HUMAN-COMPUTER INTERACTION, 1991, Volume 6, pp. 201-250

Questions, Options, and Criteria: Elements of Design Space Analysis

Allan MacLean
Rank Xerox EuroPARC

Richard M. Young
MRC Applied Psychology Unit

Victoria M. E. Bellotti
Rank Xerox EuroPARC

Thomas P. Moran
Xerox Palo Alto Research Center

ABSTRACT

Design Space Analysis is an approach to representing design rationale. It uses a semiformal notation, called QOC (Questions, Options, and Criteria), to represent the design space around an artifact. The main constituents of QOC are *Questions* identifying key design issues, *Options* providing possible answers to the Questions, and *Criteria* for assessing and comparing the Options. Design Space Analysis also takes account of justifications for the design (and possible alternative designs) that reflect considerations such as consistency, models and analogies, and relevant data and theory. A Design Space Analysis does not produce a record of the design process but is instead

Authors' present addresses: Allan MacLean and Victoria M. E. Bellotti, Rank Xerox EuroPARC, 61 Regent Street, Cambridge CB2 1AB, England; Richard M. Young, MRC Applied Psychology Unit, 15 Chaucer Road, Cambridge CB2 2EF, England; Thomas P. Moran, Xerox Palo Alto Research Center, 3333 Coyote Hill Road, Palo Alto, CA 94304.

CONTENTS

a coproduct of design and has to be constructed alongside the artifact itself. Our work is motivated by the notion that a Design Space Analysis will repay the investment in its creation by supporting both the original process of design and subsequent work on redesign and reuse by (a) providing an explicit representation to aid reasoning about the design and about the consequences of changes to it and (b) serving as a vehicle for communication, for example, among members of the design team or among the original designers and later maintainers of a system. Our work to date emphasises the nature of the QOC representation over processes for creating it, so these claims serve as goals rather than objectives we have achieved. This article describes the elements of Design Space Analysis and illustrates them by reference to analyses of existing designs and to studies of the concepts and arguments used by designers during design discussions.

1. MOTIVATION FOR DESIGN RATIONALE

The end product of systems design is a concrete artifact, that is, a software and/or hardware system or product. The output of what is normally considered the design process is a description of the artifact, such as a specification for implementing it or blueprint for constructing it. Such descriptions represent the designer's decisions of what the artifact is to be like, but they do not contain any of the designer's thinking and reasoning behind these decisions, that is, the arguments for why the artifact is the way it is. A *design rationale* is a representation for explicitly documenting the reasoning and argumentation that make sense of a specific artifact.

Design rationale is important because an artifact needs to be understood by a wide variety of people who have to deal with it. This variety of people ranges from those who design and build it (e.g., systems analysts, user-interface designers, and software implementers) to those who sell and service it (e.g., trainers and software maintainers) to those who actually use it. What is important for many of these people is not just the specific artifact itself but its other possibilities. For example, a designer decides between different possible ways to shape the artifact; a maintainer wants to change the artifact to respond to a new need without disturbing the integrity of the artifact; a user wonders why this artifact is different from some other familiar artifact. We hypothesize that an important way to understand an artifact is to compare it to how it might otherwise be.

In this article, we propose a style of analysis, which we call *Design Space Analysis,* that places an artifact in a space of possibilities and seeks to explain why the particular artifact was chosen from these possibilities. A Design Space Analysis creates an explicit representation of a structured space of design alternatives and the considerations for choosing among them—different choices in the design space resulting in different possible artifacts. Thus, a particular artifact is understood in terms of its relationship to plausible alternative artifacts.

A Design Space Analysis can be arbitrarily elaborate. In this article, we take a pragmatic approach in which we use the simplest possible analyses to solve particular problems. Therefore, we propose a very simple notation, called QOC, that focuses on representing the most basic concepts of Design

Space Analysis: *Questions,* which pose key issues for structuring the space of alternatives; *Options,* which are possible alternative answers to the Questions; and *Criteria,* which are the bases for evaluating and choosing among the Options.

We argue in this article that design rationale analyses of this kind are *coproducts of design,* along with the target artifact being designed. That is to say, documented analyses are themselves artifacts—They are explicit representations that must be designed and created by designers (by whom we mean any of a variety of players who influence the shape of the artifact in the design process, not just people who wear an official badge of "designer"). We argue that designers are capable of Design Space Analysis and that it is a fairly natural style of reasoning for them. It takes discipline and effort, however, to create such representations, and this effort must be targeted carefully at those aspects of the design process where the effort pays off. We further argue that this kind of reasoning cannot be "captured," for it is not simply a historical record of a design process or a structured representation of the dialogue among designers.

We argue that an explicit design rationale can be a useful tool in the design process in a variety of ways: from reasoning and reviewing to managing, documenting, and communicating. However, demonstrating the utility of a design methodology based on analytic rationale representations is beyond the present state of the art in design rationale research. We are at an early stage of this research—developing representations for design rationale theoretically (to understand their ability to encode the appropriate information), empirically (to understand whether designers can create them), and methodologically (to formulate effective procedures for creating them). Our objective in this article is to demonstrate progress on all of these fronts.

The article is organized as follows: First, in Section 2, we illustrate the basic elements of Design Space Analysis and the QOC notation by working through two examples and by reflecting on the important characteristics of our approach and how it relates to other work on design rationale. Then we proceed to several advanced topics of Design Space Analysis. In Section 3, we present empirical studies of designers at work and observations of how the structure of thinking in their deliberations relates to Design Space Analysis. In Section 4, we examine issues in representing various modes of justification and the coherence of design in Design Space Analysis. The majority of the article illustrates and discusses the properties of QOC-based representation of a design space. It is more concerned with the representation as a product of design than it is with the process of creating that product. We finish, however, by presenting an appendix with some preliminary advice—based on our current experiences, observations, and theory—on some heuristics for carrying out Design Space Analysis.

2. BASICS OF DESIGN SPACE ANALYSIS

In this section, we present the basic concepts of Design Space Analysis—Questions, Options, and Criteria—and describe the QOC notation. We illustrate these concepts and their interrelationships by analyzing two design examples. In the first example (Section 2.1), we examine a part of the user interface of a window system and try to understand the design of a scroll bar mechanism. The second example (in Section 2.2) analyzes two alternative designs for a bank automated teller machine (ATM) to help understand the essential differences between them. We then look (in Section 2.3) at some characteristics of our Design Space Analysis approach and how it compares to other approaches. Finally, we discuss (in Section 2.4) how these kinds of analyses fit into the context of the design process.

2.1. Analyzing an Artifact—A Scroll Bar Example

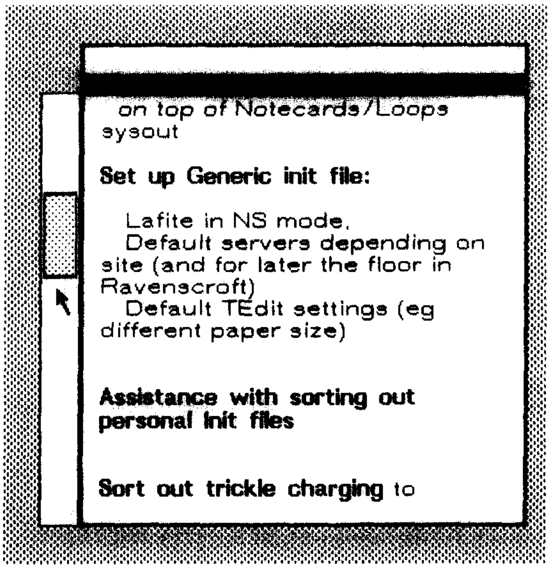
In this first example, we consider the scroll bar mechanism in the Xerox Common Lisp (XCL) environment, and we note that the scroll bar design is different from many of the more recent window environments. We want to understand why. We do this by "reverse engineering," a rationale for some attributes of the XCL scroll bar.

The windows in XCL provide views onto objects (such as documents) that are too large to be viewed in their entirety, and thus XCL provides scroll bars to control the views of the objects in the windows (see Figure 1). The first step in an analysis is to abstract from the artifact a set of characteristic *features,* which represent the design decisions that were made. We note several features of XCL's scroll bar design, for example:

- F1. The scroll bar is normally invisible and only appears when it is needed for scrolling.
- F2. The scroll bar is fairly wide.
- F3. The scroll bar indicates the position of the view in the window.
- F4. The scroll bar indicates the relative size of the view in the window.

There are, of course, many more features that characterize this particular design, and we could analyze them all. But we want to focus here on the first feature—that the scroll bar appears only when needed. How this works is that the scroll bar is normally invisible, but it appears when the cursor is moved from inside the window over the left edge (to just the location where the scroll bar appears). This is an interesting design decision, and it is different from

Figure 1. **A window in the Xerox Common Lisp (XCL) system. The scroll bar appears when the cursor is moved out of the left edge of the window.**



most of the more recent window systems that have their scroll bars permanently fixed to the windows (e.g., the Macintosh). Our problem is to understand why this might have been done in XCL.

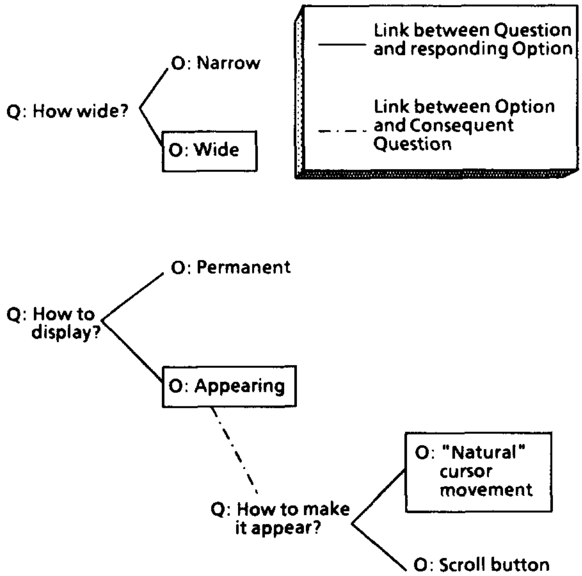
Questions and Options

We address the problem by analyzing the design space to see how a decision to have an appearing scroll bar could have been made, to understand whether it was a good decision, and to determine what the tradeoffs might have been.[1] Actually, the width of the scroll bar is a related feature, and so we analyze these two features together. The method for a Design Space Analysis is to view each feature as only one Option available among a set of other Options, to pose Questions for structuring the Options, and to enumerate the Criteria that determine the choice of particular Options. We use the QOC notation to represent this analysis.

Any specific Design Space Analysis must be placed in context. We begin with the assumption that there has been a decision to use a scroll bar as the user-interface technique to control viewing. From this decision, there follow

[1] The methodology used in this example is the "rational actor model" (Allison, 1971), in which we assume that the designer was acting rationally by making the most rational choice available to meet his or her criteria. Thus, we reason backwards from our knowledge of the designer's choice to what the criteria must have been.

Figure 2. **A fragment of the design space for the XCL scroll bar using the QOC notation, in which Questions and Options are enumerated and related. The boxed Options are the decisions made in the design of the XCL environment.**



several Questions in order to elaborate the decision into more detail. One obvious Question is how wide is the scroll bar? For now we only need to consider two qualitatively different Options: that the scroll bar could be relatively wide or relatively narrow. This is all we need to analyze the issues. We already know that there are Options to make the scroll bar permanently fixed to the window or to make it appearing, and we need to formulate a Question to relate these Options. Let us simply pose the Question as how should the scroll bar be displayed? (This is an interesting Question, because it would be easy just to assume that the scroll bar is permanent.) If we choose the Option of an appearing scroll bar, then as a consequence we have the further question of how to make the scroll bar appear? There are a number of possible Options for this. Some kind of a scroll button, either on the keyboard or on the screen, could be used to make the scroll bar appear. Alternatively, we could use some kind of a "natural" cursor movement, such as a movement of the cursor over the edge of the window to where the scroll bar will appear. In this way, we have begun to articulate the design space.

Figure 2 shows the QOC notation for the design space thus far. The notation uses a node-and-link diagram to portray the Questions, the Options, and their relationship. The links show the Options that respond to the Questions and the consequent Questions that follow from the Options. The

Options chosen (i.e., design decisions made) for the XCL environment are shown boxed. Figure 2 begins to provide a context for understanding these decisions. The alternative Options represented give an understanding of how the design could have been different (e.g., a **permanent, narrow** scroll bar). The Questions organize the Options by highlighting the critical dimensions along which the Options differ. The role of the Questions is to delineate local contexts within the design space to help ensure that like Options are compared with each other. The set of Questions provides a structured space of Options. Good, incisive Questions help to open up novel Options. That is to say, the role of the Questions is generative and structural, not evaluative. We must next consider how to evaluate the Options and to rationalize the decisions.

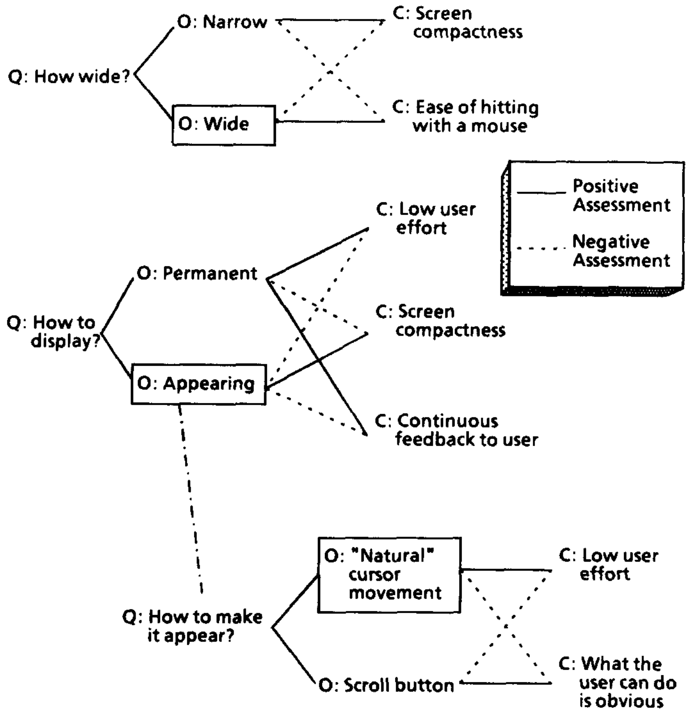
Criteria and Assessments

Choosing among the various Options requires a range of considerations to be brought to bear and reasoning over those considerations. The most important elements for organizing this reasoning are *Criteria,* and these must be added to the design space. Criteria represent the desirable properties of the artifact and requirements that it must satisfy. It is important to articulate clearly the Criteria, because they make clear the objectives of the design. They form the basis against which to evaluate the Options.

The QOC notation provides a way to represent an *Assessment* of whether an Option does or does not satisfy a Criterion. For example, one may wish to claim that a particular Option is good because it does not take up much "screen real estate" but that, on the other hand, it is not as easy to hit with the mouse. In this case, there are two Criteria: **screen compactness** and **ease of hitting with the mouse,** and there is a positive Assessment of that Option against the first Criterion and a negative Assessment against the second Criterion. The array of individual Assessments provides a context for making an overall judgment of the suitability of the Options. We give a more extensive discussion of the nature of Criteria in Section 4, but note for now that the Criteria are worded so that positive Assessments (i.e., satisfying the Criteria) are always what is desired. For example, we would not state the just-mentioned Criterion as **difficulty of hitting with the mouse.**

Figure 3 shows the design space of Figure 2 expanded to include some relevant Criteria. In addition to the Criteria just mentioned, there are the Criteria of **low user effort** (i.e., that the required user action is quick and easy), **continuous feedback to user** (i.e., that the user has a continuous visual indication of the state of the window view), and **what user can do is obvious** (i.e., that it is made clear to the user what actions are allowed). Figure 3 shows the Questions and Options to which the Criteria apply. Each Criterion is applied to all the Options for a given Question, but different sets of Criteria are applicable to different Questions. Note, however, that many Criteria are applicable across multiple Questions. This is important, as is

Figure 3. **A QOC representation of the design space for the XCL, elaborated from Figure 2 to include Criteria and Assessments. The boxed Options are the decisions made in the design of the XCL environment.**



discussed in Section 4. In the QOC diagrams presented in this article, a Criterion used for multiple Questions is represented multiple times, under each relevant Question, to simplify the spatial layout of nodes and links.

The Assessments are shown as labeled (positive or negative) links between the Options and the Criteria. For example, the Option of a permanent scroll bar is linked positively to (i.e., assessed positively against) the Criterion of continuous feedback to user (because the visual feedback from such a scroll bar is always available to the user), but it is linked negatively to screen compactness (because a permanently visible scroll bar always takes up screen space).

Now that we have laid out all the elements of the rationale, let us return to the problem we posed at the beginning of this example: Why was an appearing scroll bar chosen for the XCL environment? It may seem like a dubious decision, because the appearing Option satisfies only one of the three Criteria, whereas the permanent Option satisfies two of the three. This view is too crude, but it gives us an opportunity to discuss some of the properties of Design Space Analysis. The QOC notation is not intended to yield obvious choices by simply counting Assessments; rather, it provides the basis for a more complex discussion of the tradeoffs. One possible reason for choosing the **appearing** scroll bar is that **screen compactness** was considered to be the most important Criterion. If this were the case, however, it does not seem to be consistent with the choice of a **wide** scroll bar.

Another issue is the strength of the Assessments. If the positive Assessments favoring the choice of the appearing scroll bar were stronger (in some sense) than those supporting the permanent scroll bar, then this could explain the choice. In this example, we only distinguish positive from negative Assessments. We could very well use 3, 5, or 7 levels of assessment values (e.g., strongly positive, positive, mildly positive, neutral, mildly negative, negative, strongly negative). We are not against this in principle, but we feel that the complexity in making such Assessments usually outweighs the gains in understanding the overall picture, which is the real objective of the analysis. In this case, it is hard to differentiate the Assessments beyond what we have done. One reason is that other design decisions may influence the Assessment. For example, the negative Assessment of the **appearing** scroll bar against **low user effort** is difficult to evaluate more precisely without knowing more about the **appearing** Option. What we need to do in this case is to explore the **appearing** Option in more detail by Assessing the Options for **how to make it appear?** We note that the **"natural" cursor movement** Option was chosen, which is positive against **low user effort**. This particular version of the **appearing** Option in fact mitigates the negative Assessment of it against **low user effort**; that is, it makes the Assessment less negative, if not neutral.

This points to another property of Assessments—that the Assessments of different Options against a given Criterion are usually *relative*. For example, a **wide** scroll bar is relatively easier to hit with the mouse than a **narrow** one, and we represent this by positive and negative Assessments, respectively, against the **ease of hitting with the mouse** Criterion. Relatively speaking, wide is better than narrow, and this is all we need to say at this level of specification of the Options.

Returning to the **appearing** scroll bar issue, we note that the Assessments against the **screen compactness** and **continuous feedback to user** Criteria are fairly clear between the two Options. But the relative Assessment against **low user effort** is minimized. One might even argue that there is no difference—that the effort of scrolling by bringing up the **appearing scroll bar** with the **"natural" cursor movement** is no different from the effort of moving the cursor to a **permanent scroll bar**. The result is that **low user effort** should not be considered a very crucial Criterion for this Question.

We can see from this discussion that there is an *interaction* between the Assessments across different Questions. We have already seen that the Assessment of the **appearing scroll bar** depended strongly on the Option chosen for **how to make it appear?** There is also an interaction between the importance of Criteria across Questions. For example, the importance of the **screen compactness** Criterion depends on the choice of **how to display?** This Criterion is much less important if the **appearing** scroll bar is chosen because it does not clutter the screen all the time, as a **permanent** scroll bar would. This observation explains the apparent contradiction noted earlier as to whether the **screen compactness** Criterion was important or not to the XCL designer. The conclusion from this analysis is that **screen compactness** did seem to be the determining Criterion for choosing the **appearing** scroll bar.

This discussion illustrates the argument-based nature of Design Space Analysis. (Indeed, that has been a major emphasis of earlier presentations of our approach; see MacLean, Young, & Moran, 1989). The previous analysis, kept simple for expository purposes, can be elaborated or challenged (as perhaps most readers at this point would like to do). The objective of the QOC notation is to lay open for argument the elements of the rationale (see Section 2.3 for how the argument can be represented). The QOC notation allows us to pinpoint where the debatable issues are. The components of the QOC should therefore be treated not as fixed, structural relations but, rather, should be regarded as ongoing concerns, or even provocations. These issues are discussed further in Section 4.

2.2. Comparing Alternative Designs—An ATM Example

Let us now consider another analysis in which our problem is to compare two alternative designs of a bank ATM. ATMs are relatively simple devices with which most people are familiar, yet there is considerable variety in the design of ATMs, both in what facilities they provide and in the ways people interact with them. One interesting contrast is between a standard ATM (SATM) and a new fast-cash ATM (FATM) recently introduced by a British bank. The SATM offers a range of services, such as balance enquiries, new checkbooks or statements, and cash withdrawals. The FATM provides only for cash withdrawal. However, more than just restricting services, the procedure for using the FATM is different from the procedure for the SATM. Figure 4 shows the steps required to get cash from the two ATMs.

Our task is to compare these two designs to understand what the advantages of the new FATM design are. We do this with a Design Space Analysis. Our challenge is to produce a design space that captures both designs.[2] We do this by identifying parallel features that characterize the differences between the

[2] Note that scroll bar analysis also embodies two contrasting designs (although that was not its primary motivation)—the wide, appearing scroll bar of the XCL system and a narrow, permanent scroll bar found in many other systems. In fact, it was interesting to note, in a more extensive analysis of the design space around the XCL scroll bar than presented here, that the resulting design space included key Options for no less than four different existing scroll bar designs.

Figure 4. **The steps required to get cash from the SATM and the FATM.**

The SATM
1. Push card into slot
2. Type PIN number when prompted
3. Select "Cash Withdrawal" (from the several services offered)
4. Select "Another Amount" (you could have selected one of five preset amounts)
5. Type in amount required and press the Enter key
6. Select "No" (when asked if you would like to request another service)
7. Remove card from slot
8. Take cash from drawer and receipt from slot

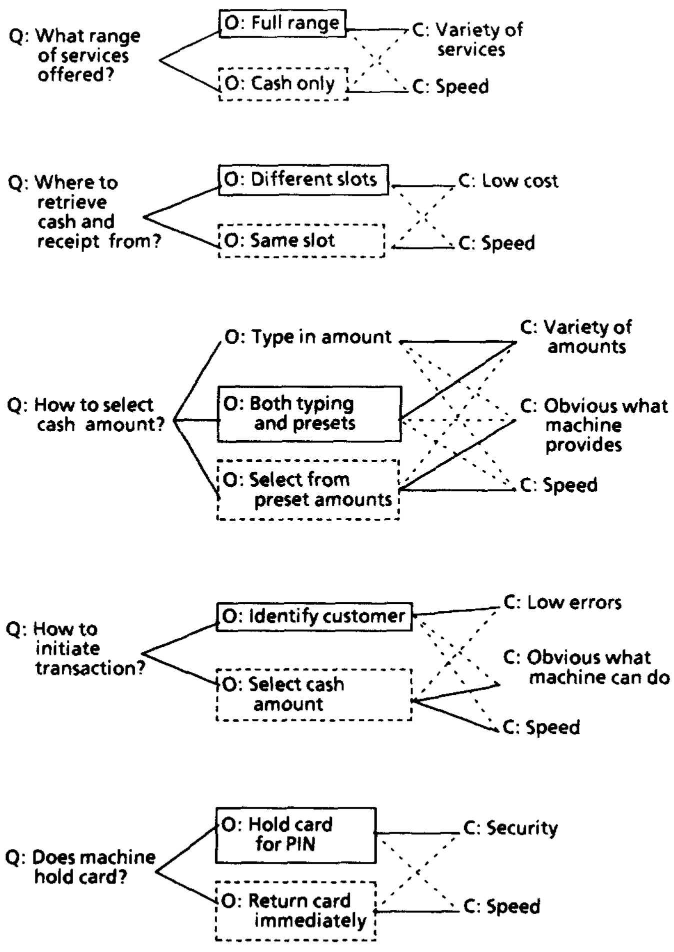
The FATM
1. Select cash amount (must be one of six preset amounts)
2. Insert card
3. Remove card
4. Type in PIN number
5. Take cash and receipt from drawer

two designs. These features are represented as alternative Options. We then formulate Questions that characterize the dimensions of these differences and Criteria to evaluate them. This analysis is summarized in a QOC diagram in Figure 5.

The most obvious dimension of difference has to do with the Question what range of services offered?, where the Options are full range for the SATM and cash only for the FATM. It is interesting to view the FATM design as challenging the standard Options of the SATM. To the Question where to retrieve cash and receipt from?, the FATM says why not take them from the same slot. Note that many features of the FATM design challenge not only the physical design of the SATM but also the procedure for using it. For example, the FATM starts off by having the customer select cash amount (rather than the SATM's having to first identify customer), which raises the question of how to initiate the transaction? Also, the FATM challenges whether the machine needs to hold onto the customer's bank card while it gets the customer's PIN and validates it.

Next we need to identify Criteria to help us understand the pros and cons of each design. Clearly, speed is the major Criterion motivating the FATM design and is relevant for all the Questions. However, speed is traded off against a variety of different Criteria in different cases, such as low cost, low errors, variety of services. For example, utilizing the same slot to get the receipt is costlier (because it involves hardware changes), whereas initiating the transaction with select cash amount could cause errors (because someone could select an amount and leave the machine in a nonneutral state for the next customer to come along). It appears that there may be some benefits in addition to speed—for example, making it more obvious what

***Figure 5.* A QOC representation summarizing the distinctions between the SATM and the FATM. The Options representing the SATM decisions are indicated by solid boxes, and the FATM decisions are indicated with the dashed boxes.**



the machine can do. The FATM has several new and independent features, but some of the Assessments of their being better for **speed** are arguable. For example, the argument that initiating by **select cash amount** is faster is based on the assumption that preparing the cash takes time for the machine to do and this time can be overlapped with the customer's identification. The argument that the **select from preset amounts** Option offers a speed advantage over the SATM's **both typing and presets** relies on disallowing slower typing selections and facing the customer with less choice. This may of course mean that some customers are unable to get their preferred amount of cash and could spend more time trying to work out how to get the amount they want.

This representation of the design space raises the more basic issue of what problem the FATM is attempting to solve: Is it the time per transaction that has to be reduced or is it important to restrict the kinds of transactions allowed? The analysis suggests both are relevant but does not really tease them apart. It is also clear that there are other possible Options for many of the Questions; therefore, other solutions may be better. Rather than suggesting definitive conclusions, this first cut at the design space sets the frame for further exploration. In fact, we carry out such an exploration in our empirical studies presented in Section 3, where we observe designers working on the same problem and expand the design space on the basis of their deliberations.

2.3. Characteristics of QOC Representation

Now that we have seen some concrete examples of QOC notation, we can step back and consider the important characteristics of our approach to design rationale. In pointing out these characteristics, we can compare and contrast our approach with other approaches to design rationale. These characteristics highlight many different aspects of our approach—representational properties of QOC, structural features of QOC that suggest the kinds of tools in which to embed the notation, and properties implicating the design process. We consider representational and structural characteristics here and consider the design process in Section 2.4.

Design Space Focus. The QOC representation emphasizes the systematic development of a space of design Options structured by Questions. The rationale in Design Space Analysis is built on the comparison of alternative Options. We can contrast this with the "claims analysis" approach of Carroll and Rosson (1991 [this issue]). Their approach is to evolve a design by exploiting the positive aspects of claims of its effectiveness while addressing the negative aspects. It is difficult at this point to compare the elements of their analysis with ours, but one obvious difference is that they continuously refine a single design, whereas we explicitly advocate the development of a space of alternatives.

Focus on Criteria. The QOC representation brings the objectives for the design, in the form of the Criteria, into explicit focus. Design rationale schemes derived from issue-based information systems (IBIS; Kunz & Rittel, 1970), such as gIBIS (Conklin & Begeman, 1989), do not explicitly bring forth criteria. IBIS has Arguments for and against Positions, but IBIS Arguments only implicitly refer to what we would call Criteria. Criteria per se are not proper objects of the IBIS notation. There are other proposals for characterizing the objectives of design. Lewis, Rieman, and Bell (1991 [this issue]) proposed that concrete problems are an effective way to represent design objectives. Others (e.g., Carroll & Rosson, 1990, 1991 [this issue]) have proposed the representation of design objectives by scenarios.

Coproduct of Design. A Design Space Analysis is not a record of the design process, but rather it is a coproduct. As an artifact in its own right, the QOC itself has to be designed. The rationale representation (a QOC) is created along with the descriptive representation (e.g., a specification) or the artifact itself (e.g., a prototype). Designers are clearly capable of producing such analyses: for example, Botterill's (1982) rationale for the IBM System/38 and Johnson and Beach's (1988) rationale for the style sheets in the ViewPoint office system. These rationales emphasize a logical rather than a chronological account. The argumentation that makes such accounts coherent has itself to be carefully crafted; it does not simply emerge from a historical record of the design process.

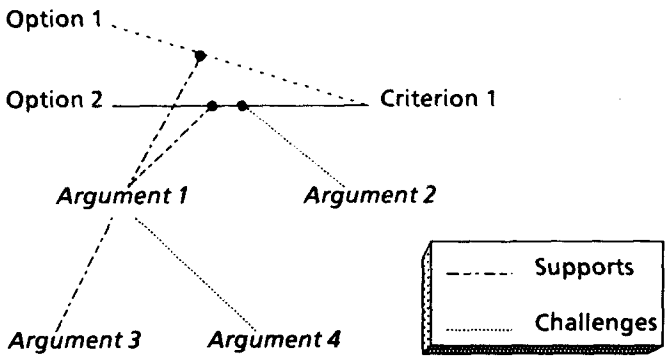
This approach can be contrasted with the IBIS-derived systems, such as gIBIS (Conklin & Yakemovic, 1991 [this issue]) and Procedural Hierarchy of Issues (PHI; McCall, 1986), whose purpose is to capture the history of design deliberations. Although the Issues, Positions, and Arguments of IBIS appear to be like our Questions, Options, and Criteria, they are quite different. Their Issues are general purpose in that they can be about any topic that comes up in design discussions, and their Positions similarly are general purpose. QOC Options, on the other hand, are specifically *design* options, and Questions are specifically to structure the design space. Also, as noted before, their Arguments are quite different from our Criteria. Because they are capturing discussion on the fly, the resulting argumentation structure is less coherent and succinct. Clearly, there is a tradeoff here between the effort to construct and the resulting coherence of the rationale. That being said, having an IBIS-encoded history would be a valuable resource in building a Design Space Analysis. In fact, we could see a QOC representation as a condensation of an IBIS history that brings out the most important elements of the history for logical argumentation. (This is similar in concept to Parnas & Clements's, 1986, notion of faking the design history.) Such a QOC analysis could suggest logical deficiencies in the discussions and could be used as a basis for directing the discussion into dealing with them. Thus, we see IBIS and QOC as having complementary roles.

Embedded in Design Activity. A corollary of being a coproduct of design is the symbiosis between the descriptive and rationale representations. The two kinds of representations are linked in that the chosen QOC Options represent selected features (i.e., those selected to be analyzed) of the artifact in the descriptive representation. Design is viewed as going back and forth between the two representations, intermingling the processes of construction and reflection in what Schön (1983, 1987) called "reflection in action." We envisage QOC representations being used in this kind of design context. In the same spirit, Fischer, Lemke, McCall, and Morch (1991 [this issue]) provided demonstrations of how rationale-based tools can be integrated into "design environments," which are construction kit facilities, along with active "critics" that give advice when conditions in the proposed design raise particular design issues. This advice from the critics is then backed up with a domain-specific knowledge base of rationale issues that are brought forth to explain the advice to the designer. The designer can also augment the rationale to include specifics of the current design situation. In our terms, their rationale representation is coupled with the descriptive representation of the artifact being designed.

Semiformal. The QOC notation is best regarded as semiformal. The basic QOC concepts (Questions, Options, Criteria) and their relations provide a formal structure for representing the design space, giving the representation a strongly diagrammatic style; however, the actual statements within any of the nodes of the diagrams are informal and unrestricted. QOC diagrams can quickly become messy and difficult to manage, even at 50 nodes, and computer-based tools are necessary. Hypertext systems (see Conklin, 1987) promise support for this kind of notation. The advantage of general purpose hypertext tools is that they allow informal annotations, as well as explicit structural relationships, to be easily represented. A disadvantage is that structures are tedious to create, and little support is provided for fully utilizing the structure. Nevertheless, useful results have been obtained with hypertext both in helping with the generation and organization of ideas (e.g., VanLehn, 1985) and in producing coherent structures for later examination (e.g., Marshall & Irish, 1989). Some of the examples in this article were created in the NoteCards system (Halasz, Moran, & Trigg, 1987), and the figures are based on NoteCards network browsers.

Argument Based. As we have seen in the earlier sections, rationale is based on argumentation, not proof. In principle, any elements of a QOC representation can be queried or challenged. Rather than being fixed, they are intended to be justifiable, based on further arguments that can be opened up to inspection, thereby making it possible for flaws in the original thinking to be identified and thus the representation improved. Behind all of the QOC elements (Questions, Options, Criteria, Assessments), there may be argu-

Figure 6. **The relationship between arguments and QOC Assessments. This diagram says: Argument-1 gives the reason for the relative Assessment that Option-1 is better than Option-2. Argument-2 challenges the Assessment of Option-2 (although it does not dispute the relative Assessment of Argument-1). Argument-3 supports Argument-1, whereas Argument-4 objects to it.**



ments supporting or objecting to their presence or their characterization. The arguments behind the Assessments are perhaps the most important. Figure 6 illustrates how arguments relate to Assessments. In Section 4, we elaborate several different kinds of arguments for justifying Assessments. In this article, we seldom show arguments explicitly in our diagrams (but see Figures 8 and 11). However, arguments are often discussed informally in the text, as may be seen in Sections 2.1 and 2.2.

The closest representation to our QOC notation is Lee and Lai's (1991 [this issue]) Decision Representation Language (DRL). DRL has a carefully defined semantic representation of decision elements and their relations. QOC's elements map very closely to DRL's (Questions are Decision Problems, Options are Alternatives, Criteria are Goals, Assessments are plausibilities of "Achieve" relations, Arguments are Claims). We expect any further elaborations of QOC to follow DRL's representation pretty closely (however, we are wary of the usability implications of proliferating distinctions in the representation). DRL is implemented in a system called SIBYL and is being used to explore various kinds of computational services over DRL structures, such as dependency management (Lee, 1990), where the consequences of, say, a revised assessment can be traced back to the decisions that were based on it. This seems to be a productive way to enhance the value of rationale representations by providing tools to aid design.

Expandable Detail. The argument basis is an example of an area where the QOC representation can be expanded to an arbitrary level of elaboration.

Although such details can be important, they can also get in the way of seeing the overall picture. The expandable-detail characteristic refers to the ability to work with manageable and comprehensible QOC diagrams that include only those parts of the rationale relevant to the particular issues at hand. Any part of a QOC diagram should be able to be regarded as a summary or shorthand for a more extensive story that may be probed for more details. This implies that adequate tools for this kind of representation must have good browsing, filtering, and display capabilities.

Purposeful. This is the other side of the coin from expandable detail. It is not practical to represent every possible detail, and it is not useful to do so. A QOC representation need not be a complete specification of a design. It may not be worth it to include well-understood or noncontentious parts of the design space. It may be sufficient to provide a Design Space Analysis for only parts of the space—where difficult issues were encountered during design, where nonobvious solutions were adopted, where it is felt that maintainers may need a clear understanding, where it is known that critical parameters of the design need careful monitoring, and so forth. In many situations, it may be appropriate to produce a rationale after the event, such as a maintainer creating a QOC to "reverse engineer" a part of a system and preserving it for future maintenance. Understanding the various purposes of Design Space Analysis requires us to consider the roles that it can play in the design process.

2.4. Design Space Analysis in the Design Process

Two issues come to mind in considering how an explicit Design Space Analysis can be useful and practical in the design process. One is the cost of creating analyses. The other is scaling analyses up to large design projects. It is too early in this research to have answers to these issues, but our current views on how such analyses may be used in design take them into account. One approach, being investigated by Fischer et al. (1991 [this issue]), is to create a large knowledge base of design rationale, which requires a considerable battery of system tools and a well-defined and well-maintained corpus of preencoded, domain-specific design rationale. We are wary of trying to build a single grand rationale structure for a design. Rather, we see a series of smaller Design Space Analyses being created during the course of a design project. Each analysis would be a focused effort serving a local purpose (and thus having a local payoff) in the design project while at the same time contributing to an overall documentation of the project's design rationale.

Explicit Design Space Analyses can be useful in design in a variety of different ways. First we consider the analyses as communication vehicles in the design process. Then we look at other classes of design activities.

Communication. Design is a social process that usually involves a variety of individuals, each with different skills and objectives. Communicating a shared understanding of the design is a crucial aspect of managing the design process. Explicit documentation of the rationale involved in the design should be a useful aid for communication between members of the design team, between designers and their users, and between the current design team and future design teams that want to build on or reuse parts of the current design. QOC representations should be effective communication vehicles, because they are simple enough to be understood by a variety of people, they are flexible enough to represent a variety of issues from a variety of viewpoints, and they are explicit enough to expose assumptions that can be challenged by others. However, the social subtleties in communication, such as what information people are willing to make explicit, should not be underestimated.

Perhaps the best way to see the importance of communicating an understanding is to see it as spanning the life cycle of a software design project. An understanding has to be shared among a wide variety of players and functions: marketing, requirements analysis, system design, user-interface design, implementation, documentation, sales, training, customer support, user customization, software maintenance, system administration, new releases, contracted enhancements, and system redesign. Design Space Analysis should help different teams within a project clarify their concerns to each other. Even within a design team, members should be better able to understand decisions with which they were not personally involved or be reminded of decisions in which they were involved earlier. Maintenance, including the adaptation to new user requirements after release as well as bug fixes, is a particularly important phase of design. Some estimate that this can occupy as much as 90% of the effort in the software life cycle (e.g., Balzer, Cheatham, & Green, 1983; Martin, 1977). Conklin (1989) claimed that as much as half of the effort in maintenance is understanding the system in order to make effective fixes and enhancements. Documented Design Space Analyses can help maintainers foresee the consequences of proposed alterations by making clear what decisions, tradeoffs, and evaluations will be affected by the change. Communication between designers and end users is also important. As we move toward more customizable systems (e.g., Mackay, 1991; MacLean, Carter, Lövstrand, & Moran, 1990), it is important to make clear not only what features can be customized but what the consequences of any change (customization) might be.

Again, Design Space Analyses require effort to create them. A major problem, as Grudin (1988) pointed out, in using them solely to enhance communication is that the people who create them are not the ones who benefit directly from them (a problem that also applies to software documentation). Therefore, we need to consider other aspects of the design process and utilize more immediate motivations for getting the analyses created. We consider the roles of Design Space Analysis in the creation, evaluation, reflection, and management of design.

Creation. We expect Design Space Analysis to facilitate innovation and reasoning in the design process by helping designers generate, represent, and think through, in a disciplined yet flexible way, their decisions—alternatives to them, the arguments for and against them, their implications, and the interrelations among them. The function of Design Space Analysis, however, is not just to facilitate making decisions. The process of developing QOC analyses exposes assumptions, raises new Questions, challenges Criteria, and points to ways in which new Options can capitalize on the strengths and overcome the weaknesses of current Options.

Reflection. Although it is difficult to be very analytical in the "heat" of the creative phases of designing, design projects are punctuated by reviews, reports, and presentations; these are natural times for standing back and reflecting on the state of the design. Design Space Analysis is an appropriate framework to help structure such reflection—for example, justifying design decisions and considering other opportunities for exploration. The QOC representation provides a uniform format with which to produce these reviews, reports, and presentations, thereby making them easier to store, cross-reference, and index, thus documenting the rationale behind the design project.

Management. Design Space Analysis should be useful in different aspects of project management. Breakdowns in design often occur because of designers' cognitive limitations (Guindon, Krasner, & Curtis, 1987). Effectively managing the complexity of design could affect both the quality and the efficiency of the design process. For example, Design Space Analysis provides a representation in which to incorporate design requirements (e.g., as Criteria) and constraints (e.g., as selected Options that impinge on the resolution of other Questions), both initially and as they change, and it can provide a map of the explorations over the design space. Embedded in appropriate tools, it could help track how well the explorations are satisfying the requirements. Because Design Space Analysis explicitly represents a design space, it is well suited to tracking changes.

It is an empirical question whether any of these proposed uses for Design Space Analysis will work in real design projects. Our first modest empirical step, presented in the next section, is to understand how the concepts and structure of Design Space Analysis fit the pattern of reasoning that designers naturally exhibit. In doing this, we are using Design Space Analysis in yet another way—as a research tool to understand the structure of design reasoning.

3. EMPIRICAL STUDIES OF DESIGN REASONING

Design Space Analysis and the QOC representation give us a useful way to organize the information about the context of reasoning surrounding a design. However, it is not clear how compatible the QOC representation is with the ways designers naturally talk about design. Because our goal is to use Design Space Analysis in the design process, it is clearly important that we understand how it might fit into the ways designers actually work. In this section, we present empirical studies of designers at work, and we use Design Space Analysis to structure the content of their discussion.

We examine two observational studies of professional software designers considering the design of the bank ATMs that we presented in Section 2.2. The first study (called ATM1) has been analyzed in considerable detail, and it is the primary focus of this section. The second study (called ATM2), carried out at the University of Michigan, has had less detailed analysis—We only draw on it for a brief example.

In these studies, we used pairs of professional software designers, so their natural activity of discussion exhibited their reasoning. The pairs had worked together in the past and, thus, did not have to spend any time adjusting to each other. The studies were carried out "in the zoo" (halfway between an artificial laboratory task and uncontrolled free behavior "in the wild"). The designers worked on a fairly natural design problem, but a problem of our choosing, carried out in a meeting room that was set up for video recording.

The problem given to the designers in the first study (Jaimie and Donald) is shown in Figure 7. Their task was to analyze the FATM (which had been proposed in response to queues building up at the SATMs), to critique it relative to the SATM, and to suggest alternative designs if appropriate. Debriefing after the session confirmed that neither designer had ever seen an actual FATM. This task was methodologically attractive, because it enabled a small but complete problem to be tackled in a relatively short time and because it naturally involved rationalization activity.

We recorded the session on video tape. The two designers sat in a room by themselves alongside an electronic whiteboard, which they used heavily. They spent about 45 min on the problem on their own, then about 10 min summarizing their conclusions to us, and then about 15 min on a debriefing during which they told us their backgrounds and experience (both as software designers and as users of ATMs). The debriefing confirmed that they felt the problem and the setting were natural for them.

We transcribed the video tape and categorized the behavior into Design Space Analysis elements. These elements were then structured into a design space using the QOC notation. We used this exercise to help us understand the extent to which the discussion can be represented using QOC and to give some insights into phenomena that do not naturally fit into Design Space Analysis terms.

Figure 7. **The problem presented to the designers in the ATM studies.**

Standard ATM

The National Barklands Bank (NB) Automated Teller Machine (ATM) is a fairly typical ATM. If you want to get cash from it, you would go through the following steps

- Push card into slot.
- Type PIN number when prompted.
- Select *Cash Withdrawal.* (from the several Services offered).
- Type in amount required and press *Enter* key.
- Select *No* (when asked if you would like to request another service).
- Remove card from slot.
- Take cash from drawer, and receipt from slot.

But

The NB bank noticed that long queues sometimes built up at these standard ATMs. They asked their design staff to see if they could speed the process up. Their proposed design (FATM) presents the customer with the following procedure:

The Fast ATM (FATM)

- Select cash amount. (Must be one of six preset amounts).
- Insert card.
- Remove card.
- Type in PIN number.
- Take cash and receipt from drawer.

Your task . . .

You are brought in as design consultants by NB, who would like to know whether you think they have produced a successful design for the FATM. We would like you to analyse the new design and

(1) summarise for us what you feel are the main advantages and disadvantages of the FATM;
(2) suggest any further improvements to the design, or better design alternatives.

3.1. Protocol Analysis

Encoding the Protocol

The video record was transcribed into a verbal protocol with annotations of the nonverbal activity so that the protocol could be understood without referring back to the video tape. Our focus was on analyzing the content of the session and building a coherent representation of the main ideas in the discussion.[3] Asides and redundant remarks were filtered out, and the protocol was segmented into 358 assertions, each of which captured a substantive point in the discussion. The assertions were numbered sequentially to give a convenient index into the session.

Before going into detail, it is worth briefly characterizing the flow of the session as a whole: The designers' discussions ranged over a number of topics, and they tended to go back and forth between them in an unstructured fashion. They jumped straight into the task and started off by trying to understand the differences between the two ATMs described. During this period, they did not identify a good reason for the FATM to use a different order of steps and maintained that the SATM order was better (Assertions 1 to 103). They then moved toward a design solution that attempted to reduce the number of steps and time required for each step. In attempting to resolve the conflict between maintaining a range of services and reducing customer transaction times, they devised a proposal for a switchable ATM that the bank staff could set into a fast-cash mode, thus restricting services during busy periods (Assertions 104 to 214). This was followed by a review of the design (Assertions 215 to 260). They realized that they did not fully understand why queues were building up and discussed possible reasons (Assertions 261 to 303). Finally, they went over the details of their proposed design (Assertions 304 to 358).

Categorizing the Assertions

One way to get a feel for the designers' style of reasoning and its relation to Design Space Analysis is to categorize the assertions into something close to the QOC elements. The vast majority of the session consisted of assertions about the substance of the design problem; only 4% of the assertions were nonsubstantive (e.g., discussion of how to make use of the whiteboard). We classified the substantive assertions into three broad categories—options, issues, and justifications—which are related to the QOC elements in ways we discuss shortly. Inevitably with data of this sort, reliable classification is difficult. Categorization relies crucially on maintaining and interpreting the context in which the assertions were raised. The grain of analysis is such that the discussion of one point often spreads across many assertions, and conversely a few assertions (about 4%) were so ambiguous that they had to be classified into two categories (see also, MacLean, Bellotti, & Young, 1990).

We observed that the designers talked frequently about specific Options. In

[3] Olson and Olson (1991) reported an analysis of the ATM1 study that focused on the meeting dynamics of the session. Their goal was to analyze present practice to look for opportunities for technology intervention. Their analyses focused on the coordination of group activity (e.g., stating goals, agenda setting, activity tracking) and on the development of issues (e.g., stating the goal and generating, structuring, evaluating, and selecting ideas).

all, 38% of the assertions were categorized as Options in the QOC sense. It was relatively easy to categorize these, for example:

> Well my favorite would be just to have a "fast cash" button at the point where you said "select cash withdraw." (Assertion 107)

The Options were usually discussed in isolation, with relatively little structure linking them; that is, there were no Questions to structure the Options. Questions in the QOC sense were hardly ever stated; the nearest example was:

> Is there any way we can improve on preset amounts? (Assertion 190)

There were some questions of a yes-or-no variety, such as:

> Do you want the receipt? (Assertion 44)

Such a question does not provide much help in exploring a space of possible Options. However, we categorized 8% of the assertions as "issues," which were the nearest thing to Questions. In addition to examples like those just shown, these were more akin to issues in the IBIS sense, for example:

> What if you asked for fifty quid, and you've only got thirty quid in your bank account? (Assertion 339)

Finally, 54% of the assertions were categorized as "justifications." Justification is used here broadly, covering Criteria, Assessments, and arguments, as well as other forms of justifying. For example, the following captures an Assessment that relates to an explicit Criterion:

> Well, they've speeded it up by taking away the other services. (Assertion 142)

Some explicit tradeoffs between different Criteria were considered:

> It is basically because otherwise you trade off speed against security. (Assertion 333)

On the other hand, a range of more general forms of argument was used extensively. Some consisted of questionable rationalizations, such as:

> If you're going to spend three quarters of the money, you might as well spend a hundred percent and have two full blown machines. (Assertions 138 to 140)

Some arguments appealed to analogy with other situations:

> You could say five items only like in supermarkets or whatever. (Assertion 187)

Some arguments utilized small usage scenarios:

> You've got to remove the card before you take the cash. (Assertion 161)

Finally, some arguments attempted to build an ad hoc theory of why queues might be forming:

> Is it because there's loads of people, or is it because people are fumbling with their cards? (Assertions 266 to 267)

In fact, 18% of the assertions were related to this ad hoc theory, and we return to it shortly. We discuss the more general forms of justification in Section 4.

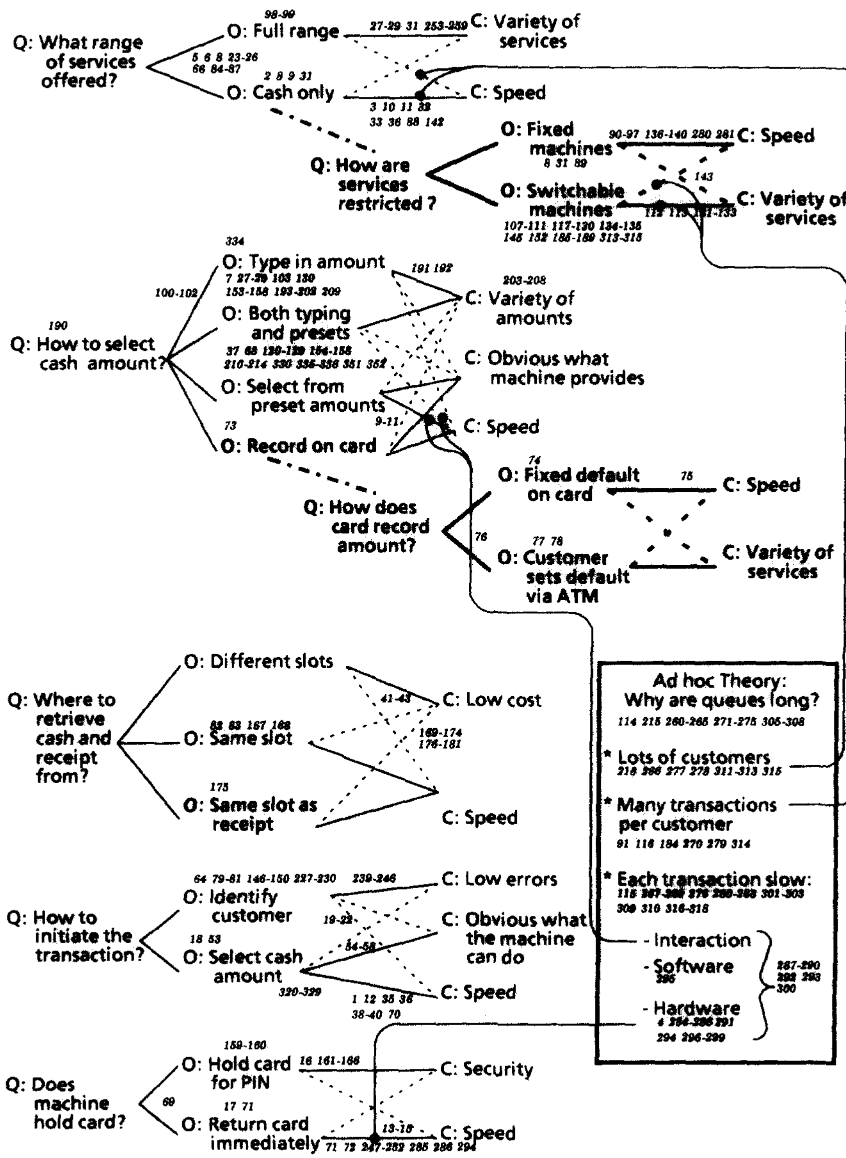
Representing the Reasoning Structure

The categorization of assertions shows that a great deal of the discussion in the session looks as if it has the flavor of Design Space Analysis. However, it gives us no sense of the structure of their reasoning. To impose some structure, we begin with an a priori analysis of the design space comparing the SATM and the FATM—the one presented in Section 2.2 (see Figure 5)—and we see how the discussion fits into that framework. This is presented in a QOC diagram in Figure 8.

Assertion numbers (in italics) are placed in the diagram in Figure 8 to show where the discussion talked about the various points represented. The position of the assertion numbers on the diagram gives some idea of how each assertion was categorized. Assertion numbers placed directly adjacent to Options or Criteria indicate explicit reference to them. Those placed in the areas to the left of Options indicate discussion of more general issues, and those placed on the Assessment links between Options and Criteria indicate discussions involving justification.

More than three quarters of all the assertions in the protocol are represented in Figure 8. At times, there is not a one-to-one mapping between assertions and points in the diagram, so some assertions appear twice. Of the

Figure 8. **QOC diagram representing the structure of the discussion of the ATM1 design session. This diagram is based on the a priori analysis diagrammed in Figure 5. The bold items are new. The italic numbers represent the assertion numbers in the protocol.**



assertions not represented here, most cover details that are not captured in this diagram (e.g., the possibility of having an interface that did not use a CRT screen).

Several points can be noted from this representation of the discussion:

1. Some discussion takes place around nearly every point we had in the earliest analysis.
2. New items (indicated in bold) that were not in the a priori analysis emerged in the discussion. The new items mostly involve more detailed exploration of the consequences of Options at the top level.
3. A considerable amount of discussion revolved around the ad hoc theory of why queues might be forming.
4. Figure 8 does not emphasize the order in which items were discussed, although it can be inferred from the assertion numbers. It is clear that the logical structure of the design space represented here does not match the chronological structure of the design discussion. Items are often revisited several times throughout the discussion (see Olson & Olson, 1991).

3.2. Some Phenomena in Design Reasoning

We have established the relationship between the content of the design discussion and the elements of Design Space Analysis. There is clearly not complete overlap, but there is sufficient correlation to suggest that it would not be unreasonable to expect designers to work with the QOC concepts. However, there are also a number of areas where the overlap is relatively weak. Some suggest that Design Space Analysis could perhaps improve the effectiveness of design practice. Others suggest ways in which a process to support the use of Design Space Analysis in design practice could be developed. We now discuss some of these.

Ad Hoc Theories

As the session goes on, the designers realize more and more that they do not know why the queues are building up at the SATMs, and they eventually agree that this is essential information to be able to tackle the design problem adequately:

> What's causing the long queue. Is it people just going through these steps, or is it people adding options to other services, and then using the other options? (Assertions 114 to 116)

The information is not given in the problem statement, and they eventually ask the experimenters for more information but are told that no more is available. They then spend more than 5 min in a very analytic phase building an ad hoc theory of why queues might be long, based on their own knowledge of the world. This is summarized in the box in the QOC diagram in Figure 8. Three classes of reason for queues being long are shown: **lots of customers; many transactions per customer** and **each transaction slow.** (A fourth class of reason to do with user errors and lack of knowledge is not represented.) **Each transaction slow** further subdivides into three subsidiary reasons: **hardware** reasons (e.g., reading the card and counting money), **software** reasons, and reasons to do with the **interaction** between the customer and the machine. As they are developing it, the designers use the ad hoc theory to help them understand reasons behind the design of the original FATM and to revisit some of their own design.

> People doing lots of transactions. Transactions take a long time. And that's what they've worked on, that's what they've solved. . . . That was that FATM. (Assertions 279 to 281b)

> People not knowing what to do, is to make a simpler user-interface so that's that. Too many people is to provide more machines, with a potential for a "fast light" machine. (Assertions 309 to 313)

Figure 8 shows some of the ways in which the details of the theory are used to argue about the Assessments represented within the core QOC notation. Given the nature of this ad hoc theory, it is not surprising that all of the reasons given relate to Assessments of the **speed** Criterion. The status and use of this kind of reasoning are similar to the idiosyncratic views designers have of users reported by Hammond, Jørgensen, MacLean, Barnard, and Long (1983). It is based on their own experience and insights rather than on more objective information—of course it is all too common in design that all relevant information is simply not available, as was the case on this occasion. This kind of reasoning has the status of an argument, which backs up the QOC representation of the design space, as we discussed earlier. Its role is also very similar to the role of scientific theories that we discuss in Section 4.

Design Biases

There was a strong tendency for the designers to look for evidence to confirm their initial biases—a well-known phenomenon in the psychology of thinking (e.g., see Wason, 1968). This can have the effect of dismissing possible good Options, because no positive Criteria are considered, or of proceeding with poor Options, because no negative Criteria are considered. It is interesting to note that in keeping with Wason's observations, and in spite of the dangers, there is no evidence of the designers changing their minds about any of the possibilities they considered during the session. We saw in the previous section that, when they questioned why queues might be building up, they used their analysis to confirm that their novel design was good. Even when this line of argument helped them recognize a problem that the FATM had addressed ("too many transactions"), they did not reevaluate it as a solution. We do not intend to suggest that there were flaws in the argument they presented—We simply want to remark on the fact that they never directly asked what the down side of their proposed solution might be.

On the other side of the coin, there were aspects of the proposed design solution given to them (i.e., the FATM) that they seemed resistant to accept from the outset—for example, the initiation of the transaction by selecting the cash amount:

> Yes, and why select the cash amount first? The problem with that is what happens if someone selects the cash amount and then goes off somewhere else. You're going to have to have some sort of cancel button or something, or I suppose you could just press another amount. I would have thought they would, um, worry people. (Assertions 18 to 22)

They identify a possible problem with the interface and immediately think about how it might be circumvented. They come up with plausible solutions but, nevertheless, do not appear to reevaluate the problem in light of them. Later, when discussing the order of steps more generally, they suggest another problem:

> I think there's a disadvantage. In fact you could write that down. If you have a different order on different machines, this is going to, er, you know, people are going to have to learn two different ways of doing things; they're not going to like that. (Assertions 39 to 40a)

Later, it is clear that they have formed a firm opinion of the relative merits of the two machines when they refer to the SATM as the "good one":

> So having one good one, and one only fast cash doesn't really seem to be a good idea does it? (Assertion 137)

The main point of these examples is that the designers make no explicit attempt to explore reasons why the FATM might be a good design, in the same way as they did not attempt explicitly to question why the switchable ATM might be a poor design. It is important for an objective design discussion to consider both pros and cons. However, we must also recognize that it can be difficult to look for arguments, or indeed evaluate available information, to counter current beliefs.

Emergence of New Designs

The most innovative result of the ATM1 design session was the proposal for a "switchable" ATM, that is, an ATM that can be switched from full to restricted service. Let us trace the development of this novel design to see how it emerges out of the reasoning process. Considering the implicit Question **what services offered?**, the Option of a **full range** of services versus some sort of restriction is discussed at various times throughout the session, both in general and in moving toward a new switchable ATM. The FATM's **cash only** Option (see Figure 6) is rejected fairly early on:

> So those are the restrictions; you've got to go for cash, and you've got to go for a "select amount", and that's supposed to save heaps of time. Don't believe it somehow. (Assertions 9 to 10)

It is gradually replaced by a notion of a **restricted range** of services. The first step is to incorporate the fast-cash notion into the SATM interface:

> Well, there are a lot of different ways of doing it. Well my sort of favorite would be just to have a sort of a fast cash button at the point where you said "select cash withdraw" . . . would be this idea where you had sort of a preliminary screen come up which would just have five different amounts of fast cash, or do something else. (Assertions 106 to 109)

The designers are then very satisfied at being able to give access to "all the other things" while not having more buttons to worry about:

> And then you'd have exactly the same number of buttons as on this one, or actions, and yet you'd have access to all the other things, and then you could do them on the same machine. (Assertions 110 to 113)

However, they realize that if they give access to all other possible services at the time, queues may build up with customers carrying out multiple transactions. They come up with a notion of having some **fixed machines** that offer only cash alongside the **full range** machines. Note the use of the analogy with a fast check-out in a supermarket for customers who only want a few items. (We discuss such roles of analogy more fully in MacLean, Bellotti, Young, & Moran, 1991.)

> What you could have is just like you've got in the supermarkets. If you've got say three of these machines all the way along, you could actually have a little light above it which said "Cash withdrawal only" and then you could sort of . . . we remove the other options, yes? (Assertions 117 to 120)

Eventually, they come up with the notion of a single **switchable machine:**

> New Design "A", One; same machine for ATM and FATM. (Assertion 145)

They then combine these ideas and flesh out a design that uses a light to signify which mode the machine is in:

> So we could use (waves his hands around in the air, and they both laugh) a "Fast light" (he writes " 'fast' light-operated . . ." presumably by the bank staff. Yes, and you could say, like five-items-only in supermarkets or whatever. . . . there's be a light above it and also other options wouldn't appear on the screen. (Assertions 185 to 189)

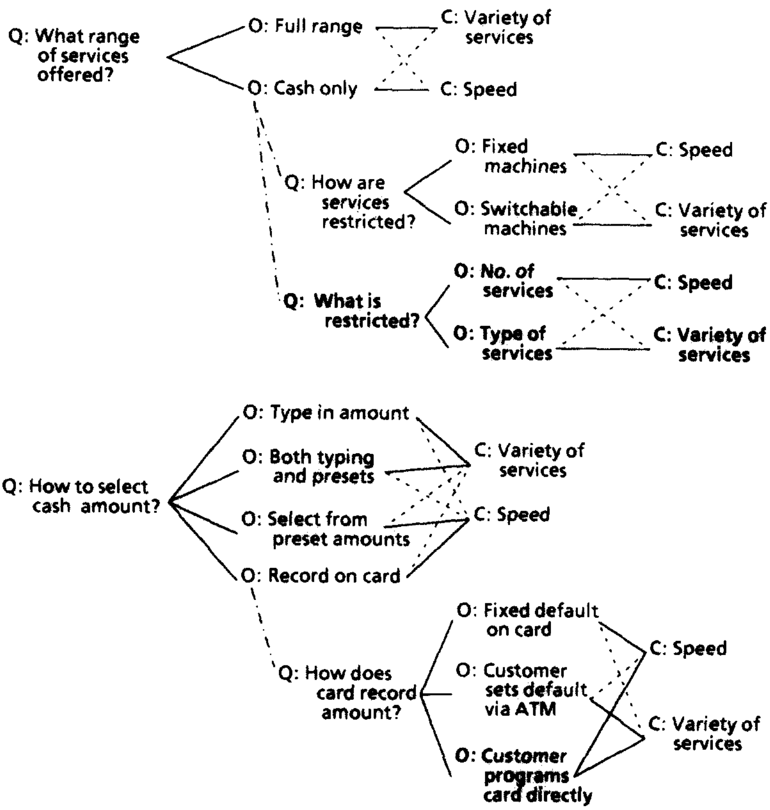
Toward the end of the session, they check their design out against some of the reasons for queues building up:

> [If the problem is] Too many people [the solution] is to provide more machines, with a potential for a "fast light" machine. [If the problem is] Lots of transactions per person [the solution] is again this fast light thing. (Assertions 311 to 315)

Note how the new design gradually emerges throughout the entire session. There is a continual interplay between analysis and innovation, and the various phases are interspersed with much discussion of other aspects of the design. It is interesting to note the conciseness of the representation of the key points of this discussion in Figure 8. All of the substantive points, apart from the details of the fast-light design are summarized under the **what range of services offered** Question.

A second example comes from the ATM2 study, where the problem was left more open ended by describing only the SATM and the problem of queues building up and asking for a new design. The designers in the ATM2 session quickly got into exploring ways of restricting services. Figure 9 is a QOC representation of some of their exploration. They quickly move beyond the kind of restrictions suggested by the ATM1 designers and discuss restricting services either in terms of the number of services customers are

Figure 9. **Part of the QOC analysis from the ATM2 protocol (new items are indicated in bold).**



allowed (e.g., a customer can only use one service during a single transaction) and/or by restricting the services available to a subset (e.g., cash only).

The ATM2 designers provide a third example of innovation toward the end of the session. One of them suddenly sees a connection with the way a local bagel store handles its lengthy queues while customers are waiting in line. It has an employee work along the queue, explaining the choices available and helping customers fill in their orders on a form. The customers hand over their forms when they reach the service counter, enabling their requests to be processed more speedily. Familiarity with this arrangement prompts a similar suggestion for the queues outside the bank. (Note again the powerful role analogy plays in suggesting the solution—see also MacLean et al., 1991.) Although the notion turns out not to be sensible in detail for the ATM problem, it plays a crucial role in leading the designers to generate their most innovative proposal—the idea of having "active" bank cards that customers

program directly while they are waiting in line. This Option is, in fact, very similar to the **defaults** on a card that the ATM1 designers considered briefly and then abandoned (cf. Figure 9 with Figure 8). An interesting feature of this Option is that it resolves the tradeoff between the Criteria of **speed** and **variety of services** as it satisfies both of them. It is intriguing to note how when the output from the two design sessions is combined in this way, yet more powerful solutions emerge. The notion of **defaults** could have prevented the ATM2 customers from having to program the card each time they used an ATM, and the notion of a **programmable card** might have made the **defaults** more attractive to the ATM1 designers.

3.3. Summary of Findings

We can learn a lot by looking at design activity from a QOC perspective. When we relate the contents of the protocols in both studies to a single QOC representation, there is considerable overlap between what the designers talk about and the concepts we use in Design Space Analysis. The data suggest areas where the use of Design Space Analysis could be beneficial. We, not the designers, provided the logical structure around which the material was organized as there was little relationship between the chronological structure of the session and the logical structure of its content. If designers used QOC to structure their deliberations, we believe that they could improve their reasoning by working with a structure more logically related to the design problem. The success of itIBIS for structuring design discussions (see Conklin & Yakemovic, 1991 [this issue]) suggests that this claim is not unreasonable. In the data of ATM1 and ATM2, searching Questions that might have helped with structuring the design space were barely represented. At best, the discussion seemed to be structured around individual Options rather than groups of related Options. Exploration of the design space was haphazard—Some points were revisited many times during the session, whereas others that looked promising were dropped. We saw that pros and cons were not thoroughly explored, leading to the potential for biased judgments to be made.

If designers could provide a logically structured output, such as we used to summarize the session, it could make the reasoning behind the design clearer to other people and it could help the designers to reason about the design while they were creating it—for example, by keeping track of what they discussed and by helping them to see missing or inconsistent parts of the design representation. We point toward how Design Space Analysis might be used to encourage such reasoning in Section 5.

Much of the talk in the ATM1 session was shown to be devoted to assertions that justified possible design options. However, many of the justifications (including the ad hoc theory) do not fit comfortably into the QOC concept of Criteria. Section 4 expands on the concept of Criteria and considers how a range of different kinds of justification relate to Design Space Analysis.

4. JUSTIFICATION IN DESIGN SPACE ANALYSIS

We have seen a variety of instances and issues of justification—the argumentation used to evaluate design alternatives. In this section, we discuss a variety of topics under this important notion. Section 2 introduced Criteria as the basis for evaluating Options, beginning with the Assessment of individual Options against individual Criteria. In Section 3, we observed that designers use a variety of forms of justification during design, not all of them appealing to Criteria. We also saw in the examples of Section 2 that there are dependencies across design decisions and that these dependencies provide another basis for justifying decisions. In this section, we take a closer look at these various kinds of justification and explore their place in Design Space Analysis. We begin with a closer look at Criteria.

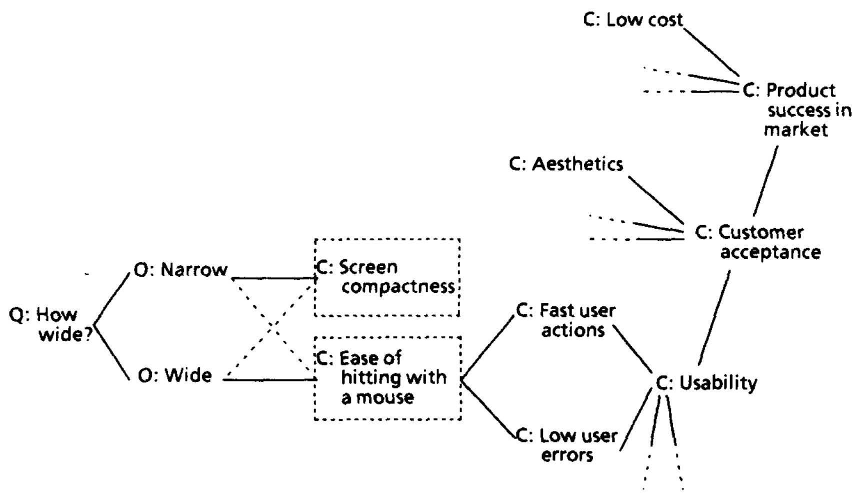
4.1. Understanding Criteria

The examples we have used so far might suggest that Criteria are fairly simple well-defined entities. In fact, it can take considerable effort to find suitable Criteria and phrase them in such a way that they achieve the desired impact. Several issues have to be taken into account. It is important to characterize a Criterion at a suitable level of abstraction relative to the Options being considered. The concept of a Bridging Criterion is used to talk about how the Criterion can be worded so as to make its relationship to the local design space clear while also being clear about the more general Criteria to which it relates. If the Criterion becomes too heavily entangled with details of the local Options, however, its status can become unclear, so it is important to understand some basic properties of Criteria that help make them useful. Finally, Criteria play a crucial role in understanding tradeoffs in the design space. If appropriate ones are to be found, key issues can be characterized as simply as possible.

Bridging Criteria

Just as Options can be justified by being Assessed against Criteria, one Criterion can be justified by reference to another. Criteria differ in their degree of generality, and the more specific Criteria can inherit their justification from the more general ones. This is portrayed in Figure 10 as an extended QOC diagram. It is more convenient to assess the Options against

Figure 10. **Moving from Bridging Criteria to General Criteria. The Criteria in the dotted boxes are Bridging Criteria. The Criteria become more General toward the right.**



Criteria that are specific to the design being analyzed instead of having to spell out the justification all the way back to a set of very general Criteria. One example in the case of the scroll bar is the Criterion of **ease of hitting with the mouse.** Such Criteria are called *Bridging Criteria* because they bridge between specific aspects of the design (e.g., the use of a mouse) and broad, general Criteria such as speed and accuracy. Unlike General Criteria, Bridging Criteria are typically of narrow applicability and are invented for their relevance to a particular class of designs. The Criterion of **ease of hitting with the mouse** arises not from considerations of usability in the abstract but, rather, assumes that decisions have already been made to use a mouse and a scroll bar of a certain kind. In return for their lack of breadth, Bridging Criteria offer the advantages of being easy to work with and of encapsulating a possibly complex set of arguments and interdependencies into a single entity within the analysis.

Bridging Criteria are justified by exhibiting their impact on more general Criteria. Thus we see, in Figure 10, **ease of hitting with the mouse** as being relevant to the width of the scroll bar by arguing that it contributes in turn to the more General Criteria of **fast user actions** and **low user errors.** As with several other aspects of Design Space Analysis, there is no fixed stopping point to this process; that is, there are not necessarily any fundamental Criteria for which further justification cannot be asked. Rather, what happens as we ask for justification repeatedly is that we get driven to broader and broader Criteria, which have less and less direct relevance to interface design. We see that **fast user actions** and **low user errors** contribute to **usability** (which is surely true, although not very helpful), which in turn contributes to **customer acceptance,** which in turn contributes to **product success.** Thus, as the Criteria become more general, they tend to become the province of the financial analyst and marketing strategist more than of the interface designer.

Some Properties of Criteria

It is hardly practical to try composing a formal definition of what is meant by a Criterion, but it is worth listing some properties that help identify appropriate Criteria. We illustrate example properties by exposing a stage we went through when we were first analyzing the scroll bar. For a while, we tried to work with a notion of **wide for mousing,** which was a sort of composite of what we call in Figure 3 the Option **wide** and the Criterion **ease of hitting with a mouse.** The notion was evidently responding to the intuitive idea that it is a good thing to make the scroll bar wide, so that it can be selected easily with a mouse. But attempting to use it as a "pure" Criterion led to a variety of problems that were resolved only when it was broken into the two separate components.

Observations that suggest relevant properties of Criteria include:

1. A Criterion measures a property of the artifact that the designer *controls only indirectly* by exercising choices over Options. (Options are under the direct control of the designer, because they represent possibilities that the designer can select.) The would-be Criterion of **wide for mousing** fails because one and the same notion appears to offer both direct control (i.e., over the width) and a measure of an aspect of evaluation (i.e., the "mousability"). The Criterion **ease of hitting with a mouse** is acceptable because there is already a commitment to use a mouse in the design space being considered. (However, if a purpose of the design space was to explore different pointing devices, a mouse would be an Option under the control of the designer and so **ease of hitting with a mouse** would not be an acceptable Criterion. In the spirit of the discussion in the previous section, a more General Criterion such as **ease of hitting** would be required.)

2. A Criterion must be *unconditional* in the sense that, other things being equal, the greater the extent to which the Criterion is met, the better the design. **Wide for mousing** fails this test. It suggests "the wider the better," but it is clear that there is a negative impact of a wide bar on the use of screen space. On the other hand, **ease of hitting with a mouse** is acceptable because it is clearly a desirable characteristic and does not prejudge a class of solution that is likely to have negative consequences. For example, a small

target that attracts the cursor when it gets near it could be another plausible solution supported by the Criterion.

3. A Criterion must be *evaluative;* that is, it must be a measure of some property of the artifact, with a definite sense of higher Assessment values being better. Notice that the notion of **wide for mousing** is unclear in this regard: The width in itself is neutral, and only the mousability is evaluative.

4. As an extension of Observation 3, it is convenient to think of a Criterion as *potentially yielding a quantitative value,* even if only on an ordinal scale. One could even consider assigning actual numbers to Assessments, combining them as if they were Expected Utilities, and applying the apparatus of mathematical decision analysis. We do not judge that course worth pursuing in detail, but taking the idea of numerical values as a metaphor proves useful to help sharpen the concept of Criterion. The notion of **wide for mousing** certainly fails this test. What would its putative value be measuring? Would it be a width, for example, expressed in millimeters, or some measure of mousability?

Tradeoffs Between Criteria

Design is always a matter of working through conflicting constraints, and one of the most important uses of Design Space Analysis is to understand the tradeoffs between different requirements. We are now very familiar with the characteristic visual pattern of parallel solid links and crossing dashed links in the QOC diagrams, which expresses the simplest tradeoff structure: two Options against two conflicting Criteria. Shum (1991) pointed out the importance of a design notation clearly expressing such functional roles as this.

To make the tradeoff structure clear, it is important to choose the Options carefully. A characteristic example is the case where a Question is about a continuous-valued parameter, such as **how wide should the scroll bar be?** We noted in Section 2.1 that we chose the qualitative values **wide** and **narrow** (rather than, say, 2 mm, 4 mm, 8 mm, 16 mm) to make clear the tradeoffs. Once these are clear, other Questions can probe other aspects of the width (e.g., **should the width be a multiple of the grid size?**). Eventually, a small set of specific candidate values can be evaluated.

Another interesting property of Options is that they are often regarded as characteristic representatives from the set of possible Options rather than as specific candidates. An example is found in the ATM1 study, where the designers were trying to decide how many values to offer for the preset amounts of money that can be withdrawn from the ATM:

> They probably are twenty and fifty—they are the most common ones. In fact that would probably do, almost. Just a big red button and a big blue button. (Assertions 198 to 200)

Our focus on understanding tradeoff structures implies that this statement is not to be taken literally as suggesting that the ATM should have one big red button and another big blue one but, rather, as representing an Option of "very few." It is treated by the designers almost as a caricature of a possible Option and serves effectively to make a clear distinction between the possible amounts. Similarly, the "twenty" and "fifty" cash amounts should not be read as a concrete proposal of those particular values but, rather, as a place holder for "two appropriate amounts." The concrete wordings of the Options help the designers retrieve relevant considerations but are clearly intended as representatives rather than as specific commitments.

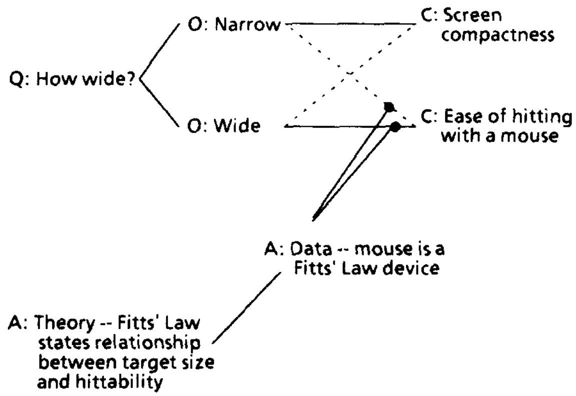
4.2. Further Justification

An important feature of the Assessments of Options against Criteria is that they can be challenged and that justifications can be given for and against them. We saw these represented in Section 2.3 as arguments that support or challenge the Assessments. The relationships between specific and general Criteria that we saw in Section 4.1 are also subject to justification in the form of arguments. The justifications embodied in arguments appeal to evidence of various kinds: logical, theoretical, empirical, and so forth. In this section, we explore some of these different types of justification and how they relate to Design Space Analysis.

Theory and Data

One way to justify is to appeal to empirical data, accepted theory, or both. This is indeed the main route by which theory and data are incorporated into Design Space Analysis. A simple example, diagrammed in Figure 11, is provided by the Assessments in the scroll bar analysis between the width Options and the Criterion **ease of hitting with a mouse.** The Assessment is that a wider scroll bar is more easily hittable. (Note that this is a relative assertion that is represented by a pair of Assessments, one positive and one negative, from the Criterion to each of the width Options.) How do we justify this Assessment? We can appeal to some experimental work showing that a mouse used to hit an on-screen target can validly be treated as a device to which Fitts's Law applies (Card, English, & Burr, 1978), and we can argue that the scroll bar can be considered such a target. Further, we can appeal to Fitts's Law itself (see Fitts, 1954), which is an empirical law giving an indication of the difficulty of hitting a target relative to the amplitude of the movement and the size of the target. As it turns out, in this case, the evidence does more than merely support the claim. It also fleshes out the relationship by providing, should we want it, a means for quantifying and calculating the effect of a scroll bar's width on the time it takes to hit it.

Figure 11. **Data and theory used as arguments to support Assessments against a Criterion involving the ease of hitting an on-screen target.**



In most cases no existing theory or relevant data will be available. The designers will have to construct an approximate explanation by formulating an ad hoc theory or collecting some "quick and dirty" data in order to produce a convincing Design Space Analysis. We saw an example of this kind of theorizing in the ATM1 design session in Section 3, where the designers proposed an ad hoc theory that the long queues at ATM machines could be due to customers carrying out too many transactions at the machine; they proposed a new machine that could be set to allow a **restricted range** of transactions (see Figure 8), which they assessed to be better on **speed**. They used the ad hoc theory as a form of argument to justify this Assessment (this is shown in Figure 8 by a link from the ad hoc theory argument to the Assessments against **speed**). It should be noted that the correctness of the **restricted range** Option depends crucially on the correctness of the ad hoc theory. If most customers are not making multiple transactions, then restricting the number of allowed transactions will have little effect on the speed.

Models, Analogies, and Metaphors

Another form of justification is conformance with a model, analogy, or metaphor based on something outside of the design itself. In some cases, the analogy may apply broadly across much of the interface, as with the notorious "desktop metaphor." More frequently, it applies to just certain aspects of the interface. Appeals to analogy as part of the design discussion have considerable effect on the process of design and the generation of ideas, as we saw in Section 3. For the present focus on justification, the important point is that appeals to such conformity are used by designers to justify particular Options.

We discuss the roles analogy can play in design more fully in MacLean et al. (1991), so we only give two brief examples here. First, analogies can have the same role as theories and data in a QOC representation. For example, if we are asked to justify why the programmable ATM card will speed up use of the ATM, we can appeal to the bagel store analogy (described in Section 3) to show that a similar separation of placing requests and receiving services works in another context. For the purposes of using such a justification, it of course does not matter whether the original inspiration for the design came from that analogy or whether the analogy was purely a post hoc justification. Second, an analogy can be the source of more detailed justification. Consider developing a QOC analysis to support the source of the analogy (e.g., the bagel store). This could be transferred in suitably adapted form to the target of the analogy (the ATM). The technique deployed in the bagel store works because it allows all the decision making and interaction to have been completed by the time the customer reaches the counter. This fragment of explicit rationale could be transferred intact to the context of the ATM. We thus represent why the analogical situation works—not just claim that it works.

Scenarios

Typically, a scenario involves envisaging what it would be like to use the artifact being designed. An important property of scenarios is that they generate a context of use that emphasizes variables not apparent from a static description of the artifact. As with analogies, scenarios can justify a design in two ways. At the holistic level, a scenario can justify a particular solution by demonstrating that an envisaged mode of use will work. Goel and Pirolli (1989, p. 28) observed their subjects making extensive use of what they called "scenario immersion" for the evaluation of design possibilities. We noted several occurrences of scenarios being used in this way in the ATM1 and ATM2 studies. The ATM2 designers, for instance, undertook a fairly comprehensive usage scenario where they imagined the operations a user would have to execute to select a service, enter a PIN, select an amount of cash to withdraw, and so on. At a more detailed level, a scenario can evoke new Criteria that the design should meet. This is most likely to happen when a scenario shows up flaws in a proposed solution. For example, in visualizing the steps required to use a new ATM, the designers might reach the point where the customer wants to select another service, only to realize that the proposed interface neither displays the services available nor provides a means of reaching them. Such a scenario highlights two things that are expressible in QOC: It suggests additional Options to resolve the problem and a new Criterion of ensuring that relevant facilities are accessible. These uses of scenarios are consistent with the walkthrough methodologies proposed by Lewis, Polson, Rieman, and Wharton (1990) and by Lewis et al. (1991 [this issue]). The main difference is one of emphasis—We are focusing on the representation of the design space, and Lewis et al. are focusing on a scenario-based design process.

4.3. Relations Across the Design Space

We turn now to the issue of the interrelationships between different design Options. In Section 2, we saw that Options are structured into a tree of Questions and Options, showing which Options are relevant to each Question and what Questions are relevant to choosing certain Options. But we also saw in the examples in Section 2 that no coherent or satisfactory design can result from considering each Option in isolation. Instead, each Question has to be answered in the light of the answers already given (perhaps tentatively) to other Questions. That is to say, the evaluation of Options is a function not only of the local decisions but also of the nonlocal interdependencies between Options. This section describes how Design Space Analysis deals with these interdependencies.

Internal Consistency—Generic Questions

One kind of justification frequently offered for Options appeals to a notion of "internal consistency." In Design Space Analysis terms, this means that a Question in one part of the design space should be answered in the same way as similar Questions elsewhere. Consistency can be imposed on user interfaces by formal rules, such as the grammar rules of the Command Language Grammar (Moran, 1981) or the Task Action Grammar (Payne & Green, 1986). A simple example of a consistency issue in the scroll bar domain concerns the assignment of functions to mouse buttons. For left and right movement, there is an obvious compatible mapping onto the left and right buttons, but for vertical movement the choice is less clear. Suppose it is decided (on some grounds, perhaps with supporting arguments) to use the left button for scroll up and the right to scroll down. If up and down movement is to be carried out with the mouse in another part of the design, then clearly the same mapping should be used. This is represented in Design Space Analysis by considering some Questions as being generic, and the Options considered and the Criteria used reflect the generic nature of the Question. The decisions made for these Questions are then considered as generic decisions. A specific design Question can be recognized as being an instance of a *Generic Question,* and it would "inherit" the generic decision. (Of course, it is always possible to have an exception to the rule, but this would involve finding good local reasons for not taking the generic decision.) Therefore, having a class structure of generic Questions is the way this kind of consistency is represented in Design Space Analysis.

Cross-Question Constraints

Another form of interdependency is where an Option chosen for one Question directly affects the choice of Options for another Question. It can easily happen that the relevance of a Criterion or the existence of an Option depends on a decision made elsewhere in the space. We saw this first in the scroll bar analysis in Section 2.1, where the decision of whether to choose an appearing scroll bar strongly depended on just how it was decided to make it appear. Another example is from the discussion of Bridging Criteria in Section 4.1, where we saw that the Criterion **ease of hitting with the mouse** applied to the Question **how wide should the scroll bar be?** assumes that it has already been decided to use a mouse to access a scroll bar. Bridging Criteria keep this dependence implicit—it being precisely the job of a Bridging Criterion to summarize the impact of a number of considerations in a single entity—but there are also many cases where the dependence is best made explicit.

A dramatic example of cross-Question constraint is seen in the basic decisions for a simple text editor. Two Questions concern **what names should be given to the editor functions** and **how should the functions be invoked by the user** (e.g., typing commands, invoking menus, clicking on icons, etc.). These two Questions appear to be independent. But what if the Option chosen for invocation is to type single-letter abbreviations? A constraint suddenly appears that the names chosen must have different initial letters! This kind of constraint acts in many ways like a Criterion, except that it *must* be satisfied for the design to be coherent. We call something that has this kind of impact on another part of the design space an *Export.* In the part of the design space where its impact is felt, we refer to it as an *Import.* This gives a convenient way of representing each part of the design space separately without having to keep track of explicit links between them. The main role of Exports and Imports is to help simplify the representation by allowing the analysis to be broken into modular pieces, with the Exports and Imports representing interdependent assumptions between modules.

Global Impact of Criteria

Criteria themselves serve as one way of representing relationships across the design space, because many Options in different places can be influenced by the same Criterion. In our own Design Space Analyses, we often find it helpful to draw up a single list of the Criteria appealed to. When elaborating a new part of the design space, such a list acts as a menu suggesting Criteria that may be relevant and encouraging the use of existing Criteria where appropriate (rather than creating new ones). In cases of a tightly determined design, we typically find that just a small number of Criteria has a pervasive influence on the design, being appealed to from many different places. In our analysis of the FATM rationale, for instance, because the purpose of the redesign was to deal with each customer more quickly, the Criterion **speed** is much in evidence, participating in tradeoffs against a variety of other Criteria in different places.

Criteria can also have a more global impact on the design space. The relative emphasis given to different general Criteria is a major determinant of the overall style and orientation of a design. In this respect, Criteria have a role similar to that of requirements in Newman's (1988) analysis of interface style. For example, if **provide feedback** is made more important than **response speed**, it can have a big effect on the kind of user interface that results. Similarly, one can see how giving a lot of weight to Criteria concerned with usability and ease of learning can lead to a general-purpose, easy-to-use interface such as that of the Apple Macintosh, whereas giving greater prominence to Criteria having to do with the close match of the design to the requirements of a particular task, and an emphasis on the efficiency of usage by trained users, can lead to specialized interfaces such as those found in airline booking systems.

5. CONCLUSIONS

For the most part, this article has concentrated on a QOC representation as a product (or at least a coproduct) of design. We illustrated the basic QOC elements in Section 2. We showed how they relate to the kinds of discussion that take place in an unstructured design meeting in Section 3, and in Section 4 we discussed several design-related issues of varying degrees of formality and showed how they relate to Design Space Analysis. We believe that it is critical at this stage of design rationale research not to confuse product and process. It is essential that we be clear about the kind of product we are trying to produce before we suggest processes for creating it. We hope that this article has served to articulate the nature of a QOC representation clearly and has shown how it relates to a variety of the more general design-related concepts with which we are familiar.

It is worth noting that a major contribution of this article has been in using Design Space Analysis as a technique for understanding designs through analysis (in Section 2) and for characterizing some aspects of design activity (in Section 3). The contributions at this stage are research techniques as much as they are techniques for use in design. Clearly, however, our goal is to develop techniques that can be used in the design process. It is no doubt obvious that an impediment to other people using our approach is that we have placed little emphasis on *how* to go about carrying out a Design Space Analysis.

Our primary focus in this article has been on the properties of the notation and the resulting design representations. However, we are beginning to consider the *processes* involved in creating a Design Space Analysis. There is no strict methodology for creating a design space, such as a top-down sequence. To try to follow a strict procedure does not work. An analysis is developed in all places at once by a mixture of inspiration and reflection, as ideas pop up, get understood, and fit into place. On the other hand, the process of developing a rationale is not random. There are systematic steps that can aid the process enormously. One approach (derived from our own analysis work and our observations of designers at work) involves focusing on parts of the QOC representation in principled ways to see how to augment it. We can formulate such principles as *heuristics* for would-be analysts to guide them in building QOC representations. The Appendix gives a brief overview of this approach to creating Design Space Analyses and gives examples of such heuristics.

We hope that the ideas we have put forward are useful in their present state insofar as they express a representation for design that can be picked up fairly easily by others. As we have seen, the semiformality of the QOC representation means that Design Space Analysis can be adopted, even without computer-based tools, simply by using pencil and paper. We hope that both researchers and designers will find it useful as a way of thinking about design, doing design, reasoning about design, and recording and communicating the arguments behind design. What is needed now is an examination of the practical implications of Design Space Analysis by groups other than ourselves, and indeed we know of several investigators who are exploring aspects of Design Space Analysis in teaching, research, and design. We look forward to seeing how the ideas presented here, inevitably underspecified, are interpreted by others and applied in practice. This kind of experience will produce the input we need to move toward process-oriented techniques for using Design Space Analysis in design practice.

Acknowledgments. Thanks to Judy Olson and Gary Olson for access to the data from the ATM2 study carried out at the University of Michigan with the assistance of Robin Lampert and Mark Carter. Thanks also to the designers who took part in the studies. We are grateful to the following people for many helpful comments on earlier drafts of this article: Liam Bannon, Tom Carey, Jack Carroll, Steve Draper, Jonathan Grudin, Mik Lamming, Jintae Lee, Judy Olson, Simon Shum, and two anonymous reviewers. We are also grateful to our colleagues in the AMODEUS project for many helpful discussions.

Support. The work reported here was partly funded by the European Commission as part of the AMODEUS project, Esprit Basic Research Action 3066.

REFERENCES

Adams, J. L. (1974). *Conceptual blockbusting.* San Francisco: W. H. Freeman.

Alexander, C. (1964). *Notes on the synthesis of form.* Cambridge, MA: Harvard University Press.

Allison, G. T. (1971). *Essence of decision: Explaining the Cuban missile crisis.* Boston: Little, Brown.

Apple user interface guidelines. (1987). Reading, MA: Addison-Wesley.

Balzer, R., Cheatham, T. E., & Green, C. (1983). Software technology in the 1990's: Using a new paradigm. *IEEE Computer, 16*(11), 39-45.

Botterill, J. H. (1982). The design rationale of the System/38 user interface. *IBM Systems Journal, 21*(4), 384-423.

Card, S. K., English, W. K., & Burr, B. J. (1978). Evaluation of mouse, rate-controlled isometric joystick, step keys and text keys for text selection on a CRT. *Ergonomics, 21,* 601-613.

Carroll, J. M., & Rosson, M. B. (1990). Human-computer interaction scenarios as a design representation. *Proceedings of HICSS-23: 23rd Hawaii International Conference on System Science,* 555-561. Los Alamitos, CA: IEEE Computer Society Press.

Carroll, J. M., & Rosson, M. B. (1991). Deliberated evolution: Stalking the View Matcher in design space. *Human-Computer Interaction, 6,* 281-318. [Included in this Special Issue.]

Conklin, J. (1987). Hypertext: An introduction and survey. *IEEE Computer, 20*(9), 17-41.

Conklin, J. (1989). Design rationale and maintainability. *Proceedings of the 22nd International Conference on System Sciences,* 533-539. Los Alamitos, CA: IEEE Computer Society Press.

Conklin, J., & Begeman, M. L. (1989). gIBIS: A tool for all reasons. *Journal of the American Society for Information Science, 40,* 200-213.

Conklin, E. J., & Yakemovic, KC B. (1991). A process-oriented approach to design rationale. *Human-Computer Interaction, 6,* 357-391. [Included in this Special Issue.]

Fischer, G., Lemke, A. C., McCall, R., & Morch, A. I. (1991). Making argumentation serve design. *Human-Computer Interaction, 6,* 393-419. [Included in this Special Issue.]

Fitts, P. M. (1954). The information capacity of the human motor system in controlling amplitude of movement. *Journal of Experimental Psychology, 47,* 381-391.

Goel, V., & Pirolli, P. (1989, Spring). Design within information processing theory: The design problem space. *AI Magazine,* pp. 18-36.

Grudin, J. (1988). Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. *Proceedings of the CSCW '88 Conference on Computer-Supported Cooperative Work,* 85-93. New York: ACM.

Guindon, R., Krasner, H., & Curtis, B. (1987). Breakdowns and processes during the early acquisition of software design by professionals. In G. M. Olson, S. Sheppard, & E. Soloway (Eds.), *Proceedings of the Second Workshop on Empirical Studies of Programmers* (pp. 65-82). Norwood, NJ: Ablex.

Halasz, F., Moran, T., & Trigg, R. (1987). NoteCards in a nutshell. *Proceedings of CHI + GI '87: Human Factors in Computing Systems,* 45-52. New York: ACM.

Hammond, N., Jørgensen, A. H., MacLean, A., Barnard, P., & Long, J. (1983). Design practice and interface usability: Evidence from interviews with designers. *Proceedings of the CHI '83 Conference on Human Factors in Computing Systems,* 40-44. New York: ACM.

Hayes, J. R. (1981). *The complete problem solver.* Philadelphia: Franklin Institute Press.

Johnson, J., & Beach, R. J. (1988). Styles in document editing systems. *IEEE Computer, 21*(1), 32-43.

Kunz, W., & Rittel, H. (1970). *Issues as elements of information systems* (Tech. Rep. No. S-78-2). Stuttgart: University of Stuttgart.

Lee, J. (1990). SIBYL: A qualitative decision management system. In P. H. Winston & S. Shellard (Eds.), *Artificial intelligence at MIT: Expanding frontiers* (Vol. 1, pp. 104-133). Cambridge, MA: MIT Press.

Lee, J., & Lai, K.-Y. (1991). What's in design rationale? *Human-Computer Interaction, 6,* 251-280. [Included in this Special Issue.]

Lewis, C., Polson, P., Rieman, J., & Wharton, C. (1990). Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. *Proceedings of the CHI '90 Conference on Human Factors in Computing Systems,* 235-242. New York: ACM.

Lewis, C., Rieman, J., & Bell, B. (1991). Problem-centered design for expressiveness and facility in a graphical programming system. *Human-Computer Interaction, 6,* 319-355. [Included in this Special Issue.]

Mackay, W. (1991). Triggers and barriers to customization. *Proceedings of the CHI '91 Conference on Human Factors in Computing Systems,* 153-160. New York: ACM.

MacLean, A., Bellotti, V., & Young, R. M. (1990). What rationale is there in design? *Proceedings of the INTERACT '90 Conference on Human-Computer Interaction,* 207-212. Amsterdam: North-Holland.

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

MacLean, A., Carter, K., Lövstrand, L., & Moran, T. P. (1990). User-tailorable systems: Pressing the issues with Buttons. *Proceedings of the CHI '90 Conference on Human Factors in Computing Systems,* 175-182. New York: ACM.

MacLean, A., Young, R. M., & Moran, T. P. (1989). Design rationale: The argument behind the artifact. *Proceedings of the CHI '89 Conference on Human Factors in Computing Systems,* 247-252. New York: ACM.

Marshall, C. C., & Irish, P. M. (1989). Guided tours and on-line presentations: How authors can make existing hypertext intelligible for readers. *Proceedings of Hypertext '89,* 15-26. New York: ACM.

Martin, J. (1977). What to plan for to manage the future of your data center. *Canadian Datasystems, 9*(3), 28-32.

McCall, R. (1986). Issue-serve systems: A descriptive theory of design. *Design Methods and Theories, 20*(3), 443-458.

Moran, T. P. (1981). The command language grammar: A representation for the user interface of interactive computer systems. *International Journal of Man-Machine Studies, 15,* 3-50.

Newman, W. M. (1988). The representation of user interface style. In D. M. Jones & R. Winder (Eds.), *People and computers IV: Designing for usability* (pp. 123-144). Cambridge, England: Cambridge University Press.

Olson, G., & Olson, J. (1991). User centered design of collaboration technology. *Journal of Organizational Computing, 1,* 61-83.

Parnas, D., & Clements, P. C. (1986). A rational design process: How and why to fake it. *IEEE Transactions on Software Engineering, SE-12,* 251-257.

Payne, S. J., & Green, T. R. G. (1986). Task action grammar: A model of the mental representation of task languages. *Human-Computer Interaction, 2,* 93-133.

Polya, G. (1957). *How to solve it* (2nd ed.). Garden City, NY: Doubleday.

Schön, D. A. (1983). *The reflective practitioner: How professionals think in action*. New York: Basic Books.

Schön, D. A. (1987). *Educating the reflective practitioner*. San Francisco: Jossey-Bass.

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

Simon, H. (1981). *The sciences of the artificial* (2nd ed.). Cambridge, MA: MIT Press.

Smith, D. C., Irby, C., Kimball, R., & Verplank, W. (1982, April). Designing the Star user interface. *Byte*, pp. 242-282.

VanLehn, K. A. (1985). *Theory reform caused by an argumentation tool* (Tech. Rep. No. ISL-11). Palo Alto, CA: Xerox Palo Alto Research Center.

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

HCI Editorial Record. First manuscript received July 13, 1990. Revisions received February 21, 1991, and April 25, 1991. Accepted by John M. Carroll.—*Editor*

APPENDIX. CREATING A DESIGN SPACE ANALYSIS

The article discussed several aspects of Design Space Analysis. We introduced the basic elements of the analysis and presented the QOC notation for representing it, and we presented several ways of extending the analysis to cover a variety of modes of justification. We saw that this kind of analysis is not unlike the naturally occurring discussion that takes place among designers, but we also observed that naturally occurring discussion seems to fall short of a logically coherent rationale when compared to a Design Space Analysis. Although there are times during design when trying to conform to a logical structure would be inhibiting, there are also many times when it would be helpful to improve the quality of design reasoning. We believe that the explicit use of Design Space Analysis can help give structure and discipline to design reasoning. There are several kinds of tasks in design where such discipline could be helpful: preparing a presentation or review of a design project, providing a map to keep track of the territories explored in a project, helping generate and evaluate new ideas during design meetings, and so on.

There is no strict methodology for creating a design space, such as a top-down sequence. To try to follow a strict procedure would not work. An analysis is developed in all places at once by a mixture of inspiration and reflection, as ideas pop up, get understood, and fit into place. On the other hand, the process of developing a rationale is not random. There are systematic steps that can aid the process enormously. These steps (derived from our own analysis work and our observations of designers at work) involve focusing on parts of the QOC representation in principled ways to see how to augment it. We formulate the steps as a set of heuristics for would-be analysts to guide them in building QOC representations.

We call this advice *heuristic* because creating the design space, as well as creating the artifact, is a discovery process; in fact, it is one and the same discovery process. The heuristics reflect general considerations about problem solving (e.g., see Hayes, 1981; Polya, 1957) and creative processes (e.g., see Adams, 1974) applied to the specific task of creating a design space in QOC notation. We start by presenting some "local" heuristics, which are aimed at helping us locally expand the notation. We then present some more "global" heuristics, which are aimed at dealing with larger patterns in the notation.

A1. Local Heuristics for Design Space Analysis

The purpose of local heuristics is to give advice for how to reason in the area of a single Question to enhance understanding of the design or to try to find a better solution. They provide for expansion of the QOC notation in a link-by-link and node-by-node manner.

The generation of possible design Options can be aided by having a clear understanding of important issues. The right Question highlights relevant issues and encourages the generation of appropriate Options, so:

Heuristic 1: Use Questions to generate Options.

However, it is usually difficult to formulate directly incisive Questions. We have noted that possible Options seem to spring to mind, apparently in isolation. Asking oneself "to what Question is the Option an answer" and reflecting on its important or novel features can lead to good Questions:

Heuristic 2: Use Options to generate Questions.

Going back and forth between Questions and Options (using Heuristics 1 and 2) is consistent with Schön's (1987) discussion of generating possible solutions, reflecting on their characteristics, and generating better solutions.

Insights into the structure of the design space can be gained if we can identify appropriate Options, by which we mean Options that bring out distinctive features in the set of possibilities:

Heuristic 3: Consider distinctive Options.

In some situations, considering extreme solutions is a way to find distinctive Options. Such Options can be a useful way to "shake up" our view of the design space, both to aid understanding it (e.g., by making clear the tradeoffs between the Criteria) and to help generate new solutions within it.

The heuristics so far have emphasized the exploration of alternative Options. The exploration of a range of Criteria is also necessary to provide a balanced view of the pros and cons of proposed Options:

Heuristic 4: Represent both positive and negative Criteria.

As a minimum for exploring the design space, we recommend that each Option have at least one Criterion against which it is assessed positively and one against which it is assessed negatively. This allows us to understand the tradeoffs. It is important to recognize such tradeoffs when they exist, but there is clearly more to design than simply evaluating tradeoffs, so we could express this heuristic more generally as, "Look for the downside as well as the upside."

Tradeoffs do not have to lead to compromise. Design should be creative—Avoid compromises by trying to find Options that by-pass them. Sometimes the alternative Options can be combined into a single Option with the advantages of each. The heuristic for doing this is:

Heuristic 5: Overcome negative, but maintain positive, Criteria.

A2. Global Heuristics for Design Space Analysis

The purpose of the global heuristics is to help us look beyond the local region of the representation. These heuristics are aimed at dealing with broader design issues: modularizing the design, looking for emergent design possibilities, and addressing the coherence of the design as a whole.

There are usually too many degrees of freedom in moving toward a design solution, and the decisions are highly interrelated. Extensive problem structuring is required to determine a fruitful way of framing the problem (e.g., Schön, 1983). The most common strategy for tackling such problems is to subdivide the problem into smaller more manageable components or modules (e.g., Alexander, 1964; Simon, 1981). In software projects, this decomposition is most obvious when different teams are given different parts of the project to work on, but the same basic strategy applies even when individuals tackle design problems. In practice, complete isolation between modules is impossible to achieve, so we try to define modules that maximize the interaction within them and minimize the interaction between them. In Design Space Analysis, cross-Question constraints (imposed by dependencies between Options) are perhaps the most critical feature in defining modules:

Heuristic 6: Identify Options that generate dependencies.

One of the main advantages of a Design Space Analysis is that several possible designs are captured within the same representation. This can provide insights into combinations of Options that have not been previously considered together but that might lead to an improved design:

Heuristic 7: Look for novel combinations of Options.

Section 4 argued that Criteria play an important role in shaping and maintaining the overall coherence of a design. It is therefore important to:

Heuristic 8: Design to a set of Criteria.

A strategy we have found useful for creating a Design Space Analysis is to identify a list of General Criteria that are most important for the design and from these to formulate a set of more specific Bridging Criteria to use in evaluating local Options. We have observed that systematic application of Criteria is rare in design practice, and attention to this heuristic would help designers clarify their motives and objectives.

A final issue is to note that design is as much about inventing themes as making specific design decisions. Locally optimized decisions do not add up to good overall design. Many core decisions must be made to establish a consistent overall policy of the design. In Design Space Analysis terms, it is vital to:

Heuristic 9: Search for generic Questions.

Generic Questions serve to select aspects of the design to frame their analysis in terms of the overall impact on the design rather than on local considerations. The responses to such Questions will determine the overall coherence and consistency of the design. For example, several commercial desktop systems are built on a set of style rules or specifications that define their "look and feel," the most prominent being the original design of the Xerox Star (Smith, Irby, Kimball, & Verplank, 1982) and the Apple Macintosh (Apple User Interface Guidelines, 1987).