Chapter to appear in *"Design Rationale: Concepts, Techniques and Use"*
Moran T.P. and Carroll J.M. (Eds.), 1995, in press. Lawrence Erlbaum Associates: NJ

Analysing the Usability of a Design Rationale Notation

Simon Buckingham Shum

Human-Computer Interaction Group
Department of Psychology
University of York
York YO1 5DD
UK
sjbs1@york.ac.uk
tel: +44 1904-433165
fax: +44 1904-433181

ABSTRACT

Semiformal, argumentation-based notations are one of the main classes of formalism currently being used to represent design rationale (DR). However, our understanding of the demands on designers of using such representations has to date been drawn largely from informal and anecdotal evidence. One way to tackle the fundamental challenge of reducing DR's representational overheads, is to understand the relationship between designing, and the idea structuring tasks introduced by a semiformal DR notation. Empirically based analyses of DR in use can therefore inform the design of the notations in order to turn the structuring effort to the designers' advantage. This is the approach taken in this chapter, which examines how designers use a DR notation during design problem solving.

Two empirical studies of DR-use are reported, in which designers used the QOC notation (MacLean et al., this volume) to express rationale for their designs. In the first study, a substantial and consistent body of evidence was gathered, describing the demands of the core representational tasks in using QOC, and the variety of strategies which designers adopt in externalising ideas. The second study suggests that an argumentation-based design model based around laying out discrete, competing Options is inappropriate during a depth-first, 'evolutionary' mode of working, centered around developing a single, complex Option. In addition, the data provide motivation for several extensions to the basic QOC notation. The chapter concludes by comparing the account of the QOC–design relationship which emerges from these studies, with reports of other DR approaches in use.

1. THE NEED FOR EMPIRICAL STUDIES OF DR IN USE

2. THE STUDIES: DESIGNERS, TRAINING, AND TASKS

3. CORE REPRESENTATIONAL TASKS IN QOC-AUTHORING

   3.1. QOC authoring as an opportunistic activity

   3.2. Classifying ideas

   3.3. Naming and renaming

   3.4. Structuring and restructuring

   3.5. Summary

4. PROBLEMS USING QOC IN 'EVOLUTIONARY' DESIGN: TRYING TO ARGUE ABOUT ONE OPTION?

   4.1. Difficulties encountered with QOC constructs
   Questions
   Options
   Criteria

   4.2. Characterising the relationship between QOC and the two modes of designing

5. QOC'S EXPRESSIVENESS

   5.1. Representing evolution within QOC structures

   5.2. Expressing constraints and dependencies

   5.3. The subtleties of expressing Options, Criteria, and Assessments

6. RELATIONSHIP TO REPORTS OF OTHER DR APPROACHES IN USE

7. CONCLUSIONS

8. REFERENCES

1. THE NEED FOR EMPIRICAL STUDIES OF DESIGN RATIONALE IN USE

Semiformal, argumentation-based notations are one of the main classes of formalism currently being used to represent design rationale (DR). Whilst from a notational perspective, graphical formalisms are well suited for recording design arguments as they arise, doing so also introduces representational overheads for the designer. This chapter is concerned with understanding the nature of the extra cognitive work introduced by argumentation-based DR. This is obviously important in the context of a fast-flowing, time-pressured activity such as software design, in which 'documentation' is already a bad word.

Usable, effective DR tools can only be developed once we have an understanding of the cognitive, group, and organizational factors implicated in the introduction of explicit DR to the design process. The present work focusses on the cognitive factors which determine the usability of argumentation-based DR notations, taking as an example the *QOC* notation and *Design Space Analysis* perspective (MacLean, Young, Bellotti, & Moran, this volume, Section 2). Whilst MacLean et al. focus on the properties of QOC as a representation for DR, and its relationship to other approaches, attention in this chapter shifts to the *process* of authoring QOC, and its relationship to different modes of software design activity. The analyses of the data gathered in these studies address issues relating to QOC's usability and scope, with broader implications for other DR approaches and their associated representational schemes.

2. THE STUDIES: DESIGNERS, TRAINING, AND TASKS

Analyses of two empirical studies are presented in this chapter. All of the data reported are drawn from video-based observational analyses of design problem solving. In Study A, 12 pairs of software designers (16 professionals/8 students) spent an hour using QOC to redesign and rationalize the user interface to a bank's automated teller machine (ATM). The task was based on that employed by MacLean et al. (this volume, Figure 4).[1] In Study B, in which different modes of design are considered, an electronics research student engaged in doctoral research described his work (designing Smalltalk-80 data structures) and use of QOC over three 1 ¼ hour sessions.

In Studies A and B, the designers underwent a training procedure which introduced DR as a general concept, and QOC specifically. Emphasis was placed on the importance of developing coherent rationales which would communicate clearly to an outsider the key reasons behind the designs, reflecting the more retrospective Design Space Analysis approach. The QOC tutorial-tasks were intended to give the designers practice in structuring natural discourse semiformally. Designers were required to translate into QOC the key aspects of several fictional design discussions, such that a third party could understand what had been discussed. By varying the length of these design discussions, and the medium in which they were presented (as transcripts plus sketches, or as a video-recording of a discussion) the representational task became steadily more demanding. Details can be found in Shum (1991).

In all of the studies, designers used pens and large sheets of paper as opposed to a software tool. Under these conditions, the authoring process could be studied with minimal interference from extraneous factors, whilst preserving or even enhancing properties of the online medium such as display space, resolution, and ease of local editing. Whilst computational tools can alleviate some of the mechanical overheads of the task, the core tasks of deciding how to express reasoning as structured argumentation remain essentially unchanged.

Throughout this paper, extracts from the design transcripts are used to illustrate points. In longer transcript extracts, the key points are shown in bold, and ideas recorded as QOC are shown **like**

---

[1] Two tasks were in fact used, in a between-subjects experimental design. The first described user-steps for a Standard-ATM (SATM) interface, and requested a new design and DR. The second additionally described the Fast-ATM interface, and several SATM usability problems. Designers were required to evaluate the FATM, and if necessary, devise and rationalize changes. The data from both tasks are combined for the analysis presented here.

this. Most of the examples in this section are from Study A's ATM design problem, which centered around reducing customer queues without sacrificing the number of services offered. Other extracts are from Study A tutorial exercises: one task was to design the remote control for a video-recorder intended for the elderly, and the other was to design an airport public information symbol to indicate a "one hour left-luggage office."

3. CORE REPRESENTATIONAL TASKS IN QOC-AUTHORING

In order to translate ideas into QOC, the designer is faced with three basic cognitive tasks: deciding what *kind* of an idea one has (classification), how to *label* it meaningfully (naming), and how it *relates* to other ideas (structuring). Before these tasks are illustrated, however, it is necessary to emphasize their non-linear relationship, that is, the exploratory, opportunistic nature of the process.

3.1. QOC authoring as an opportunistic activity

When studying designers using QOC, it soon becomes clear that externalising ideas as structured argumentation is not a smooth, top-down process. Continual revision and switching from one task to another characterize QOC authoring as an opportunistic mode of working (Guindon, 1990). The QOC evolves through multiple, sometimes embedded, represent-and-evaluate cycles, switching between different parts of the structure.

Various approaches to representing QOC were adopted, demonstrating that the process of developing QOC analyses is quite different from the orderly structure of the final product. For instance, in the following extract in which the designers discuss how the ATM should dispense different kinds of output, it was most natural to generate Options, then Criteria, and the Question last of all.

[Study A: Pair 2]

P: Halifax machines drop everything into a little drawer... the Question here is... well the ideas [i.e. Options] are **as is**, and **everything from one place**. The Criteria are...

D: what are you going to call the Question though?

P: Hmm, I get caught on the Questions....the Criteria are **natural feel** to it - getting it from different holes doesn't feel natural

D: actually, it's more like a teller, more human

P: what? If you get it all from the same hole?

D: the same kind of thing like when you go the counter, and the guy gives you it through the little slot

P: [writes] **security in mind (everything from a draw – feels secure)**. [Linking Options to Criteria] - **as is** – it doesn't have a natural feel to it, everything from different slots.

What's the Question here? (frustrated tone).

D: erm... I suppose physical layout of...

P: layout of holes [starts to write]

D: physical layout of input/output stuff [P. writes **layout of I/O for cash/card/receipt**]

Figure 1 shows the order in which QOC was constructed in another situation, illustrating switching between Questions to capture new ideas as they suggest themselves.[2]

---

2 Apart from 'cleaned-up' graphic appearance, all QOC examples used in this chapter are copied directly from designers' QOC representations, unless otherwise stated.

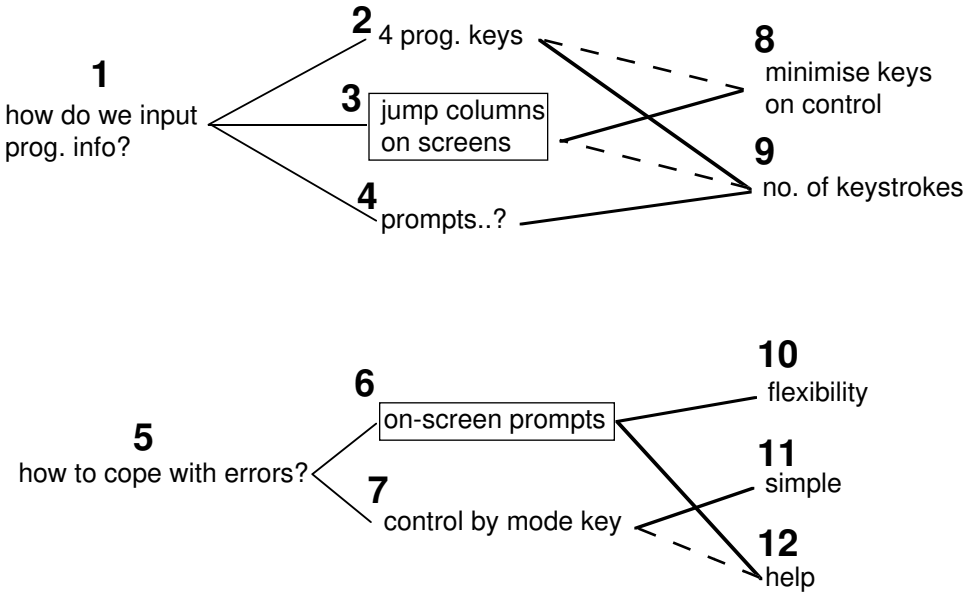

Figure 1: [Study A tutorial exercise: Pair 2] Moving opportunistically to a new Question and Options as they arise, and then back to complete the original (numbers added to show sequence of ideas). In QOC, boxed Options indicate a decision, or at least a working commitment.

In some cases, subjects explicitly adopted 'strategies' to representing the QOC, as ways of imposing some structure on their task. For example, in order to control the tendency to pursue new ideas as they arose opportunistically, one pair declared:

[Pair 12: Study A]

G: let's try to write down some of the Questions of concern here, and then some Criteria, and then I think some of the Options will come from that - possibly... 'cos Questions and Criteria are related a lot aren't they

J: so if we have one heading Questions and another here for Criteria, which might not be related

G: right - I'm going to concentrate in terms of Questions, and then the Criteria might come from the Questions. **length of queue** is a Question...

About a minute later, they started to discuss the details of a possible Option but stopped themselves intentionally, aware that at this early stage it might be a tangent:

A: so could you have a machine that behaves like a regular ATM or a fast ATM depending on where you put the card in?

A: you could... but would that be...? – right, so

G: yeah - so (writes) **variety of machines** ... (no further discussion on that Option until later)

It is important to note, however, that strategies of this sort seemed to be short-term, flexible modes of working, that is, they did not govern the structure of the whole session, and within them there was flexibility to attend to different parts of the QOC and to switch strategies, such as:

- listing Questions in advance, before elaborating them;
- generating Options and Criteria, and then the Question;
- generating Questions and Options first, and then evaluating with Criteria.

QOC authoring is not only opportunistic when used *during* conceptual problem solving of the sort required by the ATM task. Even when the decisions have been made and all the arguments are known (i.e. *retrospective* DR), working out how best to represent them is a separate task. Having recognized that the authoring process is far from a tidily sequenced activity, let us now turn to its constituent tasks.

3.2. Classifying ideas

This section focusses on the normal process of classifying ideas. When QOC is being used, the vocabulary and orientation of discussion inevitably changes, with regular references to the new constructs, e.g. *that's an Option; could that be a Question?; this Criterion keeps coming up,* and so forth. As with any language, fluency increases with use, such that arguments are more smoothly and accurately translated over time. In Study A, it was found that normally subjects classified ideas without spending much time discussing what type they should be. However, as the examples of restructuring QOC show [Section 3.4], the first translation which springs to mind is not necessarily the optimal representation. Even relative 'experts' with QOC (e.g. QOC's developers) still engage in restructuring, reclassification, and renaming, revision activities which are dealt with below. The following examples illustrate classification difficulties characteristic of early use of QOC.

(i) Figure 2 shows a typical error in classifying an idea (**natural order** is initially recorded as an Option, but then corrected to be a Criterion):

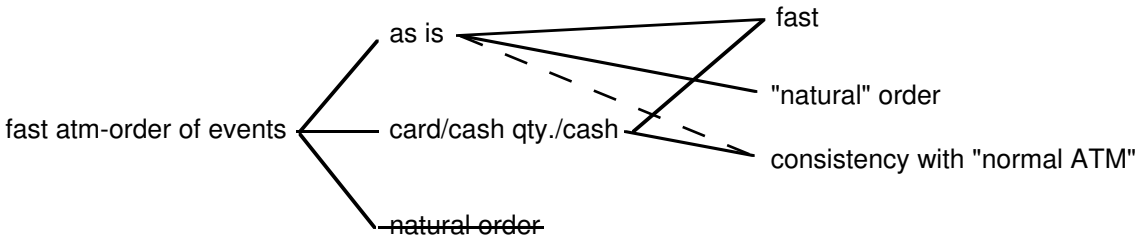

Figure 2: [Study A: Pair 2] A typical error in classifying an idea (a Criterion as an Option).

(ii) In the following extract, an Option is first represented as a Question:

[Study A: Pair 2]

D: what does this do then? – the Fast ATM do, if you press the cash amount and not (put in) the card?

P: I guess it just goes 'Ha Ha,' and clears itself. It'll have a **time out**, or **clear key**.

D: yeah, but that's an Option though – a Clear key.

P: on where?

D: well on the normal ATM. You could change the order of the events, but you have a clear key and a timeout

P: fast ATM order of events [writes this as a new Question]

... that's an Option?

D: yeah that's an Option

P: oh, it's an Option which addresses this security issue isn't it [deletes Question: clear key and timeout on ATM, and writes clear key + timeout as an Option]. What's the Question? It's sort of a vandalism issue...

(iii) This extract shows the pair generating ideas in discussion, and then striving to represent those ideas as QOC:

[Study A: Pair 5]

R: that's a design decision [pointing to the keypad sketch]. Deciding that Cancel is going to be the emergency get-out, and that we should stick with that as its mode - the mode of that key is 'get out of this, now'.

J: well it certainly is... it goes back to this question of **buttons** - is that what the Cancel... I mean we've added the Cancel key, and consider that to be a design conclusion – I like that because it's intuitive. I suppose that could be a Criterion – in fact it's a very important user-Criterion. [sigh] How do we record that?

(...)

J: We've not got much time left. We could do with identifying this Cancel key more specifically. But I don't know what the Question is.

R: in fact the Question is 'How does the card get returned?'

J: you can put it under the question of **buttons**...

...I think the Enter key and Cancel key are part of the question of **buttons**. That's the Question [indicates **Q. of buttons**] – now there are probably other Questions which feed into this, and I don't know that you can actually... Hah! [laughs] When is something a Question, and when is it a Criterion?

In the above extract, the decisions and arguments are clear in J.'s mind, but neither he nor his partner are sufficiently fluent with QOC to translate them. J. notes in particular the informality inherent in QOC (e.g. how to classify ideas).

3.3. Naming and renaming

Naming entities is often a process of renaming. The renaming of nodes was a prevalent activity for every pair of designers in Study A. Renaming reflects the problem solving process of developing ideas; if a QOC is constructed as the problem is explored, it is inevitable that node-names which do not reflect current understanding of the problem must be updated.

Naming in QOC takes up a significant amount of time for several reasons. Firstly, a node's name must be succinct, and convey the idea it represents. Secondly, to aid interpretation, a further constraint on Criteria is that they be expressed positively, e.g. **easy to learn**, **low error rate**, **low cost, high speed**.[3] Thirdly, a particularly important characteristic of names is *focus*. Focus refers to the level of generality at which the idea is expressed: a Question may address several issues; an Option may embody several key features which differentiate it from others, but not along the dimension which is addressed by the Question; a Criterion might be expressed so generally (e.g. **intuitive**; **simple**) that it is hard to see how it relates to an Option. A fourth property of a name is its relationship to others of its type: it should be *distinctive*. An Option may really be an example of another, or two Criteria might really be re-expressions of each other (e.g. each side of a trade-off—depending on the context, this might be useful or redundant). Both distinctiveness and focus in naming are characteristics of 'well-formed' QOC.[4]

Although these requirements were not made explicit to designers in Study A, they still appreciated the importance of finding good names for ideas. The extracts below illustrate the cooperative process of refining names.

[Study A tutorial exercise: Pair 5]

R: so how are we going to...

T: "keys to what kind of functions..."

R: that's not a very good way of putting it...

T: it's like the "classes of functions..."

R: classes! That's the way to put it.

T: **What classes of function keys**?

R: ...you see teletext is the only thing you read – you don't read other things – you don't read the picture.

T: what we have are two negative reasons

R: we have to make them positive though...ok, so, "easy to read?"

T: well, that wasn't the point was it? um... it was like that they couldn't actually...

R: they can't see it, so they don't need it.

T: [laughs] yeah -- it's like the Criterion is that you're providing a function which they can actually make use of, and they can't make use of the teletext because it's too small to read.

R: ok – **useful function**?

T: yeah, ok.

---

3 With this constraint, **supports** Assessment links to Options can always be interpreted as 'pros', and **objects-to** links as 'cons'; because Criteria have different weights [Section 5.3], decisions clearly cannot be made on the basis of how many **supports** links Options have, but they provide an initial visual indication.

4 Principles for well-formed structures were collated as a 'QOC styleguide.'

The following comment summarizes the experience of many subjects in having to name Criteria positively:

[Study A: Pair 5]

R: ... I mean, I really struggled on that first exercise, and found that very awkward and very difficult. In fact the thing I found most difficult was negating everything, so that the attribute was a positive attribute

J: yes

R: I just couldn't get my brain to pick out the right word to describe that attribute.

3.4. Structuring and restructuring

The primary organization which presents itself to someone browsing a QOC diagram is the Question structure, and it is under Questions which design ideas must be eventually placed. For this reason, the problems which designers choose to address through the Questions are important: the Questions addressed define the space which the team sees their design occupying, and guides the direction of future deliberation. Many examples of Question structuring and restructuring were collated in this and other studies, three of which are reproduced below. Note that renaming of Questions is covered here (rather than in the previous section) because of their importance in shaping the macro-structure of the QOC.

(i) Working out the Question together:

[Study A: Pair 12]

G: (new sheet) Our first Question - do you want different kinds of ATMs for the different – you know, a fast ATM and a fully functional one – or do you want to do it all at once?

J: ok, so what's the Question - 'cos those are the Options aren't they?

G: do you want..um.. well the Question is...em...

J: single machine

G: yeah, kinds of... do you want to have just one machine or do you want to have...

J: well those are Options

G: yeah, well I know those are Options [laughs], but the Question can kind of beg the question

J: well it could be a Question – "do you want a variety of machines?" Yes or No

G: well [Question] **number of ATM designs**: one and more than one - typically two: [Options] **fast** and **fully functional**

(ii) The subjects return to their first Question, and realize that it no longer expresses what they have now identified as the real problem (the design of the first screen):

[Study A: Pair 10]

T: oh no. [pause - returns to Question] Except that this isn't really **how to develop user interface** - it's [really to do with] the first screen isn't it?

A: what do we show initially?

T: yeah

A: [changes first Question] **what do we display on 1st screen**?

(iii) Similar to the last example, Figure 3 shows how a Question is refocussed to express the problem which the generated Options now seem to be addressing (**user interface education**).

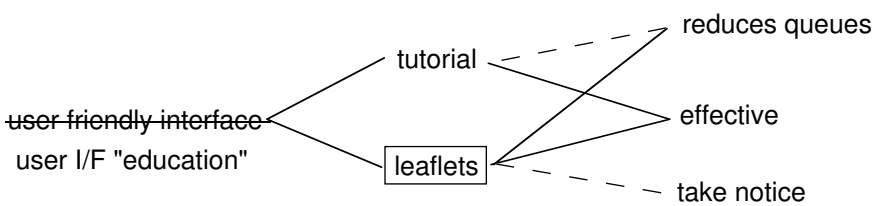

Figure 3: [Study A: Pair 2] Refocussing a general Question to capture the issue actually addressed by the Options.

As Bellotti, MacLean and Moran (1991) emphasize, asking the right Questions is critical to developing a useful design space representation, and avoiding particular mental sets or design fixations (Jansson & Smith, 1991). The data collected in these studies in fact demonstrate that Question revision is the natural process which designers follow when using QOC, even though the Study A designers (from whom all of the above examples are taken) were not explicitly told to work on refining Questions. Design Space Analysis, with its particular emphasis on asking good Questions, attempts to build on and support this activity.

The above examples showed that the reformulation of Questions often takes place in response to the Options generated. The relationship is reciprocal, however, since an insight into the nature of a Question can lead to restructuring of those Options, by moving them to new or existing Questions; another example would be making an implicit criterion embedded in a Question explicit as a Criterion. Whilst this distinction is useful for analytic purposes, the two are often tightly interwoven during authoring. Several examples of restructuring are presented below.

(i) In Figure 4, Options to a Question are moved when it is realized that the Question breaks down into two separate Questions (the Options are separated to indicate that **stopwatch** and subsequent Options now respond to a second Question).

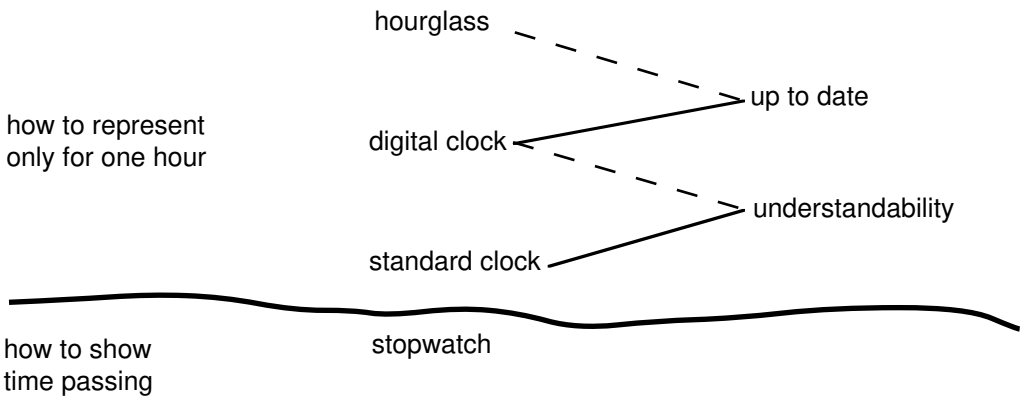

Figure 4: [Study A tutorial exercise: Pair 8] Dividing a Question into two as it is realized that the Options serve two different roles.

(ii) In Figure 5, the designers make the Criterion **increased queueing** explicit, rather than leaving it embedded in the Question. It can then be used to differentiate between the two Options.

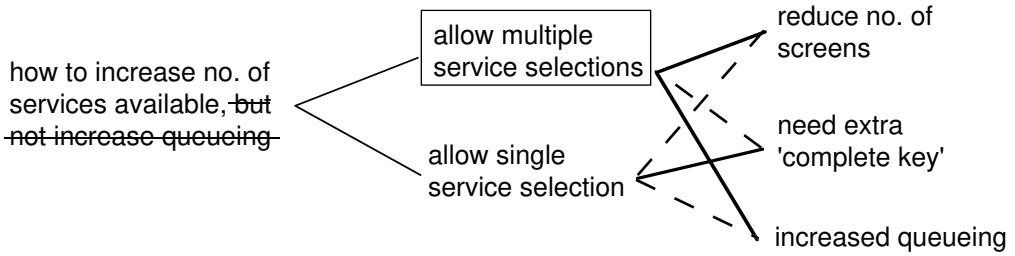

Figure 5: [Study A: Pair 8] Extracting an important, but implicit requirement in a Question, and making it explicit as a Criterion to evaluate the Options.

(iii) In another incident [Study A: Pair 10], two designers initially recorded several ideas as Options (**reduce response time, minimize key depressions, minimize no. screens**) in response to a high level Question, **How to reduce queues?** They then realized that they really wanted to choose *all* of them, which was a clue that they could serve as Criteria. The Question was restructured accordingly, and the Criteria reused in subsequent Questions. This pattern has been observed on other occasions, and reflects the process of defining goals (or requirements) as the first step to formulating and evaluating solutions. Recognizing regularities such as these is valuable to user communities as they seek to build and share knowledge and expertise in a particular formalism.

(iv) In Study B, a design session involved the gradual identification of hierarchical relationships between Options. The designer redrew his QOC structure in order to make this explicit and went on to develop the hierarchy further, as shown schematically in Figure 6.

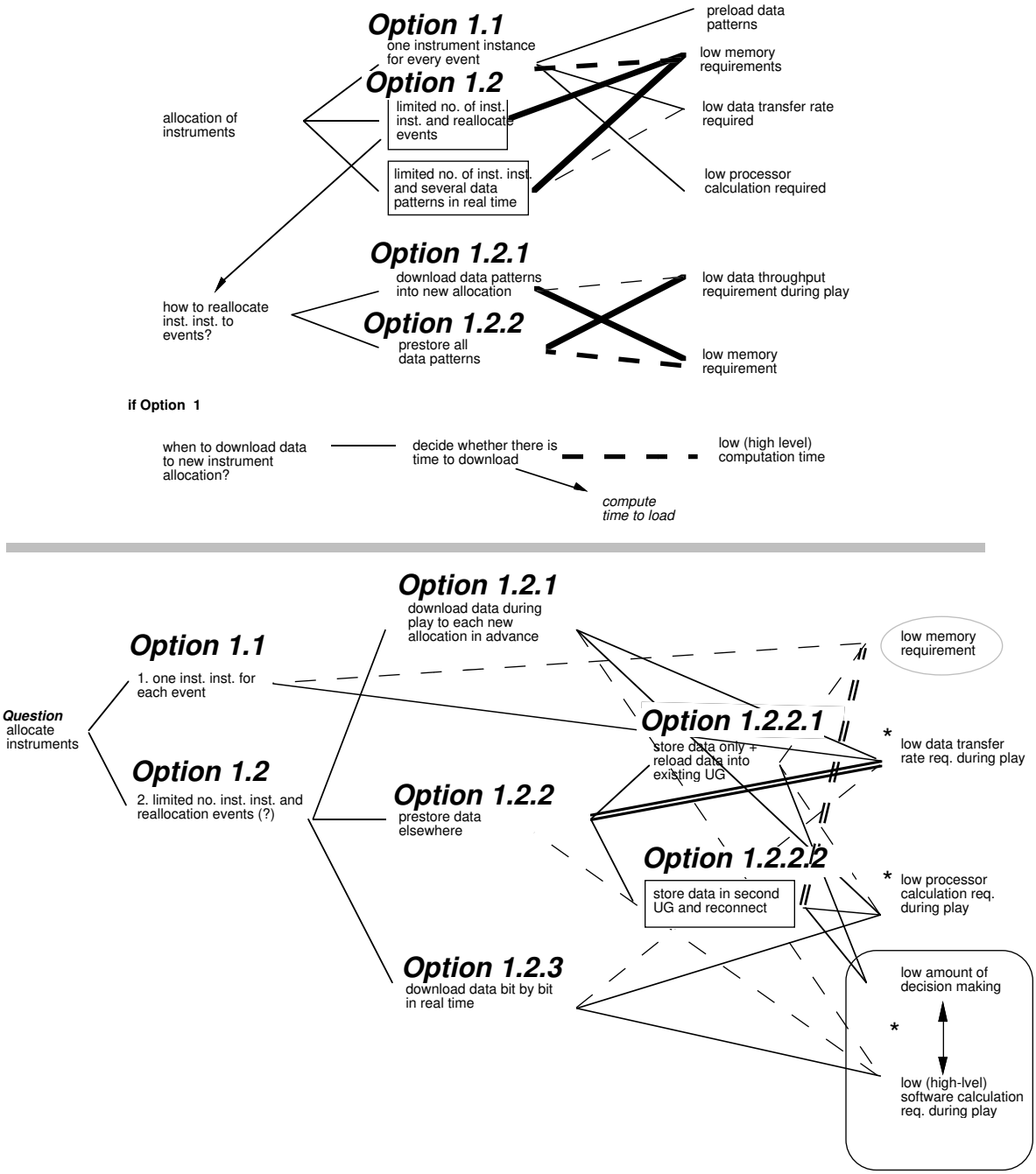

Figure 6: [Study B] Structural overview of QOC structures to show how they were restructured in order to work on the Option hierarchy (Option numbers added to show the transition). Note that the process of making the Option-structure explicit prompted the designer to change Option 1.3 to Option 1.2.3, and to decompose Option 1.2.2 one level further. (As the designer was using pen and paper and renamed and restructured extensively, the original QOC was littered with changes which have been omitted for clarity). Reproduced from Buckingham Shum and Hammond (1994), with permission, Academic Press, Ltd.

3.5. Study A: Conclusions

Open-ended, ill-structured, 'wicked' problems (Rittel & Webber, 1973) are rendered manageable only through the exploratory process of framing and reframing views in order to better understand constraints on the solution space. For the designers studied, designing the ATM user interface engendered such a mode of working, which led to extensive revision of QOC names and structure as ideas developed.

Whilst the amount of effort devoted to classifying, naming, and structuring was not documented quantitatively, it is likely that relative amounts would depend at least in part on the users, task and familiarity of domain. Ongoing experiences in using QOC does, however, suggest that these tasks persist as features of 'expert' QOC-use, although experts are able to draw on strategies for advancing the QOC in situations which might 'stall' a less experienced user (cf. Section 3.4, example (iii), and MacLean et al.'s heuristics (this volume, Appendix)). One would not in fact expect such features of the task to disappear since a claim made explicitly by proponents of semiformal notations is that the discipline of expressing ideas within a constrained vocabulary encourages a dialogue with the representation, which can 'talk back' to the designer and expose weaknesses in thinking.

The above analysis of authoring behaviour was based on data from a task intentionally selected to allow QOC to be studied—the problem domain was novel, with many issues left open, and the competing ATM designs (presented to half the designers) focussed attention on tradeoffs between Options. In contrast, the evidence described next, from Study B, points to a possible boundary to QOC's scope of application. Specifically, the study suggests that QOC's focus on arguing about design spaces defined by multiple Options is poorly suited to work dominated by the evolution of a single Option.

4. PROBLEMS USING QOC IN 'EVOLUTIONARY' DESIGN: TRYING TO ARGUE ABOUT ONE OPTION?

Three sessions were spent with a designer who was working on developing a music composition system in the Smalltalk environment. In session 1, it became clear that many of his ideas were already quite well developed, as a lot of thinking had been invested in the problem beforehand; the main task to which QOC was put was therefore rationalization and decision making. The designer was very positive about QOC's role in this context, and it was clear from the data (Shum, 1991, Case Study 1) that QOC had assisted in drawing out existing but vague ideas, and clarified relationships between Options and Criteria which would have otherwise remained unarticulated (see Figure 6 for an extract from session 1). In sessions 2 and 3, however, serious difficulties were encountered in using QOC, and no explicit DR was constructed.

Let us begin by characterising what will be termed the 'evolutionary' mode of working, that is, the iterative development of what the designer conceptualized as one, complex design Option. The designer spent sessions 2 and 3 developing two representations of two Smalltalk data structures, respectively, a hierarchy of data types, and a table of data types such that each column progressively refined the previous one.

The designer described the method of developing the hierarchy in session 2 as follows:

> [Study B]
>
> what I'm doing is a sort of consistency check – thinking through the implications of what I'm doing – this draft suggestion here. And I'll incrementally alter things *[i.e. the data structure]* – I mean I've already done that many times to get to this stage...
>
> *[points out that he's refining an earlier sketch from his notes]* ...'Gradual refinement' is the phrase. I don't know the Options until I test the previous Option.

In session 3, the content of the problem was different but the mode of working very similar:

> I'm postulating a structure, going round and round testing it, and drawing a few example diagrams of applications of that structure to a real situation – getting it into some concrete familiar objects ... checking that the abstract structure fits that, and changing it if it doesn't.

To summarize, the design problem solving in Sessions 2 and 3 was characterized by the following (c.f. Booch, 1991; Visser, 1990):

- opportunistically driven generate-evaluate cycles to refine the form of the message passing hierarchy;
- use of complex, concrete examples as test cases for the abstract structure;

- management of numerous constraints within the message passing hierarchy;
- application of much implicit Smalltalk programming knowledge.

Given this mode of working, attention now focusses on its apparent incompatibility with the tasks demanded by QOC's explicit, argumentative mode of design. The designer's mode of working should also become clearer through the additional extracts presented.

4.1. Difficulties encountered with QOC constructs

Problems using QOC's three main constructs are illustrated using extracts from the verbal protocol, from which key points are then summarized.

Questions

> [Study B]
>
> What are the decisions that I've made? *[tries to formulate Question]* "Do you have..." it's difficult to put it in terms of that kind of Question... "Is a device configuration a separate class or just one type of category?" Now I don't know how I answer that – it just fits in that it's... there's no real doubt about that, it just fits in consistently.
>
> *E: so you can make up a Question if you have to?*
>
> yes in some cases. Let's see if I can make up any more then ... this idea of having a category of submodules which it's allowed to have – that just arose as a solution to a problem – and what was the problem? The problem was that you have a structure where each object can't just have any old object that's available as its child, only selected ones. And there may be many children ... *[describes relationships between types...]*

There are two ways in which Questions could be used: either by posing an extremely general Question such as **what is the best data structure for a primitive event?** or through a long series of Questions each of which addresses the problem of the current iteration. Whilst on the one hand, a very general Question offers no analytical power to the designer and no insight to someone else trying to understand the design, on the other, the implicit nature of the designer's expertise would make explicit recording of numerous iterations as context-specific Questions unrealistic – the consequences would be enormous DRs, with a proportionate increase in authoring overheads. In sum, since Design Space Analysis Questions are meant to pick out generic or important dimensions of the design, it is difficult to see their useful application in this context.

Options

> [Study B]
>
> ...but as for articulating possibilities – they only arise consecutively. I couldn't have initially said ,"We've got two ways of doing it: like I've done it for *those* messages, or *that [i.e. a hypothetical alternative]* – now let's think which is the better one." The only way you arrive at the second one is by having the first one there and thinking, "Now, still what's wrong with that?" You go more in a linear way than a bifurcated way being implied by your DR scheme.
>
> ... As it's a linear, iterative, refining kind of design, the Options are less useful.
>
> Options are hard to parcel up – often very similar but for one detail. It's almost a problem of notation: how to record each stage, as it's an iterative process.
>
> there's no set of Options – it's just one big mass you have to sort out into categories.

In sum, for the designer, discrete Options were impossible to identify because the design of the final structure was treated as the evolution of one Option over time; the difference between each version was only one, or a few fine details; there were effectively tens of versions marking the path of the design. Thus, even had he wished to, there was no obvious way to express Option evolution.

Criteria

Criteria were the only elements of QOC which could be easily made explicit during sessions 2 and 3. In the extracts shown below, the designer tries to identify Criteria which he has been using up to that point, and is able to identify the main tradeoff between **simplicity** and **non-**

repetition of data. This incident may illustrate the benefits of explicitly considering Criteria: having enlarged on each of the Criteria, the designer concludes that in fact **non-repetition** could be easily satisfied within Smalltalk's environment (through "pointer references"). Although there is little doubt that he knew of this facility, and that **non-repetition** is a good principle in object-oriented programming, it is not clear if up to that point the connection had been made in his mind, such that he no longer worried about **non-repetition of data** as a problem (i.e. as a relevant Criterion).

> [Study B]
>
> I want **independent instrument and event**. The event's got a **signal list**, and I don't want them to be tied together. I suppose I'm trying to have the simplest data structure possible – **simplicity**. I don't want repetition – **non-repetition of data** – I'm trying to express all these positively. On the other hand there is a slight conflict between **non-repetition** and **simplicity** of data: to have **non-repetition** you have to have lots of references to things, which can make it a lot less **simple**.
>
> *E: Do you have a general policy on that for the whole design, or does it depend on each situation?*
>
> In a way I'm still learning about it. As Smalltalk has pointer references anyway, it's actually quite efficient. I can put a new object in, and unless I do a deep copy of the object, it will just be a pointer anyway, so it won't be that... so I think really **simplicity** I'm coming down to is the key element, and I don't care about data apparently being repeated, because it won't actually place much overhead on the system. ...So **simplicity**'s more important than **non-repetition**.

However, despite being able to articulate the main Criteria, the designer made the following comments about making them explicit:

> I don't know *how* I'm making half of these decisions. I think it's a whole block of expert knowledge – well experience – that I've built up of object-oriented programming; having seen examples, it's very difficult to articulate every reason for everything.
>
> *[turns to Criteria noted during the session]* Here we've come up with some Criteria – again useful to have those down, but a lot more difficult to go back over this and explain to people why I've done it this way. All I can say is 'well this one works at this stage.' Difficult to go back to another branch and say well this didn't work because ... everything *[ie. earlier versions of the design]* would work, but this **simplicity** idea is a difficult one to then give alternatives to – it's a subtle one.

Similar comments were made in session 3. In sum, although it was useful to have Criteria recorded, the difficulty in introducing any additional reasoning structure (Questions and Options) meant that Criteria could only be referred to in general terms, applying to the whole structure which constituted the 'Option.'. Later, the designer commented that it was impossible to explain why a structure was 'good' at the level of local decisions. In the final analysis, the only Criterion was 'did it work?':

> [Study B]
>
> ...now there's a very complex relationship as to why that's better – I sort of just hit on it 'cos it seemed to fit in – it just sort of happened. I mean I know how I got it – by working round the problem, drawing examples, and you just get a feel... you abstract from the concrete examples into the structure that will... sort of inductive, whereas I think DR is deductive.
>
> ...the only Criterion is, "Does it enable, or not?" There's not a set of things it could fulfill – it either does or it doesn't. If it doesn't enable you to create a detailed concrete structure, then it's no good — that's the only Criterion!

To conclude, clearly, for any subproblem in design the ultimate Criterion is **does it work?**, but for the purposes of QOC, this needs to be re-expressed as more focussed Criteria which bridge from that goal to specific design features. The designer did make Criteria such as **consistency**, **flexibility**, **non-repetition of data**, and **reduce real time calculation** explicit during sessions 2 and 3, but the difficulty in structuring the deliberation process into subproblems, and the tight interdependencies meant that Criteria like these remained useful only at a global level of application, rather than for alternatives to subproblems within the design space.

4.2. Characterising the relationship between QOC and the two modes of designing

The picture emerging from this and other QOC studies (Shum, 1991), describes the interaction between the way of working which QOC engenders, and two different modes of design. In order to clarify this picture, let us start by considering an extract from the designer's own characterisation of QOC's relation to different modes of designing:

> [Study B: Session 1]
>
> I think almost there's two different... the activity I was doing here was different from what I was doing yesterday, the actual low level structures *[i.e. in work prior to session 1]*. This *[session 1]* is more about big decisions – policies – whereas this *[refers to own notes]* is about implementation. So this *[QOC]* is certainly very useful for strategies.
>
> ... it was quite tricky to think in that way then *[for session 1]* but it was useful... but I think I'd find it almost impossible to wrench myself into thinking in that way *[for sessions 2 and 3]*; it would be unnatural almost, 'cos it's not that kind of path that you take when you're doing this sort of thing.

On two occasions he drew on analogies to describe how he had been working in the latter sessions:

> Have you ever watched someone design a circuit board?
>
> It's more... like painting a picture of something, and you ask why did you put the trees there and not there? There are so many Criteria and they all interrelate.

These images aptly characterize a mode of working for which QOC's approach to design seems ill-matched, given the present evidence. The problem seems to be that argumentation-based DR is based on a model for representing competing alternatives to substantive problems, rather than the evolution of a single alternative over time.

The designer's characterisation of the sessions as "strategic" versus "low level" design captures to some extent the contrast between laying out and deliberating over alternatives at key decision points, and pursuing one Option in detail to better understand its properties. However, this should not be interpreted as "conceptual versus implementational" design; the distinction may be better characterized as breadth-first versus depth-first design. This then allows for either mode of working (laying out of multiple Options, or exploration of single Options) at any point between upstream, high level conceptual design and downstream, low level coding.[5] In reality of course, there will often be an interplay between these modes of working, which suggests that designers intending to use QOC need to recognize when the formalism is likely to be of most use.

A point also worth considering is that most decisions made during evolutionary modes of working (whether high or low level) may in fact not be of interest to other domain experts, because they are in some sense 'routine' – the decisions are not contentious given a certain level of expertise. If this is indeed the case, then the work in evolving *Option X* has value only to the extent that it clarifies for the designer what is meant by "*X*" in relation to *Y* or *Z* as Options to a superordinate ("strategic") Question. It is in working through and laying out the relative merits of each for which structured argumentation may be best suited and most useful, both for the designer and others subsequently.

In the light of this analysis, the relationship between these two modes of working can be thought of as streams of activity occupying different 'spaces' in the DR domain (represented as parallel layers in Figure 7, after Lee and Lai, this volume). These spaces are associated with depth-first and breadth-first modes of designing to different degrees. The Design Space is what QOC represents explicitly; its development is associated most strongly with an analytic frame of mind in which issues are delineated, and Options are laid out for consideration in a breadth-first fashion. Study A and Study B session 1 demonstrated that Design Space Analyses emerge most naturally when the problem demands this mode of working (although as reported, coherent rationale is not 'waiting to be externalised' from designers' minds or 'captured' from ongoing discussion, but is *constructed* in dialogue with colleagues, the problem, and the QOC structure).

---

5 I am grateful to Tom Moran and the anonymous reviewer for drawing out this distinction more clearly.

As Lee and Lai (op cit) have observed, QOC analyses primarily address the issue, alternative, and criterion spaces, (Lee and Lai, Figure 2); these are merged within the Design Space layer of Figure 7.

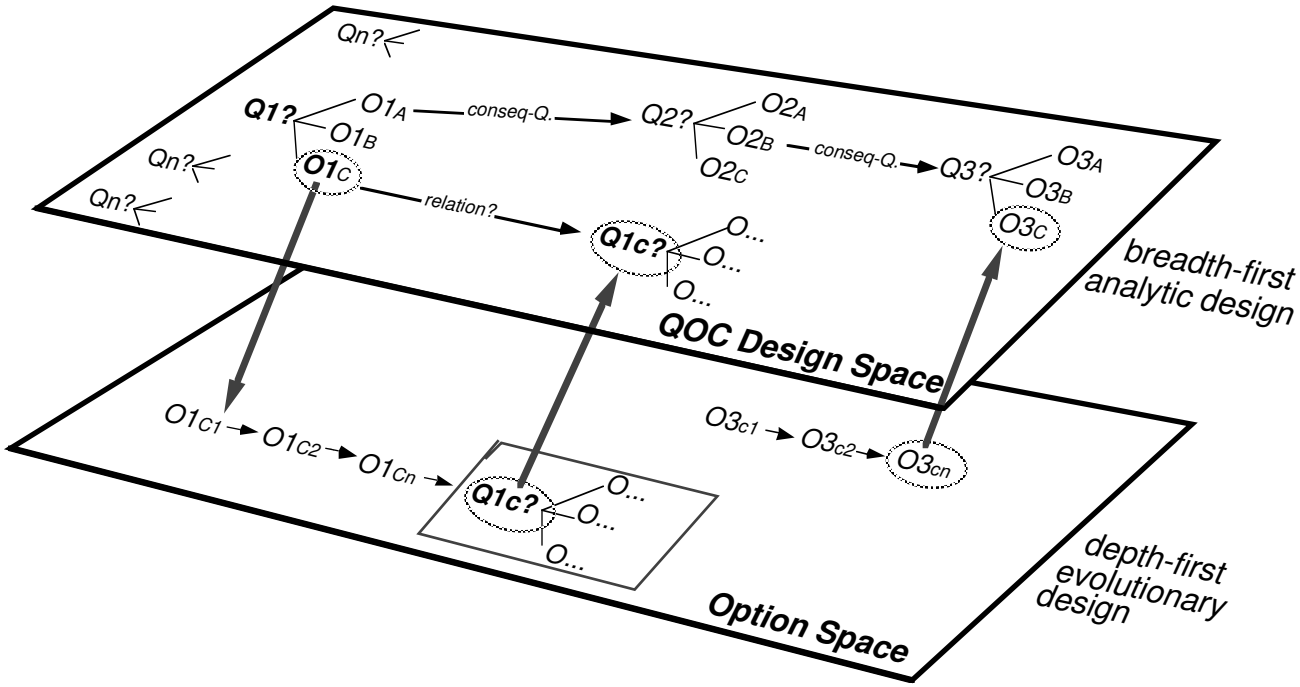

Figure 7: The relationship between a QOC Design Space Analysis, the analytic 'breadth-first' mode of designing, and the evolutionary 'depth-first' mode, as observed in Studies A and B.

The alternative space, by Lee and Lai's definition, allows relations between alternatives to be made explicit. In QOC's vocabulary, Option transformation is not explicitly supported, a property which the *Option Space* in Figure 7 makes explicit. The Option Space illustrates how an evolutionary mode of working – focussing on developing individual Options through iterative redesign and testing – relates to the Design Space with which QOC is primarily concerned. The arrows linking the two spaces reflect the fact that Options may evolve both before they are represented within the Design Space (Option O3C), and after (O1C).

From the point of view of representing a concise, intelligible design space analysis, it is often unnecessary, tedious, or impossible to make successive versions explicit—indeed, a feature of a good design space analysis is that it clarifies the *boundaries* of a given possibility space, rather than detailing the minor differences between features of a local space. However, whilst minor transformations to Options will remain to a large extent implicit, on occasion it will be useful to record a significant development in the design of an Option, and/or why one of several Options has been chosen. At this point, it may be useful to document a decision in the design space (Question Q1C). A design issue which then arises is how such rationale is integrated with the main QOC analysis, given that it derives from a different space to other Questions, and may not make sense if simply added (the relation between O1C and Q1C).

This analysis informs, and is informed by other reports of DR in use, as described in Section 6. Before this, however, attention turns briefly to some results relating to the design of the QOC notation.

5. QOC'S EXPRESSIVENESS

The *expressiveness* of a representation describes its coverage of the target domain to be represented: is the vocabulary sufficient to represent all of the important concepts, relationships, and scenarios of use? Moreover, even if certain classes of information are logically

representable, the extent to which the representation *eases* access (visually, or computationally) to important information, and hides irrelevant detail, is clearly important for an interactive design representation. Reported below briefly are several areas of notational weakness to which the QOC studies drew attention.

5.1. Representing evolution within QOC structures

Much has been said about Option evolution in previous sections. QOC's vocabulary at present cannot easily express *how* Options are transformed, preferring instead to express the *results* of those transformations within the design space – a product as opposed to process orientation. One response to data such as presented in Study B is therefore to leave such deliberation unsupported, the position being that DSA is not intended to support such design tasks, and the evidence indicates that it is a rapid, intuitive process which should not have to bear the weight of explicit representation. This approach may indeed prove to be the most pragmatic.

The alternative position is to provide constructs such as gIBIS' **specializes**, **generalizes**, and **simplifies** relations. Through selective use, significant developments in Options could be made explicit, and later retrieved, perhaps with changing design requirements or constraints. Isolated incidents in Study A suggest that in the absence of notational conventions, 'pseudo-representations' may be invented instead (e.g. Figure 8). As is always the case, the cost of tool support, and any cognitive benefits which the notation provides, is the added overhead of explicit structuring.

An alternative to notational support for version control is support from the environment in which the DR tool runs. A tentative conclusion is that since QOC was developed with simplicity and minimal overheads as a goal, the most promising way to introduce QOC evolution may be through environmental support from generalized hypertext version control mechanisms, rather than through extensions to the notation itself which inevitably introduce user overhead.

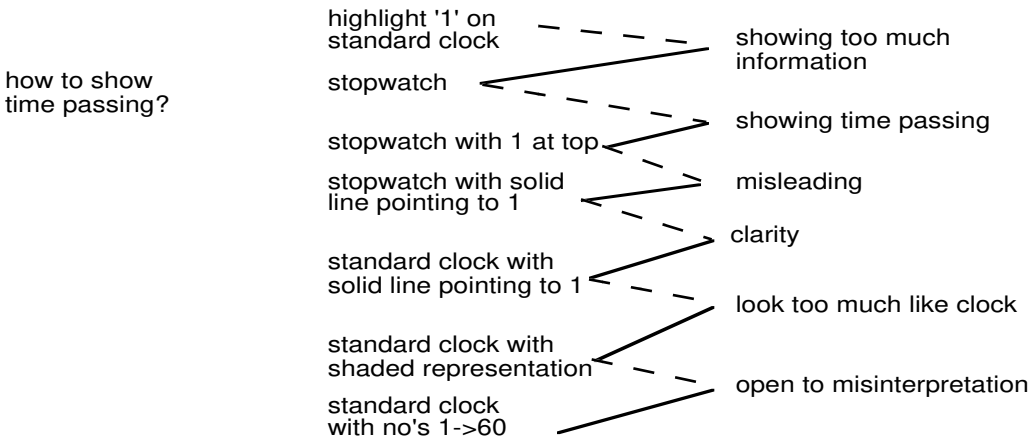

Figure 8: [Study A tutorial exercise: Pair 8] A poor way to capture the evolution of Options within QOC, as used by two designers. Each Option was rejected in favour of a better version, starting from the top. Each Criterion serves only to object to the 'current' Option in order to show the next step of refinement. For instance, the third Option, **stopwatch with 1 at top** was devised because **stopwatch** failed to **show time passing**; however, because it was judged to be **misleading**, **stopwatch with solid line pointing to 1** was devised, and so forth.

Expressively, this is a weak representation: (i) the absence of Assessment links to other Options implies that the Criterion is irrelevant, which is clearly not the case (e.g. **showing time passing** applies to all Options); (ii) it is not shown that subsequent Options in fact satisfy all Criteria more fully than their predecessors.

5.2. Expressing constraints and dependencies

In software design on any significant scale, keeping track of dependencies becomes a major task. In one sense, design is all about discovering and subsequently controlling important dependencies. The work of Lee (1990, this volume) and others exploits the property that by making argument structure explicit, DR tools can provide dependency-management facilities. In

terms of QOC, dependencies manifest as constraints on the selection of Options in different parts of the space about a design.

Not surprisingly, in the course of the QOC studies, there were a number of incidents in which dependencies between Options were encountered. Provided with only the core QOC notation and pen and paper, there was little support for representing dependencies and constraints elegantly. As the designers were not taught any notational conventions, links across the QOC were verbally noted, or ad hoc notational devices invented (e.g. Figure 9).

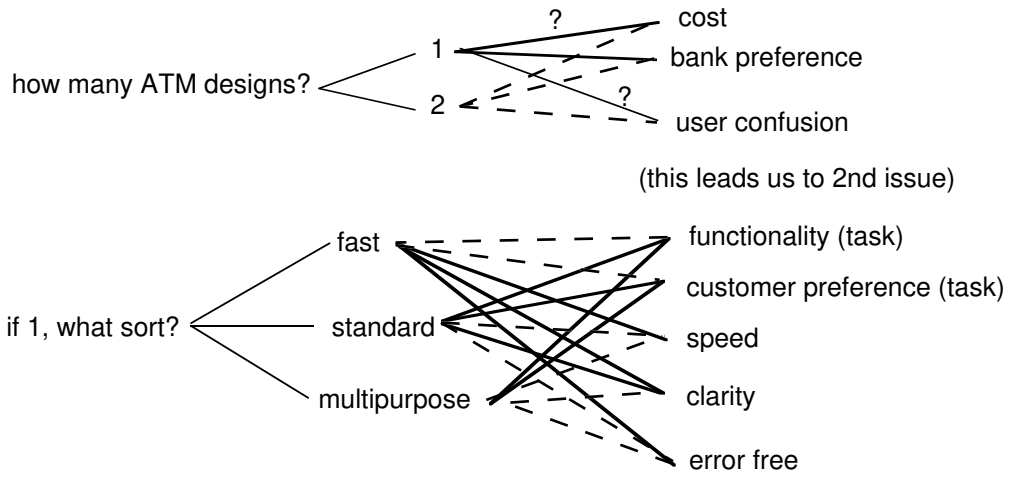

Figure 9: In the absence of QOC-conventions for expressing one Question's contingency on an earlier decision, two designers express the relationship informally through annotation under the first Question, and context-dependent phrasing of the second Question.

5.3. The subtleties of expressing Options, Criteria, and Assessments

If QOC is to be used during design deliberation, it needs to be sensitive to the exploratory nature of the process. Commitment is often delayed as alternative routes are partially developed to assess their potential as solutions. Consequently, it was found that binary choices in the status of entities frustrated designers in a number of respects. For instance, there was not enough expressive power in the *selected/rejected* distinction for Options, leading to the invention of visual devices to reflect indecision (e.g. a dashed box drawn around two Options to indicate tentative selection).

Many comments were also made about the lack of a more sensitive scheme for *evaluating* Options. Firstly, designers pointed to the need to prioritize Criteria, given that simply counting **supports** links was manifestly an inadequate way to make decisions. Given the evidence that this facility is needed, the issue of representational overheads is perhaps not at stake here: designers will make use of different weightings as the need arises.

Whilst Criterion weighting was important, the majority of incidents and comments on QOC's representation of the evaluation space related to the need to show relative strengths of *Assessment* links. Within QOC, Assessments are relative to each other for a given Question, and no semantics can be inferred between Assessments made for different Questions. However as the following examples demonstrate, the formalism runs into problems even in expressing within-Question Assessments.

Designers often described Assessments as being 'more positive' than others, and as with Option selection, invented their own graphical conventions for encoding link strengths such as double-links or thicker links to represent gradations of the **supports** and **objects-to** relations. One pair also made extensive use of what they called 'neutral' Assessments, signified by *omitting* to link an Option to a Criterion. 'Neutral' links were used eleven times in three design exercises, but under several different circumstances. Generally, they served as Assessments for Options which were judged to fall 'in between' other Options which had positive and negative Assessments, that is, Assessments not judged to be sufficiently good or bad to merit a +/– link (Figure 10).

Elsewhere, neutral Assessments were used to mean still other things (*not yet decided* and *not yet discussed*).

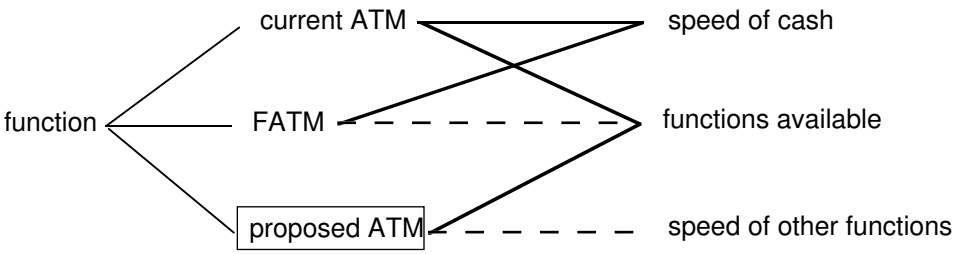

Figure 10: [Study A: Pair 6] An example of the ambiguous use of 'blank links' to reflect 'neutral Assessments.' The Criterion **speed of other functions** 'neutrally' assesses the first two Options, but for different reasons: **current atm** is neutrally assessed by **speed of other functions** as it was judged to be better than **proposed atm**, but not to the extent that it merited a supports link. However, the **FATM** (Fast-ATM) Assessment against that Criterion was also left blank, but in this case because the Criterion was simply irrelevant (the **FATM** only offers a cash service).

Another pair of designers placed '?'s over links to represent undecided status as they worked. However, this occurred under two circumstances, again leading to ambiguity for an outsider: (i) when an Assessment could not be made until further design details had been finalized, and (ii) when they were not confident that an Assessment was correct (e.g. the designers did not feel qualified to make evaluative decisions about psychological Criteria such as **display clarity** or **attentional requirements**).

As indicated, a variety of visual codes can be used to convey relative Assessment strengths. In whatever way it is implemented, the results reported here are evidence that a more sensitive Assessment scheme would be a welcome notational extension, and would not add unnecessary overheads.

6. RELATIONSHIP TO REPORTS OF OTHER DR APPROACHES IN USE

This final section considers reports of other approaches to DR in use, focussing specifically on the interplay between design and representing DR, as expressed in Figure 7. The purpose is not only to draw attention to relevant aspects of other work, but to assess the sufficiency of the analysis as a description of the authoring process for QOC, and argumentative DR more generally. More comprehensive reviews are presented elsewhere (Buckingham Shum and Hammond, 1994; Buckingham Shum, 1995, in press).

Lewis, Rieman and Bell (this volume) have argued that design is inherently problem-centered, and moreover that it is too distracting to abstract from concrete to more general characterisations of those problems as required by most DR notations. In a design project of their own, they expressed DR informally as a series of problems and alternatives. Several patterns were observed in their own deliberations as to how problems, subproblems, and alternatives interacted to move the design forward (e.g. *"micro problem derived from raw problem, design alternatives spawned by micro problem"*). Depending on the nature of the problems, these patterns can be expressed as either rhetorical moves in the Design Space (e.g. *consequent Question derived from more general Question, Options generated for consequent Question*), or as evolution in the Option Space. The observation that design is often problem-centered provides valuable insight into the ways in which Questions, Options and Criteria interact to move a design forward.

Fischer, Lemke, McCall and Morch (this volume) report several studies of design students trying to use PHI (Procedural Hierarchy of Issues – McCall, 1986). It was found that PHI was extremely difficult to use during the actual 'construction' phases of design (i.e. development of the solution, as opposed to preparatory design discussion), although the problem was being decomposed hierarchically into subissues by the designers, which in principle mapped well to the PHI method. Fischer et al. interpret these results in terms of Schön's (1983) conception of the expert's design process. Schön asserts that design comprises several mutually exclusive modes of activity, termed *knowing-in-action*, *reflection-on-action*, and *reflection-in-action*, as described below.

Parallels between Figure 7, McCall's data, and Schön's concepts can be identified as follows. Identification and comparison of Questions and Options corresponds to reflection-on-action – a 'stepping back' in order to consider the nature and scope of the local design space. PHI, like QOC, was well-suited to such deliberation. In terms of Schön's framework, instructions to use PHI as the dominant design model prolonged rationalistic reflection-on-action, sidetracking the students into 'talking round' the actual design (i.e. perpetuating *discussion about* the design space).

Study B clearly showed the difficulty of representing useful DR whilst engaging in 'artifact construction' (the data structures). This rapid testing and changing of the artifact, coupled with a reluctance or even inability to interrupt and articulate one's process is aptly characterized by the concept of knowing-in-action:

> "...the knowing is *in* the action. We reveal it with our spontaneous, skillful execution of the performance; and we are characteristically unable to make it verbally explicit." (Schön 1983, p.25).

DR's interest in the concept of reflection-in-action has already been noted, and this continues to hold in the current analysis:

> In an *action present*—a period of time, variable with the context, during which we can still make a difference to the situation at hand—our thinking serves to reshape what we are doing while we are doing it. I shall say, in cases like this, that we reflect-*in*-action. (Schön 1983, p.26)

In Study B, use of knowledge in this way was intrinsically embedded in the construction/evolution process in the Option Space. Many of the problems encountered were not worth recording – indeed, could not be succinctly expressed (cf. Lewis et al.'s perspective). However, reflection-in-action may on occasion encounter issues worth noting as rationale (e.g. several potentially workable alternatives, or a decision made under extenuating circumstances); reflection-in-action might then become reflection-on-action, moving the attentional frame to the Design Space (represented by Q1C in Figure 7). Two questions which are as yet unresolved, are firstly whether designers are sufficiently aware of what makes useful DR to recognize such situations, and secondly, whether breaking out of construction to reflect via QOC would be unacceptably disruptive. If both of these are the case, one solution might be to briefly record a word or two to capture the problem (i.e. during the "action-present"), to be returned to later for more deliberate reflection.

Turning to another key issue, Conklin and Burgess Yakemovic (this volume) focus on the contrast between "structure-oriented," retrospective DR which emphasizes the capture and communication of the logical content and structure of design reasoning, and "process-oriented" DR which preserves the form and order of ideas in the deliberation process, analogous to a "narrative" with less rationalisation.[6] Conklin and Burgess Yakemovic explicitly state that reusability is a secondary concern for narrative DR, given the process-orientation towards minimising the overheads by minimising the amount of DR rationalisation.

The evidence and analysis from the QOC work clarifies certain implications of this approach. If gIBIS Issues are simply recorded one after another as they arise, with little retrospective analysis, there may be representational problems to overcome in understanding how argumentation of different sorts holds together, that is, the content of the different spaces in Figure 7 will be merged. It is almost inevitable that additional structuring will be necessary to manage the growing DR. A DR environment could therefore support users by differentiating DR from different spaces.

Work related to gIBIS has addressed the development of a real-time collaborative version, called rIBIS. Rein and Ellis (1991) briefly report on users' experiences with the tool over 16 meetings. All but one of these meetings was described as "mostly unsatisfying and frustrating" by their participants, with significant difficulties encountered in using the IBIS method to structure discussions. It was concluded that the main causes of the problems were participants'

---

6 The expressions "capturing" versus "constructing" DR describe points on a continuum of how rationalized the DR is. In principle, of course, all DR is constructed to some extent (Tang, personal communication, June 26, 1992).

inexperience with IBIS notation, and the complexity of the rIBIS user interface. There was no analysis of the nature of the difficulties with IBIS, so these findings cannot be related to the QOC studies in detail; the comments on gIBIS also apply to rIBIS DRs if they are constructed as narrative argumentation. However, the results do serve to confirm firstly, that without sufficient training, even the simpler DR notations like gIBIS and QOC can be unnatural to use, and secondly, that great care needs to be taken in designing the user interfaces to DR tools.

The last work to be considered is that of the claims-based approach to DR, which has been directly concerned with the use of DR in design evolution. Carroll and Rosson (this volume) describe the redesign of a 'View Matcher' support tool, from its original use for novices *learning* Smalltalk, to a View Matcher to aid more expert users in *code-reuse*. They demonstrate that, in combination with their scenario-based design methodology, the structuring of claims by a high level task-analysis produced a representation for DR which helped them to build in a principled manner on the claims analysis of the original design, leading to, "greater confidence that we are standing on the shoulders of our prior View Matcher design work... and not merely in the vicinity of prior work."

In terms of Figure 7, the evolution of the View Matcher corresponds to the exploration of and commitment to portions of a new Design Space with a particular relationship to the original. The possibility of understanding the differences and commonalities between the two spaces is what Carroll and Rosson refer to as understanding the "species" of an artifact. In order to transform one design space into another, new constraints (based on the tasks to be supported) are applied, leading to the development of new portions of the space. The use of claims-based DR to support redesign, as described by Carroll and Rosson, can therefore be seen as operating at the level of the Design Space (evolution of whole Option sets), rather than the transformation of individual Options, as reported in Study B.

As a particular approach, psychological DR through claims extraction provides a more systematic way in which to identify Criteria relevant to user's tasks; it helps to answer the question often asked of QOC, "Where do the Criteria come from?" This question is also addressed by Singley and Carroll (this volume), who distinguish several modes of reasoning which in their experience served as sources of psychological design constraints.

7. CONCLUSIONS

We need to understand the relationship between designers' cognitive representations and the representations we expect them to create and use. This is particularly important in the context of a fast-flowing, time-pressured activity such as software design, in which 'documentation' is already a bad word. Within the scope of the studies reported, the present work is a step towards describing the usability of the argumentation-based formalisms on which much DR (particularly HCI DR) work is based. This chapter has described and discussed the demands of the core representational tasks (the 'nuts and bolts') of QOC authoring, evidence relating to the scope of QOC's application to two common modes of working in user interface and other software design, several areas in which QOC is weak expressively, and considered lastly, consistency with reports of other DR formalisms in use.

There are still many unanswered questions. Analysis of other DR approaches in use would be valuable in assessing the sufficiency of the account presented here. There still remains a critical question mark over the benefits and pragmatics of making the structure of design arguments explicit (Shipman & Marshall, 1992), an often assumed, but as yet largely unproven property 'inherited' from earlier approaches to argumentation tools (Shum and Hammond, submitted). Analyses at this level should complement other work addressing the relationship with 'design' at levels other than the cognitive: the role which DR representations may play in group representational work (Tang, 1991),[7] in project documentation (Shum, Forder, MacLean, & Hammond, 1992) and project management (Conklin and Burgess Yakemovic, this volume), in participatory design (Goldkühl, 1991), and the social and organizational contexts (Grudin, this

[7] Shum (1991) presents a preliminary analysis of ways in which QOC representations were used in group design deliberation, drawing on the framework for shared workspace activity proposed by Tang (1991).

volume; Minneman, 1991) in which DR must operate, and which will themselves inevitably change if DR comes to be used widely.

Acknowledgements. I am grateful to many people for critical comment and discussion on the content and form of this chapter, particularly Jack Carroll, Nick Hammond, Allan MacLean, Andrew Monk, Tom Moran, and the anonymous reviewer. Thanks are also due to Logica Cambridge Ltd., Warehouse, University of York Music Technology Research Group, and the Impact Project at Nestlé Rowntree, York for providing designers.

Support. This research was funded by Rank Xerox EuroPARC, Esprit Basic Research Action 3066 (The AMODEUS Project), and SERC CASE Award 88504176.

8. REFERENCES

Bellotti, V. M. E., MacLean, A., & Moran, T. (1991). Structuring the Design Space by Formulating Appropriate Design Rationale Questions. *SIGCHI Bulletin*, 23, 4, 80-81. Extended version available as: Amodeus Working Paper RP6/WP6. Cambridge, UK: Rank Xerox EuroPARC.

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

Buckingham Shum, S., & Hammond, N. (1994). Argumentation-Based Design Rationale: What Use at What Cost? *International Journal of Human-Computer Studies*, 40, 4, 603-652

Buckingham Shum, S. (1995, in press). Design Argumentation as Design Rationale. To appear in: A. Kent and J. G. Williams (Eds.), *The Encyclopedia of Computer Science and Technology*. New York: Marcel Dekker, Inc.

Conklin, J., & Burgess Yakemovic, K. C. (1991 & this volume). A Process-Oriented Approach to Design Rationale. *Human-Computer Interaction*, 6, 3&4, 357-391.

Fischer, G., Lemke, A. C., McCall, R., & Morch, A. I. (1991 & this volume). Making Argumentation Serve Design. *Human-Computer Interaction*, 6, 3&4, 393-419.

Goldkühl, G. (1991). Information Systems Design as Argumentation: An Investigation into Design Rationale as a Conceptualization of Design. *14th IRIS Conference (Information System Research In Scandinavia).*

Grudin, J. (this volume). Evaluating Opportunities for Design Capture. In T. P. Moran and J. M. Carroll (Eds.), *Design Rationale.* Hillsdale, NJ: Lawrence Erlbaum Associates.

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

Jansson, D. G., & Smith, S. M. (1991). Design Fixation. *Design Studies*, 12, 1, 3-11

Lee, J. (1990). SIBYL: A Qualitative Decision Management System. In P. Winston and S. Shellard (Eds.), *Artificial intelligence at MIT: Expanding Frontiers* (pp. 105-133). Cambridge, Massachusetts: MIT Press.

Lee, J., & Lai, K. (1991 & this volume). What's in Design Rationale? *Human-Computer Interaction*, 6, 3&4, 251-280.

Lewis, C., Reiman, J., & Bell, B. (1991 & this volume). Problem-Centered Design for Expressiveness and Facility in a Graphical Programming System. *Human-Computer Interaction*, 6, 3&4, 319-355.

MacLean, A., Bellotti, V. M. E., & Shum, S. (1993). Developing the Design Space with Design Space Analysis. In P. F. Byerley, P. J. Barnard & J. May (Ed.), *Computers, Communication and Usability: Design Issues, Research and Methods for Integrated Services*, 197-219. Amsterdam: Elsevier: North Holland Series in Telecommunication.

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

McCall, R. J. (1986). Issue-Serve Systems: A Descriptive Theory of Design. *Design Methods and Theories*, 20, 8, 443-458

McCall, R. J. (1991). PHI: A Conceptual Foundation for Design Hypermedia. *Design Studies*, 12, 1, 30-41.

Minneman, S. (1991). The Social Construction of a Technical Reality: Empirical Studies of Group Engineering Design Practice. (Technical Report SSL-91-22). Palo Alto, CA: Xerox Palo Alto Research Center.

Rein, G. L., & Ellis, C. A. (1991). rIBIS: A Real-Time Group Hypertext System. *International Journal of Man-Machine Studies*, 24, 3, 349-367. Also in Greenberg S. (Ed.) *Computer Supported Cooperative Work and Groupware*, 223-241. (1991) Academic Press: London.

Rittel, H. W. J., & Webber, M. M. (1973). Dilemmas in a General Theory of Planning. *Policy Sciences*, 4, , 155-169. Reprinted in: *Developments in Design Methodology*, 1984, 135-144.

Schön, D. A. (1983). *The Reflective Practioner: How Professionals Think in Action*. New York: Basic Books.

Shipman, F. M., & Marshall, C. C. (1992). Formality Considered Harmful: Experiences, Emerging Themes, and Directions. *Technical Report.* Xerox Palo Alto Research Center

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

Tang, J. C. (1991). Findings From Observational Studies of Collaborative Work. *International Journal of Man-Machine Studies*, 34, 2, 143-160. Also in Greenberg S. (Ed.) *Computer Supported Cooperative Work and Groupware*, 11-28. (1991) Academic Press: London.

Visser, W. (1990). More or Less Following a Plan During Design: Opportunistic Deviations in Specification. *International Journal of Man-Machine Studies*, 33, 3, 247-278.