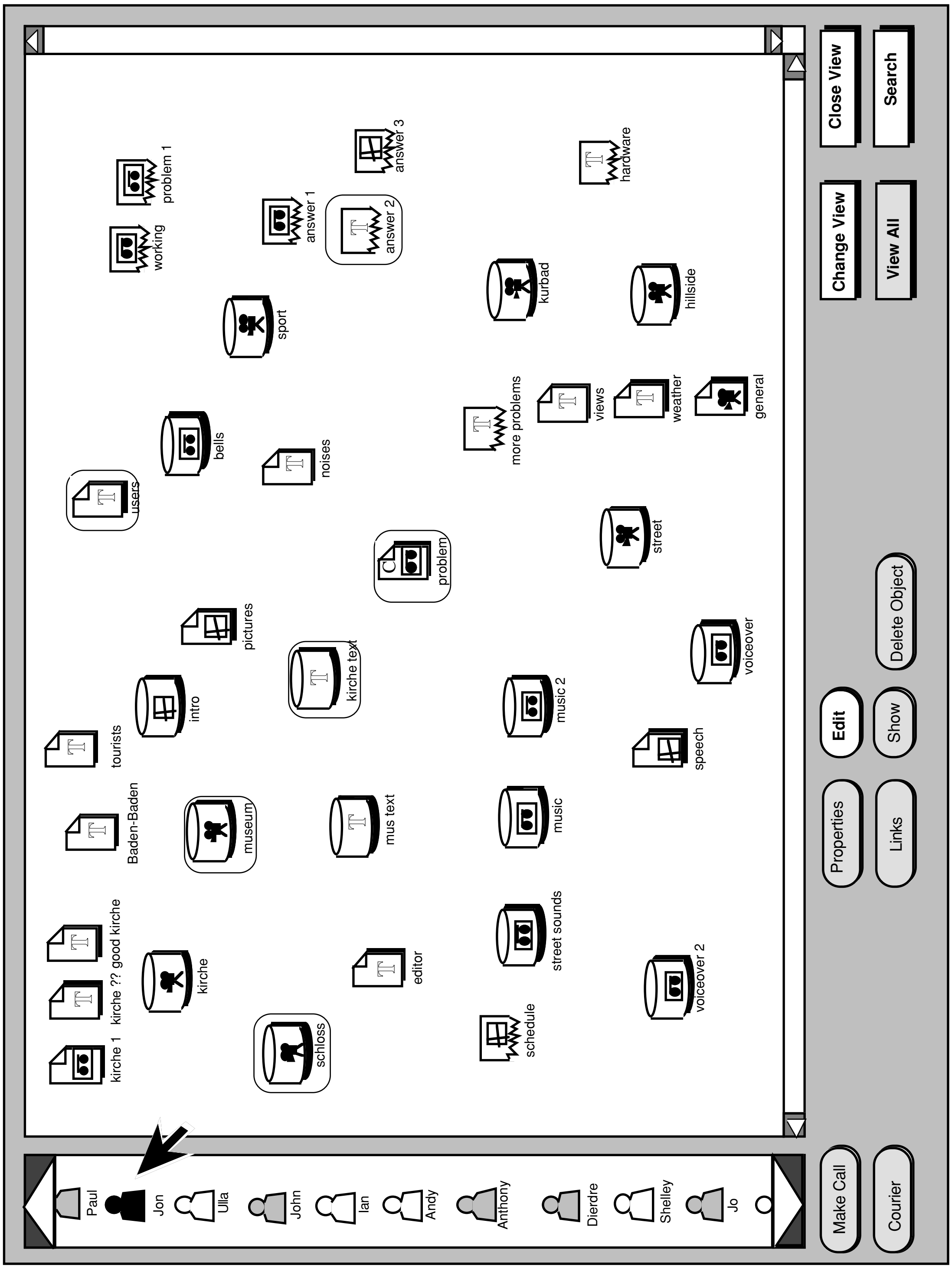

Figure 9 The suggested design with the person icons in a separate window.

B) *People in separate overview (Figure 9)*. The person icons could be represented in a separate overview, and users would have to select them and click on a 'Properties' button to find out what they were working on (along with other information including their role and their 'do not disturb' status which would be editable by the individual user). This would mean that the information was not immediately available, and only one person's information could be seen at a time.

What advice can you provide? Are the designers focusing on the correct issues? Could your approach suggest any alternatives?

*Subsequent problems:* The designers have been working on the assumption that the desktop overview of the objects would be large enough to display icons for all of the objects in the worksystem. For large worksystems, this might not be the case, and the overview would have to be scrollable, as some objects may not be visible within the current view. There could also be too many people in the worksystem to represent in a separate workgroup overview without making that also scrollable. How would this affect your advice?

## The Context

*System*: A prime consideration is that the system should use the company's existing workstation, which incorporates a large colour monitor with a top/centre mounted wide-angle lens camera (for video communication), mouse, keyboard, microphones, speakers, laser printer, scanner and document camera. In both cases, the user arranges the 'objects' on their desktop to suit themselves; 'entities' are located adjacent to 'objects' automatically by the system. The buttons located below the 'objects' windows appear white when available and grey when unavailable: these buttons are associated with a selected 'entity' or 'object'. For both versions this includes communications functions, but additionally status information for the second version. Simultaneous or multiple selection of 'entities' and 'objects' is not allowed.

*Target users*: Professional publicity and promotional staff, with high levels of task-specific skills in production and layout of paper-based brochures and videos using existing technology.

*General tasks*: Co-operative planning of documents; editing and assembling of existing materials using integrated editor functions. A typical task situation would involve a user requiring access to an 'object', for example to edit a section of videotape, but having to contact the current user to request access.

*Design situation*: Small group of designers with stated intention to provide designs that support the users' goal tasks. Product intended as flagship demonstrator for international shows and exhibitions in first instance rather than directly for the marketplace. No client other than the company that makes the workstations.

## Open Issues

*Information presentation*. The applications screen layout is as defined in the figures.. Unconfirmed aspects include the presentation within the workstation environment of the editing windows for the different 'objects': including what control over them, for example in resizing and moving, that the user can exhibit. Also unknown is the form, positioning and precise effects of the "Properties" button mentioned in design B. Design questions which have not yet been addressed include how the system will position entity icons once a user has finished using the application associated with an object, and are succeeded by another user. Also unaddressed (for design A) is the question of what would happen if a user arranges the objects on their desktop in such a manner that when the system updates the entity positionings, another object is occluded by an entity.

*Task*. Little explicit detail about *why* the users would want to access the 'object' applications, and therefore no parameters about frequency of use or on the ramifications of user-user interaction or protocols for access.

*Dynamics of the interaction*. Underspecified in terms of the users' action strategies and system responses. We know in general terms what the user has to do, but not the overall structuring of this within the context of an ongoing interaction.