The AMODEUS Project
ESPRIT Basic Research Action 7040

Integration Techniques for Multi-Disciplinary HCI Modelling: A Survey

The Amodeus-2 Consortium

Editor: D.J. Duke
Department of Computer Science
University of York, Heslington, York, YO1 5DD, UK

June 28, 1995

Integration and Design: ID/WP48

AMODEUS Partners:
MRC Applied Psychology Unit, Cambridge, UK (APU)
Depts of Computer Science & Psychology, University of York, UK. (YORK)
Laboratoire de Genie Informatique, University of Grenoble, France. (LGI)
Department of Psychology, University of Copenhagen, Denmark. (CUP)
Dept. of Computer & Information Science Linköping University, S. (IDA)
Dept. of Mathematics, University of the Aegean Greece (UoA)
Centre for Cognitive Informatics, Roskilde, Denmark (CCI)
Rank Xerox EuroPARC, Cambridge, UK. (RXEP)
CNR CNUCE, Pisa Italy (CNR,CNUCE)

Integration Techniques for Multi-Disciplinary HCI Modelling: A Survey

D.J. Duke, M.D. Harrison & J. Good — Dept. of Computer Science, University of York, U.K.
J. Coutaz, L. Nigay & D. Salber — LGI, Université Joseph Fourier, Grenoble, France.
G.P. Faconti & F. Paterno' — C.N.R.-CNUCE, Pisa, Italy.
D.A. Duce — Rutherford Appleton Laboratory, Didcot, U.K.
A.E. Blandford, P.J. Barnard, J. May & R.M.Young — MRC Applied Psychology Unit, Cambridge, U.K.
A. MacLean & V. Bellotti[1] — Rank Xerox Research Centre, Cambridge, U.K.
N.O. Bernsen, F. Jensager, S. Lu, M. May & J. Ramsay[2] — CCS, Roskilde University, Denmark.
J. Löwgren, S. Hagglund,
T. Timpka & C. Sjöberg — Dept. Computer Science, Linköping University, Sweden.
A.H. Jorgensen & A. Aboulafia — Dept. of Psychology, Copenhagen University, Denmark.
N. Hammond & S.J. Buckingham Shum — Dept. of Psychology, University of York, U.K.
J. Darzentas, J. Darzentas,
T. Spyrou & E. Banaki — Dept. of Mathematics, University of the Aegean, Greece.
J. Nielsen — Copenhagen Business School, Denmark.
S. Verjans — Katholieke Universiteit, Leuven, Belgium.

Abstract: A growing number of HCI models provide insights into human- and system-oriented aspects of design artefacts. Experience with modelling in practice suggests that the more difficult design problems are cross-disciplinary, and there is a need to be able to integrate both the results and the process of HCI modelling. This document reports on the techniques for integration that have been developed and applied within a multi-disciplinary HCI project. We describe the techniques, their roles within the project, and where appropriate the results of using specific techniques to integrate multi-disciplinary modelling of large design spaces. We draw some conclusions about the relationship between techniques, their role in the design process, and their future potential.

1 Introduction

Amodeus-2 is a multi-disciplinary project with a remit to extend HCI modelling techniques to address design issues surrounding interactionally sophisticated technology. In addition, we are concerned with the problem of integrating these multiple viewpoints within the design process, and in assaying the effectiveness of the techniques when applied to practical design problems. The objective of this paper is to describe the integrative aspect of the project, and to set out at least a coarse framework for comparing the different approaches. It is not intended to be an objective assessment of the techniques that have been developed and used - they are still too immature for that kind of assessment. Our aim, rather, is to provide a survey of the approaches used to integrate the analyses of case studies undertaken by the Amodeus

[1]Now at Apple Computers Inc, Cupertino, California, U.S.A.

[2]Now at South Bank University, London, U.K.

modellers. We report on the perceived benefits and difficulties involved in using the various frameworks, and draw some tentative conclusions about the potential scope for the integration of multi-disciplinary modelling in HCI.

Section 2 places this report in context. It provides a brief description of the modelling techniques used within Amodeus, and the case studies or exemplars to which they have been applied. It also introduces the integrative techniques that will each be described in this paper. A significant problem in conducting any form of comparative assessment is that integrative techniques vary widely, both in terms of their stage of development, and in their role in the design process. DR/QOC for example is an established framework for expressing design rationale in terms of questions, options, and the criteria used to assess options. Its use within the project, described in Section 3, is as a way of expressing and integrating the results of multi-disciplinary analyses; DSD (Section 4) is similar in this respect. Co-Modelling (Section 5) is, in contrast, a highly experimental process that aims to integrate insight from multiple perspectives during development of models. Each of these techniques can be thought of as 'global' in that they are (relatively) independent of the modelling techniques over which they integrate. Section 6 summarises the lessons learnt from this form of integration before we move on to consider 'local' integration techniques. These are approaches that bring together specific modelling techniques. In this way local integration can take advantage of the particular properties of its starting points. Four such techniques developed and used by groups within the project are discussed in Section 7. Two software tools that demonstrate quite different approaches to supporting multi-disciplinary modelling are then discussed in Section 8. Tools provide one means of transferring results or techniques from theory into design practice; Section 9 draws out some more general issues concerning the prospects for transferring these techniques to designers. The conclusion in section 10 summarises our experience in bringing disparate modelling techniques to bear on significant design problems within HCI.

2 Modelling and Integration: An Overview

The field of human-computer interaction contains a wide range of theories that seek to represent, explain and predict various aspects of interaction. However no single theory is ever likely to cover all issues relevant to a given design. Human factors analysts need to consider multiple views on a design artefact, and thus there is a need to understand how these views can be integrated within design. For instance, do they provide consistent insights? In trying to answer this and other questions about integration we will often refer to various human roles, specifically: modellers, integrators, designers, and users. Informally,

- A *modeller* is someone who, during the design of an artefact, constructs a representation that is used to assess the artefact against certain criteria or requirements.

- An *integrator* is a person who combines different modelling techniques; types of integration are discussed below.

- A *designer* is someone who constructs an artefact, either to meet user requirements, or possibly as part of a technology driven research or development programme.

- A *user* is anyone with an interest in the design of an artefact. This ranges from 'end users' who may have to use the artefact directly to accomplish tasks with a work domain through to the personnel who have to maintain and support the operation of the artefact.

Although the roles are distinct, we acknowledge that any individual may fill multiple roles during the design of an artefact, either sequentially and/or concurrently. For example, while it is often recommended that users should participate in design as distinct parties, separate from designers, there may be cases where the designers are the users, for example in the development of a tool to support the design process. It is more likely, given constrained availability of human resources, that modelling and integration roles will be shared by individuals, perhaps by a human-factors analyst within an organisation.

This report considers three levels at which integration can take place:

1. **Integration of result**. Integration takes place after modelling. The modellers represent and assess design issues separately, then integrators attempt to combine and relate the contributions of the modelling approaches.

2. **Integration of process**. Integration occurs during modelling. The modellers work on distinct representations of the design artefact, but during the construction and assessment of the model assume the role of integrators by communicating questions and issues either informally or through some common notation.

3. **Integration of theory.** Integration takes places before modelling. Distinct design representations are integrated by embedding them within a common language or shared theory. This does not mean that it is possible to define a 'unified' theory covering all aspects of HCI; such 'local' integration is only effective where the models operate on concepts that can be inter-related.

The concept of global integration mentioned in the introduction corresponds to levels 1 and 2, while local integration, which makes use of the semantics of the techniques, corresponds to level 3. Levels 2 and 3 require that the modelling techniques involved are applied concurrently, or at least within the same part of the overall design process. Later in the paper (Section 6) we discuss whether iterations of modelling with integration by result can yield an effect like that of integration by process.

2.1 Modelling Techniques

Amodeus-2 employs six main approaches to modelling aspects of human-computer interaction. These can be classified broadly as either *system* models that represent the structure and performance of software and/or hardware devices, *user* models that address the cognitive resources and knowledge that a user of a system will need to employ in order to achieve domain-relevant goals, and *representational* modelling that considers the modalities of human-system interaction. In the summaries that follow, each approach is listed together with references to selected papers that describe or illustrate that approach. Further details of each modelling approach and examples of their use can be found in the Amodeus-2 Executive Summaries [bs94]. These are also available in hypertext form via the world-wide web, as described at the start of the reference section.

2.1.1 System Modelling

A characteristic shared by the Amodeus-2 system modelling approaches is that they are based on a notion of agent [c87] or interactor [dh93, fp92]. This is conceptually similar to the object model used in software engineering (e.g. [b91]), but more specialised in that agents and interactors identify a presentation through which information about the internal state of the interactor is made perceivable to users.

LIM (Lotos Interactor Model): this model uses the formal description technique LOTOS to model a system as a collection of interactors, and to express requirements on the behaviour of those interactors using machine-checkable logics such as ACTL.

See also: pf92, ffz94 (interactors), p93 (properties), pm94 (ACTL), pm95 (case study)

FSM (Formal System Modelling): in this approach a model-oriented language such as Z [s92] or MAL [rfm91] is used to describe the relationship between internal and perceivable states of an interactor and the actions that modify the states.

See also: dh93 (interactors), dh94a (presentations), dh94b, dh95 (case studies).

PAC (Presentation, Abstraction and Control): PAC (Presentation, Abstraction, Control): A software architectural model that recommends to structure an interactive system into a collection of specialised agents that produce and react to events (i.e., stimuli). Each agent has three 'facets' and all agents communicate through their Control facet. PAC-Amodeus is an extension of PAC and is a blend of the components advocated by Arch and the refining process in terms of agents advocated by PAC.

See also: cns93 (MSM), c93 (conceptual architecture), snc94 (extending PAC), nc95 (fusion).

2.1.2 User Modelling

User models are concerned with representing and analysing aspects of human cognition that are significant for carrying out interaction with a design artefact.

CTA (Cognitive Task Analysis): CTA is a systematic approach to evaluating the information and mental resources that a user needs in order to understand and interact with a given interface. Its theoretical basis is ICS (Interacting Cognitive Subsystems), a unified cognitive theory.

See also: bm95 (ICS concepts and definition), mb95, msb95 (application), mb94 (case study)

PUM (Programmable User Models): The focus of PUM is on the goals that a user wants to achieve with a given interface, the tasks that will support those goals, and the information that the user has or can obtain in order to plan.

See also: ygs89 (concepts), by95a (applications), by95b (case studies)

2.1.3 Representational Modelling

Representation-modelling approaches focus on the input/output modalities used to represent information at the human-computer interface.

IMAP (Information Mapping Methodology): IMAP systematically tries to map task domain information to input/output representations. The IMAP-rules used to perform this mapping are deduced from Modality Theory.

See also: bv95a (IMAP-introduction), b94 (modality theory), bv95b (case study)

2.2 Common Exemplars

Significant effort within the Amodeus project has been directed at the analysis of three ‘exemplar’ systems, case studies that served each of the projects three main objectives:

- extending the scope of modelling techniques to address user-oriented issues of sophisticated interface technology,
- developing frameworks to support the integration of multi-disciplinary modelling, and
- transferring the results and techniques of modelling into the design community and assaying the utility and effectiveness of each approach.

The second objective is the focus of this paper, and to understand some of the problems and achievements of the integration effort it is useful to discuss briefly the nature of the exemplars.

ECOM The European CSCW Open Development Environment (EuroCODE) is an Esprit project (6155) aimed at developing a set of computer-mediated communication tools to support the Danish Great Belt bridge and tunnel construction project. One such tool is a media-space, based on existing work such as Portholes [db92] and Raven [gmm+92]. One of the Amodeus partners (Rank Xerox EuroPARC) was involved in the design of the user interface, called ECOM, for the media-space, and AMODEUS was asked to provide an assessment of the initial prototype. The design process for ECOM is both imitative, drawing on ideas from existing media-space systems, and exploratory, with extensive use of design rationale and prototypes.

See also: bbm94

CERD The Computer Entry and Readout Device is one component of a large information system called CDIS developed for the British Civil Aviation Authority by Praxis Systems Ltd, a software engineering company located in Bath, U.K. CERD is a touch sensitive panel used by Air-Traffic Controllers to inspect and modify the approach sequence of aircraft into major airports. The designers of the CERD employed rigorous software engineering practices including the use of formal methods (specifically, VDM) to describe the system's functional behaviour. Amodeus has been involved in carrying out user-oriented specification and assessment of the interface.

See also: bsd+94

MATIS The Multi-modal Air Travel Information System is an experimental system, developed at LGI in Grenoble (an Amodeus partner) and Carnegie Mellon University in the United States, to experiment with and evaluate techniques for integrating different modalities within human-computer interaction.

See also: ncs93

The exemplars thus span not only a wide range of interfaces and design issues, but also very different types of design process. We will discuss the significance of this observation in Section 9, where the prospects for the take-up and development of the integration techniques are discussed.

2.3 Integration Techniques

One initial focus of integration effort within Amodeus-2 was on two techniques, QOC and DSD, that had already found use for capturing and recording design rationale outside the specific scope of multi-disciplinary HCI. That is, they are design notations with application beyond their role as integrative frameworks. Both are *result-oriented* techniques that come with a means of representing the overall structure of a design space.

QOC (Questions, Options and Criteria). QOC organises the design space in terms of design issues (the questions), options for resolving issues, and the criteria which either support or argue against these

options. The space is hierarchical in that options can have consequent questions; the framework provides a graphical and tabular notation for relating the three aspects.

See also: mym89, myb+91 (definition), bm94, bbm95 (experience)

DSD DSD/DR, as developed in AMODEUS 2, plays a dual role in design support. Firstly, it provides temporal and logical structure to design processes through its two notational frameworks: one for design space structure and one for design rationale representation. DSD/DR provides structure to design spaces, in that it assumes that artefacts are complex products of an optimisation process which take into account such factors as COSITUE (Cooperation, Organisation, System, Interface, Task, User and user Experience), and that artefact design aims to optimise user performance on the intended artefact. Secondly, DSD/DR provides design support, in the form of a shared integrative platform, for representing the output of multi-disciplinary modelling analyses.

See also: br94, rb95 (definition and guidance), rb94 (case study report)

In parallel with the use of QOC and DSD, the group at Linköping were investigating the use of a similar framework to capture the evolving commitments and trade-offs inherent particularly at the early stages of design, for example at an organisational level. Like QOC and DSD, argumentative design (aRD) is neutral with respect to the kinds of analysis that it represents, again using natural language or a semi-structured tabular notation. In contrast to the result-oriented models mentioned above, aRD emphasises the iterative process through which design decision are reconsidered or modified in the light of new input, and for this reason it is classified in this document as a *process-oriented* technique.

aRD aRD (Argumentative Design) provides multi-disciplinary groups (consisting of, for example, software engineers, human factors designers, and a client's managers and employees) with a framework for discussing design proposals in organisational settings. It provides graphical visualisation aids which encourage the discussion of design issues on equal terms by all parties (in contrast to many software development representations), and it provides rules and agendas for collaborative design meetings between such parties.

See also: s94 (definition), s95 (application)

One of the problems identified early in the project was that there are often strong dependencies between modelling approaches. For example, in claiming that some feature of a system will present a specific problem for users, a PUM or CTA analysis may be making implicit but unfounded assumptions about the behaviour of the software system. Similarly, a system modeller may claim that an interface is appropriate because it satisfies some general property such as conformance between the state and presentation [dh94a]. However, unless the cognitive consequences of that property are known with respect to the system under discussion, it is possible that the claim can not be justified. The 'simple' solution of documenting assumptions is impractical here; it may lead to considerable waste of effort, and one modelling approach may be blocked simply because it lacks information or support available from another. This motivated the development of a new process-oriented framework:

Co-modelling. In this approach user and system modellers focus their analyses on a specific set of design issues. Under the guidance of a co-ordinator, the modellers contribute towards a shared view of the problem, building iteratively on each others' analyses.

See also: ybd+94 (definition and case study)

In the course of the project, development of individual modelling approaches and enriched understanding generated by co-operation on the shared exemplars led to the development of 'local' integration techniques,

mostly at the level of linking the theory underlying particular models. While these techniques are less general than the result and process-oriented frameworks, their notations and concepts provide greater intellectual 'leverage' for tackling difficult or specific issues of human-system interaction.

IF (Interaction Framework). In contrast to DSD and QOC, IF is specifically aimed at the problem of expressing 'neutral' requirements on the interaction between multiple agents. Rather than integrating the results of modelling analyses, it is intended to identify requirements that can then be developed within other models. Interaction Framework is not a 'finished' framework, but a topic of ongoing research.

See also: bh89, hbb94, bhb95

CARE Another approach to integration is exemplified by work on the CARE (Complementarity, Assignment, Redundancy and Equivalence) properties. In this case, properties of the computer system are considered also from a user perspective, and the corresponding user properties are identified and defined. Thus we acquire system- and user- perspectives on a set of properties of the interaction.

See also: cns+95

Syndetic Modelling. This approach integrates the contrasting representations of user and system modelling into a single model. By expressing the behaviour of the user and system in a common formal language, properties of an interactive system can be described and understood in terms of the conjoint behaviour of both agents.

See also: dbd+95 (concepts and case studies), d95 (application to gesture).

Semiotics. Semiotics is the science that studies human and non-human signs as systems of communication, meaning and interpretation. Through its focus on the sign as the central concept of the man-machine interface, semiotics serves an integrative function, for instance by building a library containing all system- and user-aspects of each particular sign and their combinations.

See also: e76 (semiotics), a90, v94 (computer semiotics)

Sections 3 and 4 each focus on a particular approach to result-based integration, respectively QOC and DSD, as these are effectively the most 'mature' of the techniques covered in this paper. Process-based integration, covering Co-modelling and aRD, is discussed in Section 5. Each approach is described in more detail, and is illustrated through an example from one of the project-wide exemplars. Where appropriate, commentary from modellers about their experience with the technique has been included as a subjective account of the project's own experience in using the technique on a 'real' example. The 'local' or 'semantic-based' techniques for integrating HCI theory are each discussed further in Section 7, following a review of the global approaches in Section 6.

3 Integration by Result I: QOC Design Space Analysis

3.1 Questions, Options and Criteria: A Description

MacLean, Young, Bellotti and Moran [myb+91, mym89] present QOC as an approach to representing design rationale for software design decisions, with particular emphasis on HCI design. QOC is an *argumentation-based* approach [bsh94], that is, it seeks to assist reasoning by encouraging designers to break their arguments down into constituent parts, thus making the structure of their reasoning more explicit. In QOC, arguments are broken into four key components: *Questions* are used to encapsulate key issues which shape the design, *Options* are alternative answers to Questions, and *Criteria* are appealed to in choosing one Option over another. Fourthly, *Assessments* are the relationships between Options and Criteria (*supports* or *objects-to*). These elements are summarised in Figure 3.1. Boxed Options indicate a decision, or at least a working commitment.

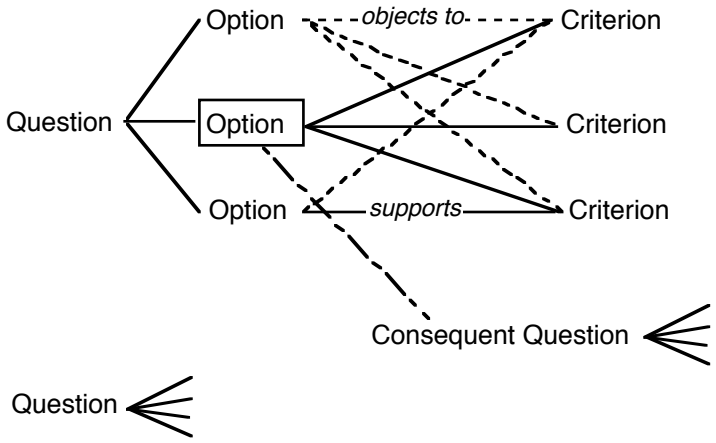

Figure 3.1: The vocabulary of the QOC notation, used to represent Design Space Analyses.

The approach asserts that a design rationale can effectively serve its purpose – that is, answering *Why?* questions about a design – if it clarifies how the design sits in relation to alternative designs in the domain. This *design space* can be used to compare designs and alternative ways of implementing necessary functionality. The design space can never be represented in its entirety, as one can continue to ask Questions to an infinite level of detail, and from numerous perspectives (e.g. implementational vs. human factors).

QOC is able to serve a modelling-integration role in AMODEUS by expressing the output of very different modelling approaches in a common form—as semiformal design arguments. In this role, the QOC design space around a particular design problem serves as a common point of reference for the different modelling groups. Each modelling group then conducts its analysis of the design problem (as expressed by the QOC Questions), and then translates the resulting insights into QOC compatible terms. Thus, the modelling may highlight previously unaddressed issues (new Questions), suggest new ways of implementing a solution (new Options) or new trade-offs which must be negotiated (new Criteria).

3.2 Example: the ECOM digital media space

The modelling of the ECOM media-space interface involved the following overlapping phases: QOC analysts monitored the ongoing progress of ECOM's design, by interviewing the design team, attending meetings and examining documents and the evolving user interface prototype. To ensure that the AMODEUS modellers were working with real problems specified by designers, rather than issues which they might select to show off the strengths of their analytic techniques, we asked each ECOM designer at EuroPARC what they felt were key issues for the design of the ECOM media space. They raised three important issues for media spaces which became the basis for focusing analytic work in AMODEUS:

- Making and breaking AV connections.
- Displaying general levels of availability for communication, and controlling the access that other users within the media space have to oneself (see Figure 3.2).
- Controlling audio and video parameters (frame rate, quality and bandwidth).

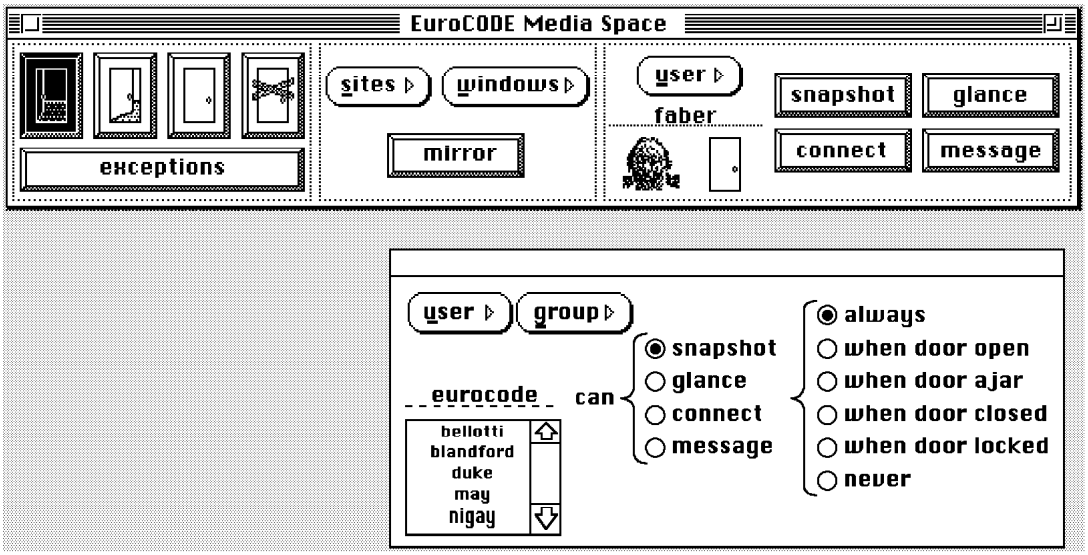

Figure 3.2: The ECOM media space prototype user interface, showing the interface to ECOM's person-specific access control mechanism. By hitting the **exceptions** button in the main dialogue box (top left), the user brings up the exceptions dialogue box (right hand side of the screen). They then select a **user** or **group** (members are displayed in the scrolling window), followed by a connection type (one of **glance**, **snapshot**, **connect** and **message**), and finally a state with which this connection exception is to be associated.

A QOC design space was initially developed to represent the *designers'* rationale around these issues. As noted above, this was based on notes from meetings and discussions with the ECOM designers and another media space designer at EuroPARC, plus research literature and experience of media space use. The design space was verified by the designers as a valid summary of their thinking about the prototype. It consisted of about one hundred QOC Questions with related Options and Criteria. Figure 3.3 gives an example of one of them.

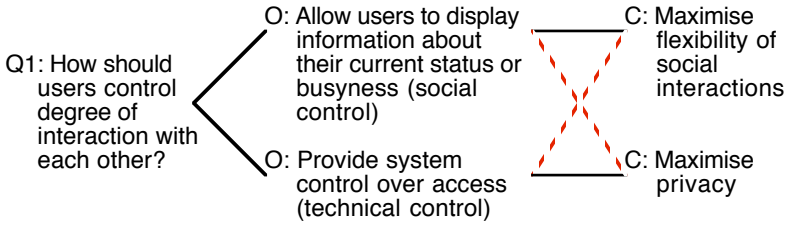

Figure 3.3: Example of QOC used to represent the designers' own rationale, prior to modelling. Criteria (C) are the bases for deciding among alternative Options (O). Questions (Q) identify issues which structure alternative Options. Solid Option-Criterion links are positive assessments, and dashed links are negative.

The modellers were asked to assess the ECOM design from their own perspective and to submit a report of their contributions to assist in refining the prototype. They were instructed to focus on the issues, alternatives and assessments represented in the design space and to provide input in QOC, or QOC compatible terms, in order to avoid misinterpretation of their design contributions by any third party. This enabled the design space analysts to insert these different contributions (with page references back to the analytic reports) into the design space to show how they related to each other and to the ECOM designers' thinking. This *modelling-enriched design space* then served as an overview of the modelling, and as an index into the set of modelling reports.

By way of example, the cognitive modellers showed that ECOM made it hard to separate the task goal of making oneself more or less *generally accessible* (i.e., available) from that of choosing to be more or less *accessible to a particular individual or group*. The modellers proposed controlling general and specific accessibility levels separately. Furthermore, the user's current accessibility would be hard to keep track of due to the complexity of the possible exceptions. They recommended that the setting of person-specific exceptions should only be possible at one level of general availability. Figure 3.4 shows an example of the way in which this modelling highlighted the problem of confusing general availability and person-specific accessibility once it was integrated into the initial design space as shown in Figure 3.3. These analytic contributions were always indexed back into their source in a modelling report so that details of the modelling could be examined. A comprehensive account of this approach to integration, focusing on the specific problem of controlling accessibility and availability in a media space, has been reported in [bb+95].

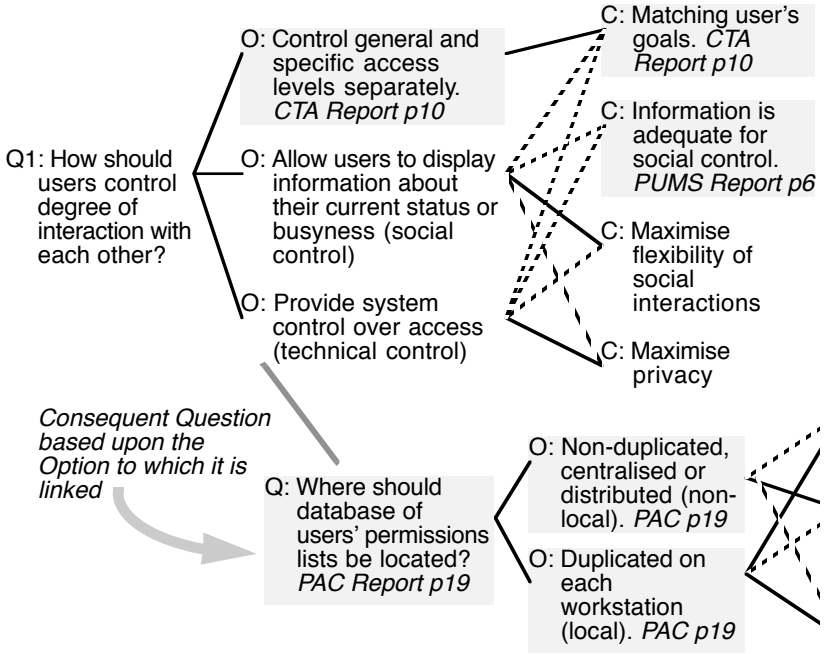

Figure 3.4: Example of how the QOC design space was used to summarise modelling analyses. In this example, the analysis in Figure 3.3 has been added to by the PAC architecture modellers and the CTA and PUM cognitive modellers.

3.3 QOC/DSA as a modelling-integration scheme

One purpose of the integration effort in AMODEUS is to understand how modelling approaches relate to each other. A central part of this is to explore the properties of different integration representations and techniques for integrating the output from multiple modelling techniques. This section discusses the value of using QOC/DSA as such an integration scheme. We summarise the main conclusions we have drawn from our experiences throughout AMODEUS-1 & 2 in using QOC for modelling-integration.

The main points which need to be brought out are organised into two broad categories: QOC's *strengths* as a representation for modelling-integration, and its *weaknesses* which, at least at present, cause difficulties. Measures for overcoming these weaknesses are considered. Unless stated otherwise, most of the conclusions are drawn from the largest QOC integration exercise (analysing ECOM). This brought to the fore most of the issues of which we were already aware from earlier, smaller modelling-integration exercises, as well as highlighting a number of others related to the complexity of managing much larger QOC representations.

Fuller reports and discussion of the work on ECOM can be found in [bbm94, bbm95, bm94], and for an example of an earlier QOC modelling-integration exercise, see [b93] AMODEUS Working Papers presenting QOC modelling-integration on other HCI design problems are listed in the Appendix.

3.3.1. QOC for modelling-integration: Strengths

Provides succinct summary: An obvious feature of QOC is its terseness, compared to, for example, the content of DSD frames as described in Section 4. A QOC structure shows the *product* of a modelling exercise, as regards the design problem in question – what does Modelling approach X have to contribute in terms of Qs, Os and Cs? However, as used in the ECOM exercise, links from the QOC are needed to reference the source material (modellers' reports). This betrays QOC's weakness with respect to its terseness, as discussed below – often it is hard to express or understand the modelling expressed purely as QOC.

Clarifies impact of contribution on design space: By stripping out the modelling process and detailed discussion, and by defining an agreed upon, common representation of the design problem, a QOC design space shows what overall impact the modelling makes to the design space. That is, it shows from a particular modelling viewpoint, which *issues* (Questions) and *trade-offs* (Criteria and Assessments) are at stake, and thus *what are the important differences between design solutions* (Options).

Highlights weak/incomplete design reasoning: The argument-structure used by QOC can highlight confirmation bias (consideration of only evidence which supports one side of an argument) since an excessive tendency towards only positive or negative assessments will show up visually. For an example of this, see the QOC analysis of user interface designers' decisions reported in [mby90]. In the context of ECOM modelling integration, QOC highlighted modellers' tendency towards negative critiques of the designers' decisions.

Highlights modelling inter-relationships: In the ECOM modelling-integration, there were instances where representation as QOC highlighted modelling inter-relationships of interest:

- *complementarity*: the presence of input by multiple modelling techniques into different parts of the QOC design space expresses visually that the techniques address different aspects of a particular problem; however, since complementarity can occur for a range of reasons, whilst easy to detect in a QOC diagram, this kind of relationship does not itself represent very strong integration between modelling techniques.

- *synergistic Criteria*: since QOC Criteria are informal (natural language), a particular Criterion can be expressed in multiple ways. QOC can help to identify when two modelling analyses are in fact describing the same Criterion in different ways, since they will have the same Assessment links to all Options (though the converse doesn't hold). Synergy of this sort between diverse modelling techniques represents an important kind of integrational relationship.

- *interdependency*: this can be said to occur when a usability analysis requires multiple modelling techniques in order to show what the central problems, and potential solutions, are. Interdependency can be detected in QOC design spaces by following through a series of design Questions, whose resolution is only possible when Qs, Os and Cs are contributed by multiple techniques.

Examples of these are given in [bm94].

3.3.2. QOC for modelling-integration: Weaknesses

Can be cryptic: Whilst QOC can provide a succinct summary, it also runs the risk of being over-cryptic. A summary or overview is generally only useful if the reader already knows something about the detail being summarised. However, as a means of communicating new material, whether rationale from one designer to another, or usability analyses from a modeller to a designer, QOC often needs to be supplemented, whether it be by verbal explanation, supporting text, or other kinds of design representation such as formal modelling, sketches or code.

As regards model-integration in particular, there is no doubt that a QOC representation of modelling output cannot express important elements of an analysis, in particular the formal basis which gives a modelling technique its power. This is of course only to be expected when the process and product of a formal analysis are re-expressed semi-formally, and purely as product input to a static design space.

We have therefore come to see QOC expressions of modelling as a graphical index into a larger corpus of material. On the one hand, stepping through a QOC design space is a structured way of navigating, and familiarising oneself with, the modelling as it relates to a given design problem; in turn, as one 'immerses' oneself in this design space, the QOC becomes a resource for understanding its structure and how modelling relates to it, as described in the previous section.

However, to reiterate the main point, for QOC to become such a cognitive resource, the user needs to gain familiarity with the details of the design space, which the QOC can only summarise.

Computational support required: The ECOM exercise emphasised the kinds of structure manipulation problems which must be faced when scaling up a QOC analysis. The space soon becomes multi-dimensional (not just 2-D), and the value of online facilities such as hypertext jumping, layering and the selective hiding of detail soon become clear.

After several years of using QOC both for design and modelling analysis, the requirements for a software tool to support QOC construction and management are now much clearer. Studies of QOC in use, and autobiographical reports of experiences with it have specified a range of requirements for, and in some cases explored the development of, prototype QOC tools [bs95a, bsh94, cc92, mm94, r92, s91]. Other research groups have developed tools to support different argumentation-based design rationale schemes, but have in the process addressed many issues pertinent to a QOC tool [abc91, cms91, cb88, l90].

All of these requirements are relevant to the use of QOC for modelling-integration and analysis. The highly collaborative nature of the ECOM exercise, given the many sources of input to the design space, emphasised in particular the potential of a tool capable of managing simultaneous construction of design spaces by geographically distributed users.

3.4 Modellers' comments on QOC

As a large scale QOC analysis was undertaken at a comparatively early stage in the project, it was possible to collect detailed personal comments from the modellers involved in the exercise. The following five sub-sections contain subjective and personal accounts of the perceived benefits and difficulties of using QOC to understand a large design space and and as a framework for recording the results or insight of modelling.

3.4.1 PUM

As PUM modellers, we were able to express some of our criticism of the prototype ECOM, and most of our suggestions for re-design, in terms of the QOC notation. In doing so, we were led to propose all three of Questions, Options, and Criteria, in roughly equal numbers. Of course, the *content* of our analysis, criticism, and discussion has not been translated into QOC, and in particular many of our criticisms, such as pointing out inconsistencies in the existing design, do not lend themselves to being expressed in QOC terms. Rather, the QOC serves to *index* or point to our discussion by capturing just the outcomes for design. From the QOC point of view, much of the PUM analysis plays the role of *argumentation* which supports some *justification,* i.e. an assessment of an Option against a Criterion. We did not explicitly link fragments of our text to specific O-C links, and were not sure how such an exercise would fare given the rather secondary status given to Criteria in the original QOC. Contrary to our initial intuition, it proved possible to relate many of our suggestions to the original QOC space — or at least to its main outline — even where we had to alter it or locally re-structure it. However, when we took a "broader view" of the design task, including doing a small-scale task analysis, we did not explore how the results from this re-analysis related to the original QOC nor what implications it would have for its re-structuring.

Using QOC effectively involves an element of craft skill, in the same way that user modelling does. The fact that one of us had some years' familiarity with QOC was noticeably helpful in making the translation. This leads us to suggest that modellers would have benefited in the exercise of expressing their modelling output in terms of QOC if they had received explicit training in QOC techniques (at least being given the existing heuristics about 'what make good QOC questions'). Better still, the 'translation' from modelling output into QOC should have been a real collaborative exercise in which the modelling expert and QOC expert jointly (face-to-face) worked out the translation.

3.4.2 CTA

CTA provides an assessment of the usability of an aspect of an interface (whether partially designed or well-formulated). Its 'topic' of analysis will, in QOC terms, be the cumulative product of a series of Questions and Options. Ideally, CTA would be used as the design space evolved, providing guidance about the adequacy of an Option (by providing cognitively appropriate Criteria and the respective O-C links) and providing support for the generation of new Options. Indeed, by guiding the designer towards a consideration of cognitively appropriate issues, CTA should help them judge the adequacy of the Questions that they are posing, as well as the Options. The effect of this will be to ensure that the topic of the CTA will be the 'proximal' Q-O pairing, rather than an earlier Q-O pairing somewhere 'higher-up' in the decision space.

In applying CTA to the existing ECOM design space, we were faced with the problem that the 'higher' Q-O links had not been evaluated or constructed in this way. When we started to model Options at the 'lower' end, we could not make assessments that were only relevant to the proximal Q-O link. We found that we were repeatedly driven to debate the 'higher' Q-O choices that had led to the subsequent Questions. This forced us to examine the 'higher' Q-O pairings first. Inevitably, as soon as our modelling disagreed with a chosen Option, the design space below that point became inappropriate, and if we proposed new Options, or posed new Questions, we started to venture into new design spaces that were not being addressed by the other modelling teams.

This was a particular problem in the ECOM design space, since it consisted of Questions that were primarily technology-driven, addressing how to implement connection types, control, and selection protocols, with little focus on why the user might want to be making connections, controlling access, or selecting objects. We found ourselves repeatedly suggesting novel Options early in the design space, and suggesting that the wrong Questions were being asked. Again, since we had not been involved in the evolution of the design space, we found it difficult to understand what many of the Questions and Options really meant, since their representations within the QOC notation were necessarily very brief, and the paths of development were not always obvious.

When it came to representing our modelling in QOC, we felt constrained by the absence of 'justification' from the notation. The primary content of our modelling is explanatory, providing reasons why a particular Option succeeds or fails on a given Criterion, with the explicit intention that these justifications help the designer to reason about their success or failure. Omitting this aspect of our modelling removes an essential part of its support for the designer. Perhaps because of this, we tried to produce Criteria that contained their own justifications, but which were seen as too detailed by the QOC practitioners. In contrast, we found that the existing Criteria were too vague and ill-specified to be able to relate our modelling directly to the existing O-C links.

3.4.3 FSM

Formal system modelling is derived from the use of formal specification techniques in software engineering where design is organised around a software lifecycle model. In this situation the input to the specification phase is usually a set of user requirements documents, and the purpose of the specification is then to define *what* the system should do to implement those requirements, without prescribing *how* the designer should realise the system. That is, specification usually operates on a single level of abstraction, which is then reified towards an implementation by replacing declarative constraints by procedural decisions. This seems at odds with the view of a design space provided by QOC, where the use of consequent questions and specific design options carry with them at least an apparent commitment to particular design solutions. It was for this reason that we experienced some difficulty in using the existing QOC model as a starting point for our analysis of ECOM. The approach that we ended up taking was to put the QOC aside, using the informal description and scenarios to define an abstract model of what a system such as ECOM is actually required to support. In short: we felt that the QOC model conflated issues about *what* ECOM was required to do with (sometimes highly specific) details about *how* the interface would support those tasks.

Once we had an abstract model of the ECOM interface, we could begin to consider some of the design issues by relating the components of the abstract model to the features discussed in the design space. As FSM is a model-based notation, it is most effective in dealing with *safety* or *stability* properties, for example that some relationship will always hold between the state and presentation of a system component. In the case of ECOM, we initially examined the relationship between the connections that a user was permitted to make, and the feedback about connections available on the screen.

Most of FSM's contribution to the design space was in the form of criteria such as the 'conformance' property that could, at least approximately, assess some of the options for feedback. A significant problem is that these criteria are coarse: either an option does or does not satisfy a formal property. For one issue, the way that the system allowed users to request connections, we were able to suggest some options by refining the abstract model into a collection of more 'concrete' interactors. Again, the most significant problem was the different backgrounds of the representation. A formal specification can be thought of as a model of a design space, implicitly capturing those implementations that will satisfy the requirements represented by the axioms of the specification.

In subsequent work we have returned to the ECOM system and developed a model of the access control problem and related this to analyses conducted by CTA, PUM and PAC modellers. The outcome of this analysis is successfully being represented within QOC. We suggest that these improved results have two reasons. Firstly, the models used in FSM have been simplified, and it is now much easier to use the approach on a tightly focused issue. Second, we are not trying to express our own insight in QOC terms, but rather are answering specific questions posed by other modelling approaches.

3.4.4 LIM

A first impression of QOC was that it represents what people do when they have to make a decision. The problem is that it does not provide a lot of support in the evaluation process as it is not a very structured notation. In the QOC approach the most important questions are those which are provided at the beginning as they determine the next questions and so on. This implies that it may be difficult to work on a design which has been started by other people because they may give different relevance to some aspects with respect to what we would have done. Another difficulty with QOC is that evaluation of an option with respect to criteria is often subjective. We found that in some cases a formal approach can overcome this type of problems: it is possible to formally specify different options and automatically verify if criteria expressed, for example, in temporal logics are satisfied or not. The problem in this solution is that options and criteria cannot always be specified formally. Ironically, the wide applicability of QOC derives at least partially from its ability to capture all forms of rationale (formal or otherwise).

When we had to develop a design rationale space for software interaction objects we found it more useful to have a list of predefined questions for each component rather than to have a cascade of questions. The first word in the set of predefined questions clearly indicates the type of question we are considering. For example we have questions such as: What information should be graphically represented? How should this information be rendered? When should rendering be performed? Where does it receive output data from?

3.4.5 PAC

PAC is a software architecture model. As software designers, we used QOC as a way to structure and justify our software design of a user interface. We therefore address a specific part of the design space: the software design. This has led us to introduce specific software-related criteria. These criteria are based on McCall's properties [m77], which are commonly used in the software design community. We extend the software engineering properties of McCall to usability aspects [acn92]. Bringing in these new criteria raised a terminology problem: the meaning of some criteria is context-dependent. For example, when user-centred, flexibility expresses the scope of freedom among a set of possible choices. In software engineering, flexibility is related to product revision. It denotes the capacity of the software to accept modifications. When ambiguity occurs, we suggest that the names of software engineering criteria should be prefixed with "SE" (standing for "Software Engineering").

QOC was used in the EuroCODE Amodeus assessment to promote a shared multi-modelling view of the design space. Since we then had easy access to the other modelling approaches' criteria, this led us to consider these criteria and to integrate them in our approach. Consequently, our final design space has been influenced by the usage of these non-software engineering criteria. The use of the QOC notation has also enforced the documentation of multiple options as possible software design solutions. This is usually not done so systematically in the software design process.

We also have two general suggestions to extend the QOC notation.

1. We would like to extend QOC into a QO*R*C notation. In QOC, options are linked to criteria only. We have found it useful to link options to requirements as well. For example, in the particular case of EuroCODE, option “proactive feedback” requires (or at least suggests) that the database for access control be active. If the data base at hand is passive, then honesty may not be achieved satisfactorily. If so, proactive feedback cannot be considered as a realistic design option.

2 We suggest that the QOC notation be extended with an encapsulation mechanism in the same way that procedures are used in programming languages. If two subtrees are strictly similar, then one could replace them by the same name and that name would denote a QOC subtree. The encapsulation facility would increase readability, would point out similar lines of reasoning, and would augment consistency across the user interface of the system being designed. They would eventually serve as reusable rationale applicable by other designers to other design problems (just like libraries are ready for use pieces of code that are reusable by different programmers in different contexts).

Aside from our role of software designers, we have derived some heuristics about classes of questions that user interface designers (as opposed to software designers) should formulate when using QOC. These are based on the quintilian model (From Marcus Fabius Quintilianus: Roman rhetorician of the first century A.D.):

- **What?** e.g., what information is relevant to the task? (this question is sometimes addressed in the QOC examples we have seen. Should be addressed more systematically)

- **Who?** e.g., who is concerned, who is in charge of the task? (this question is rarely addressed in the QOC examples we have seen. For example, the question "How are accessibility levels defined?" is actually an example of a "who" question. It should be rephrased accordingly

- **Where?** e.g., where, on the screen, to present information (this question is frequently addressed in the QOC examples we have seen)

- **When?** e.g., when to present information (this question is rarely addressed in the QOC examples we have seen)

- **How?** widely covered in most QOC examples we have seen

- **Why?** rationale and cause covered by QOC criteria.

To emphasise the "what, who, how, etc." nature of questions, we have suffixed question names with their rhetorical type.

4 Integration by Result II: Design Space Development (DSD/DR)

4.1 Design Space Development (DSD): A Description

DSD [br94], as developed in AMODEUS-2 plays a dual role in design support: to provide temporal and logical structure and traceability to design processes; and to integrate the contributions of different modelling techniques. It does this through its two notational frameworks: one for design space structure and one for design rationale representation. DSD provides structure to design spaces, in that it assumes that artefacts are complex products of an optimisation process which take into account such factors as cooperation, organisation, system, interface, task, user and user experience, and that artefact design aims to optimise user performance on the intended artefact. DSD has a simple notation for explicitly representing the reasoning which generates new design commitments. The notation serves to represent the *design problem*, the *option(s)* considered for its solution, the design *commitments* applied in attempting to solve the problem through selection between options, the *resolution* of the problem and the *justification* for the solution. DSD has been applied to design processes for the support of remote, collaborative work, to large scale design processes, such as a spoken-language dialogue project, to exploratory design processes, to generic technology design, to a commercial project, and to a redesign project with multiple design objectives.

DSD's role in all of the above projects has been in organising and driving the design process. More recently, however, attention has turned to investigating the second role DSD may play in design support, namely that of a multiple-modelling integrative platform for representing the output of multi-disciplinary modelling analyses. The example discussed here describes how DSD has been used as a multiple-modelling integrative platform for modelling done in the safety-critical domain of air traffic control.

4.2 Using DSD as an Integrational Platform

A series of modelling analyses had been carried out on a set of design problems offered to the AMODEUS-2 modellers by a design team involved in the development of the CERD (Computer Entry Readout Device), part of a larger air-traffic control system. Using DSD as a multiple-modelling representation for these analyses involved the construction of a set of standardised representations of the respective modelling in DSD's design rationale representation, in this case, various approaches to cognitive and system modelling.

Five parallel rationale representations were constructed, one for each of the AMODEUS-2 modelling approaches, FSM, PAC, IMAP, CTA and PUM. The intention was that each representation would evolve from DSD frame 1 (initial design space) towards DSD frame 2 (resulting design space) by representing the output of only one modelling approach. DSD frame 1 was in every case an identical, agreed representation of the CERD design space containing factual and evaluative commitments. The modelling groups had selectively addressed the same set of design issues, with the intention of comparing and contrasting modelling output. The standardised representations of the modelling then formed sub-spaces that provided input to a multiple-modelling space. This multiple-modelling space incorporated the set of issues modelled and the modelling *output*.

4.3 Methodological Problems

In the ECOM design space, modellers providing input to the space did so in terms of Questions, Options and Criteria, and as such exerted influence over the shape and scope of the space. The CERD exemplar

exercise on the other hand proved itself to be a much more highly defined space, with the questions, or issues addressed, having been predefined by an independent human factors designer. Operating within a pre-shaped design space caused a degree of imbalance between the demands placed upon a modelling technique to deliver, and the appropriateness of employing that particular technique to handle certain issues. The issues considered by the five modelling techniques ranged from the course-grained "are the CERD operations the most appropriate?" to much more fine-grained such as "is display-suckback a problem?" By way of illustration, the IMAP methodology, which exerts its power in the early stages of the design process by specifying minimal functionality necessary to fulfil the design goals, was not able to handle the finer-grained topics. Other modelling techniques were hampered by building upon incomplete domain information. Thus it would appear that it is necessary to allow modelling techniques (or rather the modellers themselves) the freedom to scope the design space themselves in an exercise of this nature. Just how this freedom to select an appropriate scope for each modelling technique affects the potential for cross-disciplinary integration is an open problem for each of the global integration techniques.

4.4 The relationship between modelling and DSD/DR

4.4.1 The handling of Formal System Modelling (FSM)

The use of DSD frames to capture the initial domain requirements and the subsequent commitments on the basis of modelling provided (in principle at least) a clear audit trail for following changes to the design over time. The caveat lies in the amount of detail that needs to be represented in the initial DSD problem statement. In principle, it contained all the information needed by the formal system modellers to produce their initial model of the system. In practice, it is not yet clear how well DSD structures the information compared to an informally written requirements document. Would it, for example, be feasible to separate specific types of design requirements into individual DSD frames, with an overall 'contextual' frame bringing the requirements together? This would be analogous to the way that system models are broken into structures (interactors), though clearly the two structures would not necessarily coincide as the decomposition is for different purposes.

Another positive aspect of the design rationale representation was the separation in the frames between design problem on the one hand, and options, resolution and justification on the other. From a modeller's viewpoint this supports separating the model-driven identification of a design problem from the partially craft-driven generation of design resolutions. One question that remains open is whether the justification section is necessarily informal prose, or whether, in certain cases, it would be useful to include or point to formal arguments or proof. This might be particularly relevant to the use of DSD within the design process for high-integrity systems.

4.4.2 PAC architecture modelling comments on DSD

The PAC software architecture model is concerned with software design. User interface design however, can be informed and driven by properties which can be expressed as usability commitments for the DSD frames. PAC can also provide input for user interface design by evaluating the complexity and implementation cost of a given design. This can be expressed in the 'General feasibility constraints' slot of a DSD frame. For example, the DSD frame regarding the CERD issue: "Can the message interface be improved ?" provided three PAC options. These design options were driven by properties. One option was a way of achieving browsability: waiting messages are not directly perceivable but the interface provides a way to make them perceivable to the user (scrollable list). The second one was directly motivated by the observability property (observability of the number of waiting messages). The third option was also driven by observability, but is a degraded version of the second one: the interface only shows that there is a high priority message waiting. As with QOC, DSD could be used to assess issues related to software design and

architecture. A given software architecture design issue could be considered in a DSD frame and possible options could be assessed with regard to constraints. This would be useful for software designers in helping them make design decisions. Software engineering properties (e.g., reusability, maintainability) would serve as commitments. An interesting benefit from DSD is that it clearly points out the hardware constraints in the design domain frame (i.e., first DSD frame), and software constraints, e.g., that a particular tool should be used. These constraints are essential for PAC modelling. In the case where DSD would be used for software design, the 'design process type' slot might mention the software design cycle to be used (e.g., V-cycle). The 'designer preferences' slot could mention particular constraints related to software design (e.g. an object-oriented approach, or a given development environment).

4.4.3 CTAs comments on the DSD representation

The process by which the CTA report on the CERD was produced and then represented made the modellers feel uneasy. The report had initially been produced with the intention of showing how modelling could be used to influence redesign. In particular, in the original modelling exercise, they had been asked by the human factors designer to identify ‘design issues’ to indicate areas of the design space that it was felt needed further investigation (but not to explore them), and to produce illustrative ‘design options’ showing how the CERD could be redesigned to avoid any usability problems that they identified, a task that the modellers felt properly belonged to the designers.

In consequence the report was neither directly representative of their technique, nor was it directed towards providing the sort of information required by DSD. There was a need for the design rationale frames to be filled by interpreting the material that had been provided by the modellers for another purpose, but since the DSD team did not want to misrepresent the modelling approach by rewriting the material, the result was a selective ‘cut and paste’, with material intended for one role ending up in slots designed for different roles. An obvious drawback is the need in DSD’s rationale representations to provide ‘options’ for design, a ‘resolution’ between them, and a ‘recommendation’ for implementation, all of which had to be filled by identical ‘Design Options’ from the modelling report, which was not actually ‘part’ of the modelling.

The CTA modellers had attempted to meet the communicative requirement of the initial report by using typographical layout to relate points of the modelling with a plain text commentary, and by using a narrative flow to show how the identification of cognitive usability issues could draw the designers’ attention to aspects of the design that could be altered advantageously. By re-representing the text of the report within DSD’s design rationale frames the rhetorical structure of the report was destroyed, without the tabular format of the frames being able to add anything. If the material had been more closely matched to the design rationale notation, the added-value of the representation might have been more apparent.

4.4.4 Programmable User Modelling (PUM)

The PUM modelling group felt themselves to be one step removed from the expression of the modelling analysis in DSD. Like the CTA modellers there was difficulty experienced in comprehending the categorisation of modelling content in the frames. For example, several things that they might have classed as a "commitment" or a "problem" were categorised as "option"s. As a general means of presenting modelling analyses, it was felt that the exercise of restructuring the information into design rationale frames served to obscure, rather than clarify.

4.4.5 The Information Mapping Analysis (IMAP)

The way the IMAP analysis was represented in DSD’s design rationale frames was seen to be problematic in the following ways. Firstly, IMAP is not of an evaluative nature, rather it is a design support tool,

usefully employing DSD as a requirements gathering tool. The CERD analysis was nonetheless conceived as an evaluative study of an existing interface, focused around a number of potentially problematic issues. Unfortunately, IMAP cannot address lower-level interface design issues, explaining why many issues in the frames remain unresolved.

The remarks described above, however, do not stem from the nature of DSD, but from the specific application of these frames in the CERD exercise. Nevertheless, there were some problems which did originate from the nature of the DSD method. IMAP starts from information and functionality that needs to be represented in an artefact, whereas DSD aims to solve certain design problems or questions by presenting a number of options and then deciding between these options. The starting point for the two techniques is fundamentally different.

However, the representation of the IMAP analysis in DSD's design rationale frames was held to be very helpful in a number of ways. The modellers themselves learned some lessons about the IMAP methodology. Sometimes a resolution to a 'problem' was justified by a number of IMAP rules. In other cases, however, an explicit justification for a certain decision was unclear or absent. The use of a DSD design rationale frame forced the modellers to consider the justification of each design recommendation made, and to clearly specify each justification. In future IMAP studies, the modellers realised that their justifications need to be clearly specified, however obvious they might seem.

4.5 The Multiple-Modelling Space

After obtaining standardised representations of the modelling in the design rationale semi-formal notation, the *output* of the single-model frames was carried forward to a set of *multiple-modelling spaces*. For each issue, the modelling groups had produced a series of options, which were now evident from the previous design rationale representations. Some examples are given below by way of illustration. The full set of spaces, which runs to 80 pages of single-model representations, and 16 pages of multiple-model representations can be found in Ramsay (1995).

For the following issue: "The same display can be produced for different reasons. Is this a problem?" two of the modelling groups had input (table 4.1 below). Option one was the current CERD state, option two was put forward by the cognitive task modellers, and option three by the formal system modellers.

1	If a flight diverts, the "gap" left will be removed on using Tidy. This gives exactly the same display as when another flight moves out of the rest menu and the gap was filled **(CERD)**
2	Smooth scroll the 'onscreen' flights to new slots in SAS one at a time following a Tidy instead of just displaying new SAS **(CTA)**
3	Replace the single NAS-update action defined in display-SAS with explicit actions to represent flights landing, diverting, or changes to the SAS brought about by the passing of time **(FSM)**

Table 4.1: Excerpt from design rationale representation

Inspection of the above reveals that the cognitive and system modelling arrived at an identification of the same problem, and essentially proposed a similar repair of the problem: the cognitive modellers by recommending explicit directional feedback, and the formalists by explicitness. The formalists did not go as far as to propose specific design *solutions*, rather the nature of their option is generic. The cognitive modellers' solution is, however, very concrete, since their modelling identified a perceptual problem that needed to be remedied.

Are CERD's current operations the best ones to cover the tasks which need to be performed? Could they be better supported by another set of operations?

1	Assign, Reposition, Resequence, Swap, Tidy **(CERD)**
2	Use a graphical indication of wake size, limit information to one line per key. A 'quick' operation to modify the position of one flight within the SAS, and a 'multiple' operation to change the positions of several at once. Distinguish single and multiple names, and group together. The Assign operation should be nearby, but not form 'part' of the same group **(CTA)**
3	Create four static graphic interactors which clearly indicate alternative types of action: an Assign interactor, a Reposition interactor, a Resequence interactor and a Swap interactor, all labelled as such, with interaction through pointing gesture or touch. **(IMAP)**
4	Two-handed interaction for Swap operation **(PAC)**
5	Have distinct actions corresponding to the different conceptual operations. The warning message appears as soon as a code has been selected, but with the 'bktrk' key still enabled, as well as 'confirm' and 'cancel'. Repositioning errors could be avoided by introducing active areas representing "between" flights. A go-back button would return to the screen display showing before the last "goto" or series of scrolling operations should be considered **(PUM)**

Table 4.2: Excerpt from design rationale representation

The above issue prompted four model-based outputs. CTA follows the phrasing of the question and talks in terms of *operations,* IMAP in terms of *interactors,* PAC in terms of *interaction mechanism,* and PUM in terms of *actions*. This is possibly prompted by the open nature of the issue, as well as the modelling technique itself. IMAP and PUM seem to be close in spirit in so much as they both advocate a clarity of correspondence between operations and actions, albeit at different levels of abstraction.

A potential advantage identified by the modellers stemmed from the layered nature of the representation. Where QOC obscures the modelling by presenting only its results (as O-C assessments), it seems possible to have one level of DSD design rationale frames including modelling argumentation in 'justification' and 'comment' slots, while providing more design oriented material in a second layer of frames. While the designers may not need to read the modelling, it would be available for them should they want to. This would be helped by a finer breakdown of options and a clearer cross-referencing of frames, and would also inevitably be more apparent in a second, more detailed level of sequenced frames, where the need to backtrack or to trade-off conflicting options was more obvious.

To have proximal spatial location of explicit accounts of modelling recommendations appears to assist the reader of the space in a compare and contrast exercise, allowing for example, the identification of synergies between modelling outputs, and the justification for commitments and options has its place in the space, albeit at another, more explicit level. With other notations such as QOC, it had been reported that such justification was not easily expressed.

4.6 Modelling, craft skill and design

The emphasis of the integrational exercise described, was upon the representation of multi-disciplinary *modelling*. Importing the analyses' output and its justificational content proved problematic in so far as modelling was merged with elements of craft skill. Taking the example of PAC architecture modelling, it is necessary to understand that any proposal emerging from the analysis is craft skill. The dialectic between

modelling and craft skill involves delving into the architecture and enquiring as to whether a particular course of action will be costly to implement or not. Thus it is necessarily an iterative process, with less obvious direct traces between the architecture representation and its output in terms of design options. Whether there is craft skill mixed with modelling is less of an issue than the ability of the integrational representation to support the access of justification. Techniques which provide an overview of the whole design space, as well as supporting the presentation of detailed regions of argumentation, are clearly required if the information represented is to be manageable. The fish-eye lens notion (the Bellcore SuperBook project) which supports optional opening up of areas may show the appropriate way forward.

Perhaps the appropriate perspective with which to conclude this discourse on DSD, comes from a consideration of representational *intention*. DSD has evolved as a design support tool with the primary purpose of representing *design,* rather than modelling. As one modeller pointed out, modelling might provide some of the content in a DSD design rationale frame, but will not be able to provide the option selection that is necessary for design. Thus, any application of DSD to a pure modelling exercise should recognise this reduced ambition in its use.

5 Integration by Process

The previous section has focused on two approaches which achieve integration by providing structures around which the results of modelling (and indeed other design inputs) can be structured. When the results of modelling are expressed in the QOC or DSD design space, their relationships to the overall solution and to each other are highlighted by the structures provided. However, because of their different emphases, the precise kinds of relationships which are highlighted differ. To over-simplify a little, QOC tends to emphasise contrasts and optionality within the design space and DSD emphasises the ways in which the modelling results support the solution currently favoured.

If we consider the practicalities of integrating modelling in a real design setting, it is clear that the details of how the design process is organised will be important in determining how, and how well, it actually works. This is acknowledged in both QOC and DSD in that some parts of the work emphasise the processes by which the design space is produced. For example DSD has produced a manual for providing guidance in its use [rb95] and indeed has process elements built into the structure via the evolution of DSD frames. Much of the QOC work has emphasised the use of QOC from within the design process (e.g. mm94]. An account of the development of a QOC design space is given in MacLean et al [mbs93].

Although there is a strong emphasis on the design process in the broader QOC, DSD and related work, it has not been specifically developed for the purposes of integrating modelling *per se*. There are clearly issues in the ways in which the structures these approaches provide might support (or inhibit) the mediation of modelling approaches from a process perspective. However, these have not been explored in any detail as yet. In the remainder of this section we describe two approaches that support integration through thedynamics of the design setting. The first of these is Argumentative Design (aRD), driven from within an Action Design perspective which regards process as the primary organising focus around which the view points of different stakeholders needs to be considered and represented. Like QOC and DSD, aRD is an existing approach that has application beyond integration of multi-disciplinary modelling. The second proces-based technique is a specific development within Amodeus called ‘co-modelling’. This explores integration by using the dynamics of the design setting to allow interchanges among modelling approaches to be expressed directly in terms of the modelling approaches themselves, rather than mediated through a

"third party" representation. The next section will examine the implications that the findings of this and the preceeding sections have for the role mediating expressions might play both in terms of the short term dynamics, and in terms of structuring and preserving the results of the modelling in the longer term.

5.1 Argumentative Design

Argumentative Design, or aRD [s94, s95], is a method and corresponding notation devised at Linköping University to support early phases of participatory systems development. It has been developed as one of the tools in the Action Design methodology, within which it is used to represent design rationale and support resolution of design options in a way that supports participation of diverse groups and viewpoints. In aRD, the QOC concepts of Question, Options and Criteria are re-expressed as Change Needs, Measures and Goals and the underlying argumentation is brought to the fore in the representation producing a more verbose, but more detailed, representation of the design information which is co-produced by stakeholders within the design process [s94]. The terminology and structure is motivated by a concern with having a good match to the needs of the specific process perspective adopted.

The aRD method consists of a number of steps, summarised below.

- Change need analysis: Change needs derived from an existing design, or a goal/problem analysis of an organisation are documented and analysed.
- Generating measures: Each change need generates one or more measures to address that need.
- Goal analysis: Change measures are related to organisational goals of the organisation, with the aim of identifying goals that support or oppose one or more measures. The relationship is documented in an aRD table (discussed below). Here conflicting goals can exist in the same aRD table.
- Consequences: Once goals are related to measures, the consequences of a measures can be generated. That is, the impact of the measure on the organisation and its goals is identified and documented.
- Revision: Measures and consequences that have been documented should be reviewed (from step 2), possibly leading to the identification of further measures or consequences..
- Decision: A change decision is made on basis of what has been discussed and documented with aRD. This decision may involve *not* carrying through with any measure, that further analysis should be made of one of the measures, or that one of the measures will be carried through design and implementation.

The notation used within aRD to documented goals, measures and consequences is a significantly modified version of QOC. A typical aRD structure is shown below; this is based on a need related to access control with the ECOM exemplar. As with QOC, a graphical notation is also supported, but is not shown here.

Need: Determine recipient availability level

Change measure	Goal G3: Privacy	Goal G5: Co-awareness
Technical link capacity	– Potentially disturbing - Breaching confidentiality	+ Maximal awareness level
Recipient specifies	+ Control	+ Respecting social needs - Less than max awareness

The table is a hypothetical example from the development of a media-space system. Briefly, it documents a piece of argumentation concerning availability levels: are other users on the system available for email, glance, video conference, or perhaps not available at all. The issue here is how to generate the information that the system presents to a current user. Issues like this are called *needs* in aRD, and correspond closely to the questions of QOC. Two alternatives (or options in QOC terms) have been identified: the system can monitor the connection status and always present the "richest" availability level possible, or else the intended recipient of the connection can specify how available he or she likes to be to other users. In aRD, the alternatives are called *change measures*. The alternatives are assessed against *goals* that were formulated and agreed upon earlier in the process. These goals serve a role that is similar to criteria in a QOC structure. The main difference between aRD and QOC is found in the body cells of the table: instead of a simple positive or negative assessment, the aRD structure documents the positive and negative *consequences* of the change measures in relation to the different goals.

5.2 Collational Co-modelling

Collational co-modelling is the newest of the integrational approaches within Amodeus, although an earlier version was explored in the multi-disciplinary analysis of Shared Undo [ya94]. It differs from DSA/QOC and DSD in that it does not require the use of a new notation with its own set of commitments. Instead, co-modelling involves (a) direct communication between the modellers, and (b) iteration of the modelling, together with (c) an attempt by a "collator" to pull together a shared, multi-disciplinary analysis of the issues under discussion. The iteration requires modellers to re-visit their own analyses in the light of others', to build on others', and to contribute to others' both critically and constructively. The role of the collator includes occasionally probing particular models (to aid both clarity and relevance) and occasionally drawing the attention of one modelling group to a particular aspect of another group's analysis, in order to help catalyse the co-modelling. Co-modelling begins with a design issue (on which the modellers will normally have done prior, preparatory analysis), although further ones arise during the process. During the session, modellers pick up points from each other's analysis, raise further issues and problems, possibly engage in additional modelling, and propose (partial) solutions. In doing all this, they work towards convergence on an agreed, shared analysis of the original issue and possibly of others that have emerged. Integration, as such, occurs when an issue or solution from one modelling technique generates a new issue or solution for another modelling technique, and especially as the common analysis gets constructed. This stands in contrast with "imposed" design frameworks such as DSA/QOC and DSD, in which contributions from different modellers come together only later, after the modelling has been done.

The process just described is best regarded as an *ideal* form of collational co-modelling, which in practical circumstances can probably never quite be attained. So far we have conducted two experiments with co-modelling. In one of them the discussion was conducted over email, so that although the process took the form described above, it was slower paced and lacked some of its interpersonal dynamics. The other experiment was carried out face-to-face, retaining the rapid dynamics, but the short sessions (60-90 minutes) did not permit the detailed construction of the shared analysis. Instead, the session was video- and audio-recorded, and it was left to the collator (helped by others) to reconstruct the story later. In other circumstances one can imagine that the modellers will tackle issues which they have not yet analysed. Such "on the fly" modelling will affect the character and depth of the interchange, but the process remains recognisably that of co-modelling. The technique of collational co-modelling is still being evolved, and further variations will doubtless emerge.

5.2.1 An Example: the CERD

Co-modelling has so far been applied only to a handful of design issues from the CERD exemplar. The first co-modelling exercise was carried out by email, and this provides a good overview of the general approach.

As mentioned in Section 2.2, the CERD is an interface used by Air Traffic Control Officers (ATCOs) to inspect and modify the approach sequence of aircraft into airports. One function of the CERD is to display messages that inform the ATCO of major events in the support system or convey information about incoming flights; the latter may cause to the ATCO to re-order the approach sequence. Messages sent to the CERD are held in a priority queue (there are 3 priority levels), with only the 'front' message being available. This is removed from the queue and discarded when the ATCO touches the message area of the display, and the next message in the queue then appears. The initial focus of the co-modelling effort was a design question posed by the CERD developers:

CERD Design Issue No. 7 (from end pages of bs94a):

> *"Should messages other than the current one be accessible? And if so, how?"*

The initial FSM and PAC analyses of message handling raised a potential design issue: it seemed that messages, in some circumstances, could be lost. This *claim* was supported informally by an *illustrative scenario*, taken here from [ybd+94]:

> A medium-priority message is currently displayed. The user has finished dealing with it, and reaches across to touch the message area in order to delete it. However, at the same time, the system is in the process of sending a high-priority message to displace the current, medium-priority one. The message changes just milliseconds before the user touches the panel, so the new message is deleted. The user is unaware that the new message even existed.

Once the potential for the problem was understood, other modellers identified scenarios that lead to similar problems, while the system models (in this case) identified *consequences* of the unexpected behaviour. The *synthesis* of the analysis is presented in as a series of arguments or comments, linked to the *relevant modelling* by extracts or pointers in the appendix. The co-modelling analysis concludes with *recommendations* for potential design changes that would address the identified problem.

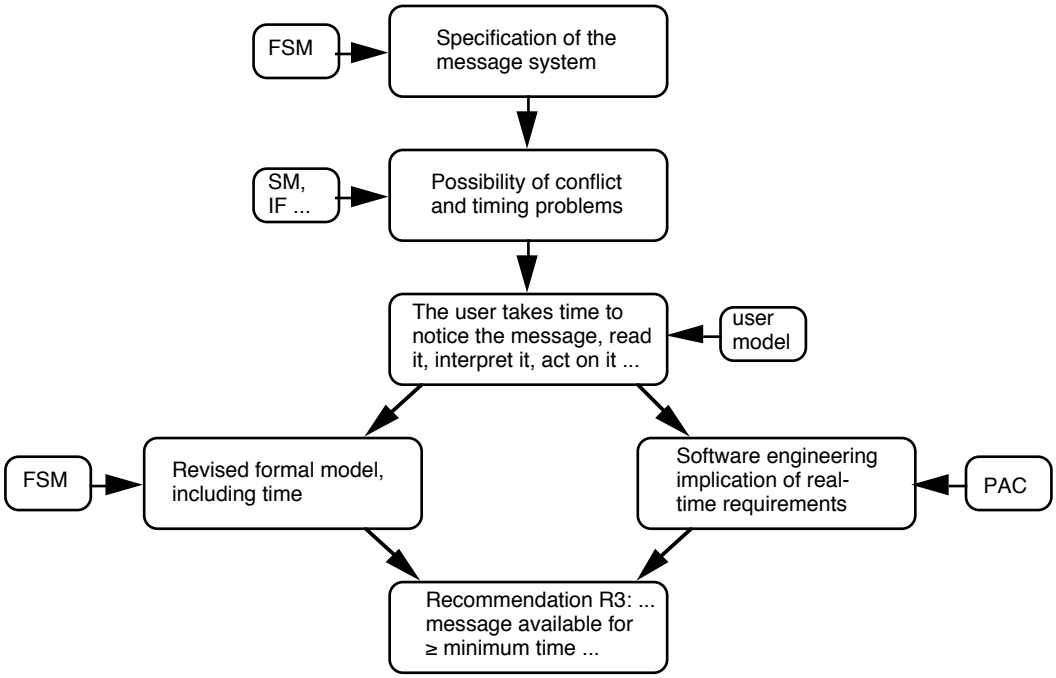

Figure 5.1: One strand of the co-modelling analysis.

One strand of the analysis is shown in Figure 5.1, and will serve to give a flavour of the co-modelling [ybd+94]. This part of the story begins with the formal description of the message system provided by FSM. Both the FSM and the IF notice that because the CERD can receive messages from two independent sources — the user and the NAS — problems of timing and conflict can arise if messages arrive from both sources at about the same time. The user models pick up that point, and say "Now that you're talking about time, do you realise that it *takes time* for the user to perform actions with regard to CERD?". Both FSM and PAC in turn respond to the user modellers. FSM is able to revise and extend the original formal model, to reflect the fact that user reactions take time. PAC points out that any requirements placed on the real-time response of the system have implications for the software engineering of the design. The modellers agree that one way forward would be to consider re-designing the message interface so that each new message is displayed for some minimum time before it can be deleted, though the user modellers remind everyone that this by itself cannot guarantee that each message gets read.

5.2.2 Co-modelling as an Integrational Framework

Collational co-modelling is obviously an integrational enterprise. Its input takes the form of diverse activities and materials from the modellers, and its output is an agreed multi-disciplinary analysis of the given design issue. It can also be seen as providing a possible answer to the narrower, more technical, question of how to relate specific system- and user-modelling approaches. In this, it occupies an intermediate position between the loosely-coupled approach of devising usability-oriented "properties", and the far more tightly-coupled approach of syndetic modelling (described in Section 7.3).

Contrast with other global techniques

The main difference between collational co-modelling and the integrational frameworks based on Design Rationale (DSA/QOC and DSD) lies in there being no fixed, *a priori* notation interposed as a mediating representation between the modellers. Instead, the modellers communicate directly with each other. Modellers enter the co-modelling session with some initial ideas and tentative structures based (ideally) upon prior analysis of the relevant issues, but without a commitment to any one particular notation to be used as the *lingua franca* for discussion. The language of interchange is instead driven by the needs of the co-modelling. The modellers will exchange natural language, diagrams, sketches, fragments of analyses, etc. — whatever is required to communicate to each other and clarify the relevant ideas and aspects of the modelling.

An important difference for the modellers is therefore the concrete and detailed form the discussion can take when it does not have to conform to the demands of an externally imposed notation. DSA/QOC, for example, abstracts each modelling contribution to the point of simply wanting to know, "Is it a Question, an Option, or a Criterion? And where does it fit into the QOC structure?". QOC therefore provides only a "low bandwidth" for mediating between different modelling approaches. In co-modelling, by contrast, the interchanges between the modellers preserve the full content of the modelling: they are not forced to strip it away. Co-modelling thus provides a "high bandwidth" route for modelling techniques to cooperate with each other in order to produce an integrated analysis.

Different realisations of co-modelling

To date, we have carried out co-modelling exercises both by email and face-to-face (or "live"). As already mentioned, either of these only approximates the "ideal" (and probably unattainable) version of co-modelling. Presumably still other arrangements are possible. Obviously the decision between them will depend upon the detailed circumstances of each case, and the choice requires making a number of trade-offs between conflicting factors.

The live co-modelling has a number of obvious advantages, including the faster, dynamic interchange between modellers (which is also highly motivating), and the fact that the face-to-face interpersonal context makes it easier to interpret — and if necessary to query or challenge — the contributions of other modellers.

But carrying out the co-modelling by email also has a number of advantages as compared to the live situation. The slower pace and the discretionary timing provide the individual modeller with more time to reflect on others' contributions and to perform further analysis, including actual modelling. In addition, email offers the big advantage for the collator that the results of the discussion and analysis are all available in the email messages, whereas from the live co-modelling they have to be pieced together after the event from a mixture of audio and video recordings, participants' notes, flipcharts, and fading memories.

Requirements on the models

One can pose the following important question about the relation between collational co-modelling and the modelling techniques we have employed: Is there anything special about the Amodeus modelling techniques that makes them particularly well suited to co-modelling, or can co-modelling be done with *any* techniques for modelling users and systems? Arguments can be brought forward for either view.

On the one hand, one can focus on the *process* aspects of co-modelling, and argue that the procedure can be carried out for any modelling techniques. According to this view, what is needed for co-modelling is that there be different kinds of user and system modellers, that they agree to tackle the same issue, that they communicate directly with one another and react to each other, and so forth. Nothing in this description of the process relies upon any special properties pertaining to the particular Amodeus techniques: PAC, CTA, and so on. So the conclusion should be that there is nothing special about the Amodeus techniques. Other user and system modelling approaches could be substituted for the ones we have, and co-modelling could still proceed in the way we have explored.

On the other side, one can argue that special requirements are indeed needed of modelling techniques for them to partake successfully in co-modelling. According to this alternative view, in order to take part in the co-modelling the techniques need to have mutually adapted to each other over time, so that they can analyse the interface in terms of shared concepts, and so that what one modelling method produces as output can be used by another as input. It should be noted that these inputs and outputs are not necessarily the official "data" and "results" of the methods. A user modeller may well take inspiration or guidance (rather than data) from a detail or aspect (rather than official results) of the system modelling, and vice versa. It is also important that the modellers themselves have a moderate degree of familiarity with each others' techniques, so that (at least) the necessary translations of terminology can be made. All of this suggests that for the co-modelling to succeed, the different modelling techniques must have coexisted for a period of time, possibly measured in years. That in turn implies that, while there may be nothing "magic" about the particular techniques investigated within Amodeus, simply taking a collection of other models and having the modellers go through the motions of co-modelling will not lead to a successful outcome.

The arguments each way seem, perhaps, equally persuasive (and inconclusive). Without experience of co-modelling done with modelling techniques other than those explored in Amodeus, the question will have to be left open, to be answered in the future.

6 Global Integration in Review

The global techniques are each able to integrate insights from a diverse collection of modelling approaches. This generality is possible because the representations used by the integration approaches are separate from those used by the modellers. Instead of each modellers having to understand the language and representation of the other modellers to share insight, all that is needed is for some party to be able to translate the output of the modelling into the ''shared'' representation. In the case of QOC and DSD, this representation is a structured 'mediating' notation and consitututes the 'result' of integration in itself. For co-modelling, there is no fixed notation. Instead, the shared understanding of the evolving problem space is expressed informally - the mediating representation is whatever the modellers choose to communicate their insight and analyses, for example graphs, diagrams, natural language or even parts of modelling expressions where there is shared understanding of a particular technique. The fourth technique mentioned so far, aRD, does contain a notation for recording design measures and argumentation, but is classed as a process based notation because of its emphasis on the use of the notation to support participation and iteration rather than as a reporting means. Ideally, transfer from modelling expressions into a 'shared' representation should be done by the modellers themselves to reduce the potential for mis-interpretation, but the process may be assisted or carried out by a 'scribe' or 'co-ordinator' experienced in the integration technique and its representation.

It would be wrong to assume that global integration is completely independent from the modelling techniques. Both the pace at which modelling can be carried out, and the interdependencies between models in the form of assumptions or questions, may constrain integration. For example, a formal system model of a particular issue may require a substantial amount of thought and rewriting before clear insight into the issue can be gained. In contrast, user modellers may be able to generate insight rapidly, based on an approximate model of the context. This may hinder (but not prevent) the use of process-based integration. Similarly, the kind of detailed information that user models need in order to assess *how* information should be presented may not be available at the point in the design process where system models might be most appropriate, when considering what parts of the state should be perceivable. Put simply, global integration techniques have an operating assumption that the modelling over which they integrate can sensibly be carried out at roughly similar points in the design process.

It is clear that when we consider the local *dynamics* of a design setting that it is most satisfactory when there is minimal intrusion from translating the elements of the discussion away from their own terms; this seems to be supported by the process-based techniques described in Section 5. However, if we take a broader view over the design lifecycle we have to balance this against the maintenance of a design history and the provision of an account against which further design input which was not part of particular local discussions can be located. The goals of structured design representations such as QOC and DSD (and to an extent aRD) emphasise this last point. If we decide to record a structured account of design rationale, we may have to decide when and how to move from a dynamic unmediated development of design thinking to the chosen structure. In the context of QOC, this issue is discussed in more detail e.g. see [myb+91]. In the case of McKerlie and MacLean [mm94] had a particular person in the design team produce QOC summaries after a meeting, and used these to support future meetings and discussions. With this approach only one member of the team had to be able to produce QOC, but all had to be able to make sense of the resulting design space. For the particular project in which that approach was used it worked very satisfactorally, but this is almost certainly not the only way to organise the role of mediating expressions within the design process and other possibilities need to be explored in more detail. Further understanding of how different result-based techniques can usefully support or inter-operate with process-oriented integration approaches is a topic for future work.

The strength of global integration - that its independence from modelling gives it general applicability - is also its key weakness. By separating design insight from modelling theory it becomes more difficult to then understand *why* some design issue is problematic, or *how* the design could be improved. Although the integrated model can contain pointers back into the original modelling, this is not so helpful where the design issue or its resolution relies on an emergent or synergic interplay between different techniques.

7 Local Integration

Over the duration of the Amodeus project it became clear that there was a need to supplement the global integration techniques described in Sections 3 to 5 with new approaches that are more sensitive to the results, representation and use of the underlying modelling. One reason, the need to understand the interplay between different theories, has already been mentioned; this for example is one rationale for the syndetic approach described in Section 7.3. Other forms of semantic integration are driven by the need to develop a common understanding of an issue, such as a property of the interaction. This may involve accommodating perspectives from different modelling techniques. In this section we cover four approaches developed within Amodeus that may be said to involve a semantic level of integration. These approaches are: Interaction Framework, CARE, Syndetic modelling, and Semiotics.

Interaction Framework serves an integrating role by putting selected results (such as requirements and properties of a given system) from user and system modelling at appropriate points in a space described in terms of interaction 'trajectories' built from events representing communication between agents. It does not require that system- and user-modelling provide input on the same issue, but by putting design issues into a shared model, it encourages the development of a cross-disciplinary model in a manner similar to that of co-modelling.

Syndetic modelling is also a modelling technique. It models how system and user are likely to interact "locally" in a design space. In contrast to IF, it is concerned directly with modelling the behaviour of the individual agents to the interaction, and is based on particular modelling techniques, namely FSM and ICS. Relevant aspects of ICS are expressed in terms of FSM axioms, to bring the theories into direct 'contact'.

The other two approaches are concerned with understanding how system and user modelling can contribute to the understanding of a particular issue. CARE gives system and user perspectives on properties of interactive systems. Semiotics attaches modelling analyses (from user and system viewpoints) to specified signs. Apart from the syndetic approach, none of the techniques is committed to any particular modelling techniques (and it should be noted that FSM is loose in that it represents a way of using formal methods to model interactive systems, without commitment to a particular notation).

7.1 Interaction Framework

The Interaction Framework (IF) supports an approach to designing and analysing interactive systems by defining properties of interactions in terms which take account all the agents involved. These properties are expressed in abstract — sometimes formal — terms. It offers an approach which is not biased by computer system or user concerns, but treats both as equally important. Many principles or properties of interactive behaviour cannot be understood from just one perspective. Two examples of general properties will show what we mean.

(1) An interactive system may satisfy the property of having "high potential", such that there are many ways to achieve different tasks. An interactive system has this property if the computer systems involved support this high optionality and the users can exploit the optionality.

(2) A system satisfies the property of "predictability" if it is possible for users to perform tasks with confidence that the computer system is going to behave satisfactorily. Predictablity requires support from both sides of the modelling divide. Explicit visibility of components of the state of the computer system will be required, but it is also necessary to ensure that those components of the state are interpreted appropriately by the user.

IF is compatible with current research on domain modelling in requirements engineering. Its purpose is to build representations of interactive systems that are neutral with respect to computer systems and human agents, so as to ease the problem of mapping function to computer system or human user. Interaction requirements are expressed in terms of agents, events and the goals that the system is to perform. Interaction Framework is intended to provide a means of expressing these properties precisely. User and system modelling approaches can then be used to measure and explore the consequences of the interaction requirements on software system components or user components. It should then be possible to make appropriate trade-offs. For example, a lack of predictability might result in a computer system change or a requirement to train users or both.

IF focuses attention on the design of the interaction by considering the possible patterns of interaction, and properties of those interactions. Designing with IF involves considering requirements which the interactive system should satisfy, and assessing how well a particular design satisfies them.

An interactive system is described in terms of:

- the agents (users and computer system based) involved in the interaction;
- the events which the interaction consists of;
- pre- and post- conditions on the state of the interactive system (i.e. conditions which must be true for the interaction to take place, and conditions which will be true if the outcome of the interaction is 'successful'; these include conditions relating to aspects of the user — such as user knowledge — as well as aspects of the computer system state).

Abstract interactional requirements are expressed in terms of predicates on the state of the interactive system or properties of event trajectories. For example, one requirement on the interaction may be that the system must be "reset" to a state such that further interactions are possible whatever the outcome of the current one. It should be noted that this is a requirement on the total system; the computer system should be reset to a state from which further interactions are possible, and the user should also be aware of the current state of the interaction — e.g. if the current goal has been abandoned for any reason, such as computer system failure.

An approach based on interactions can serve several important purposes; those most relevant to the role of IF as an integrative framework are that it provides "hooks" into system- and user-modelling.

The main way in which IF provides "hooks" into system modelling is through the refinement of the specification and by asking pertinent questions of system modellers (e.g. "if we make these assumptions about user behaviour, and we have this interactional requirement, what implications does this have for the design of the computer system?")

Similarly, it provides "hooks" into user modelling by asking similar questions in reverse; e.g. "if the computer system component is implemented in this way, how is it likely to be perceived and responded to by the user? How well does this satisfy the identified interactional requirements? Is there an alternative way of designing the computer system component such that it has properties that are likely to better satisfy the interactional requirements?"

We are not committed to just one approach to modelling aspects of user behaviour, or the computer system, but can recruit support from a range of possible techniques, selecting the most appropriate one for addressing the current issue.

Unlike QOC or DSD, I.F. is not a framework that can accommodate all design considerations, and it does not serve as a "decision capture" notation. It is a locally integrative modelling technique that focuses on user- and system-considerations that influence the properties of the interaction. It makes demands of, and takes input from, both system and user modelling, and therefore serves a very different integrational role.

7.2 CARE

The approach of integrating system- and user-modelling by jointly analysing properties is exemplified in the analysis of the CARE properties [cns+95]. The CARE properties characterise the usage of multi-modal interaction: the Complementarity, Assignment, Redundancy, and Equivalence that may occur between the interaction techniques available in a multi-modal user interface. We have provided a formal definition of the CARE properties from a system point of view : the system CARE properties; these definitions provide a formal framework for reasoning about the features of multi-modal systems. The CARE properties of the computer system have a counterpart in corresponding properties of the user: the CARE-like properties. As with the system CARE properties, the user properties are concerned with the choice between different modalities for communicating with the computer. We use the notion of compatibility to show how the system CARE properties interact with user CARE-like properties in the design of a system. Questions raised by user modelling can serve as a useful list of concerns during design. This approach to integration does not capture all design considerations, but focuses on particular properties of system and user behaviour, and seeks to understand their interrelationship. The CARE properties happen to be relevant to the design of computer systems that accept multi-modal input. However, as an integrational technique this exemplifies a general approach that could be applied to other properties that have been viewed initially from just one perspective.

7.3 Syndetic Modelling

Syndetic models [dbd+95, d95] are specifications that span the state and behaviour of both user and system agents. As such they sit between Interaction Framework and software architectural models such as the PAC agent approach. A syndetic model supports reasoning, at various levels of detail, about the relationship between system features and the cognitive structures and abilities needed to perceive and effect change in the system. It achieves this by expressing both a system model and a model of cognition (specifically, the ICS model underlying the CTA approach to user modelling) as axiomatic theories: once both models are in a common language they can be combined by the simple expedient of creating a third model that includes both – the syndetic model. In this model additional observables and axioms are used to define how features of the system - for example perceivable states and actions - are mapped onto the cognitive resources of the user. This approach allows direct (and formal) comparison between the capabilities and limitations of the two parties in an interactive system. As the underlying cognitive and system theories are built into the model, the reason why some problem exists can be identified and addressed. Alternative design solutions are then driven by theoretical insight, rather than through a potentially expensive 'generate and test' cycle of ad-hoc changes.

As syndesis uses mathematics to express the structure and behaviour of the system and its user, it could be argued that for HCI, the cost of building these models is high, and that they require skills that are not common in the user interface design community. However, formal models are playing an increasing role in software engineering, and the myths surrounding formal methods are gradually being demolished [bh94]. Syndetic modelling also requires some understanding of ICS, though we should emphasise that the formal model of ICS is generic; it can be reused across applications by contextualising the architecture to the specific application. In the short term, the main value of syndesis is in the analysis of complex or novel techniques and applications, for example gesture [d95]. In the longer term, the approach should provide a versatile framework for expressing and understanding interaction between users, systems, and the context in which both are located.

7.4 Semiotics

Semiotics is the science that studies human and non-human signs as systems of communication, meaning and interpretation. In the context of HCI, semiotics - more specifically computer semiotics [a90], amongst others - will focus on interfaces as constellations of signs, constructed by human programmers and interface designers, that need to be interpreted by other humans: the users. The specific focus on signs at the interface distinguishes semiotics from other approaches discussed in this paper. Computer-based signs are the essence of HCI, but very little systematic basic research has been done on a description of all aspects of the computer-based sign and on the interaction between different signs.

Semiotics can serve an integrative function in HCI, by making signs the central concept of the human-machine interface. Computer semiotics offers a kind of thesaurus of computer-based signs, containing all system- and user-aspects of each particular sign and of each combination of signs. However, whether such a comprehensive theory is possible is an open question. An example may illustrate what is meant here, compare also [v94].

Consider the thesaurus-entry "mouse pointer". From an implementation-centred perspective, this mouse pointer consists of a number of lines of code in a specific programming language. This program module translates electric currents from the mouse device to changes in the X-and Y-coordinates of a pointer on a screen. From a user-centred perspective, this mouse pointer indicates a certain position on a computer screen where specific actions can be triggered using the mouse button(s). Which actions can be triggered, depends on the context in which the mouse-pointer is being used. Contributions from user-centred approaches within HCI to this item in the thesaurus of computer-based signs include studies on the problematic conceptual and motoric relation between the horizontal movement of the mouse on a desk and the vertical movement of the mouse-pointer on the computer screen for some tasks, e.g. in [kma86]. System-centred contributions include formal representations of the relation between the mouse device and the mouse pointer [mcr90], amongst others. The integrational effort of semiotics is to serve as a vehicle to bring together these different contributions.

It should be clear from the example that the way in which computer semiotics can integrate contributions from different approaches, is different from the other integrative techniques discussed in this paper through its focus on the sign. The framework that computer semiotics can offer - such as a thesaurus of computer-based signs - seems to open interesting perspectives for design practice. Unfortunately, no systematic effort has been undertaken to develop such a large-scale project.

8 Technical support for integration

One way of 'encapsulating' a design notation or process for transfer from research into design practice is to provide software tools that support the use of a notation or provide guidance for following a design process. For some approaches, for example those that employ 'formal' software development methods, the absence of appropriate tool support has been cited as a factor contributing to their slow uptake by practitioners. This situation is exacerbated in multi-disciplinary analysis, where not only must the individual techniques be supported, so must the integration of the output generated by those techniques. In addition, given a range of disparate modelling approaches, designers may require some systematic basis for deciding which approaches might provide the most leverage for their problem. This section describes two software tools, developed as part of the integration effort within Amodeus, that attempt to address these concerns.

8.1 Jackie and Walter

The Jackie and Walter demonstrators were developed for two purposes. First, they illustrate possible support for global integration of modelling results by technical means. Secondly, they serve as a vehicle for concretising and better understanding the inter-relationships between the different integrative notations involved: aRD, DSD/DR and QOC. The basic assumption underlying the work on technical integration is that the different notations can be seen as views onto the same design information.

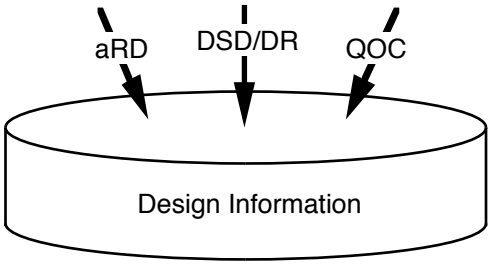

Figure 8.1: Supporting multiple views onto design information

The information base contains representations of the evolving artefact along with the intended user tasks and constraints on the design space. Modelling input can be in either one of these categories: potential problems can typically be seen as constraints, whereas the suggested user procedures and user-interface designs would belong in user tasks and artefact representations, respectively.

8.1.1 Jackie

The purpose of the Jackie demonstrator is to illustrate the behaviour and appearance of a technical integration tool from the designers', integrators' or modellers' perspective. It is created as an envisionment in HyperCard and illustrates a range of potential tools to support integration throughout the design process. The tools broadly fall into two classes: browsing and manipulation of the contents of the information base, and creation of argumentative views onto the information. The figure below illustrates how a certain piece of information is created through a type-specific editor and then serves different argumentative roles in a QOC structure and a DSD frame.

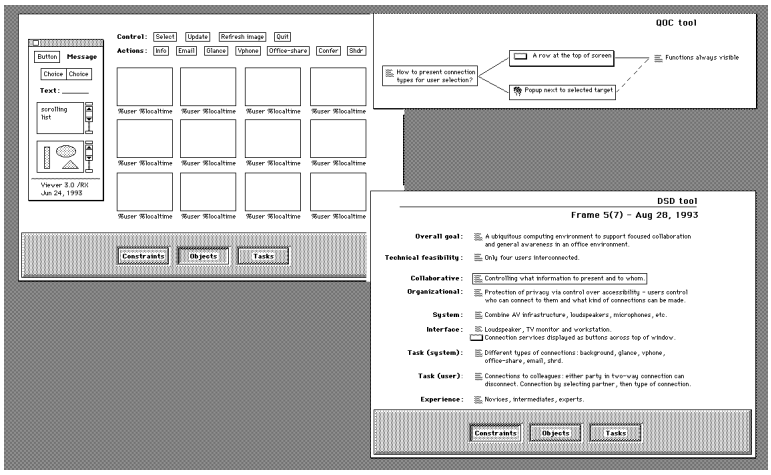

Figure 8.2: Screenshots from Jackie. The left window shows a user-interface design created an imaginary, but typical user interface builder. In the top right window, the design proposal serves as an option in a QOC structure. The lower right window shows the same design proposal appearing in the interface aspect of a DSD frame.

8.1.2 Walter

In creating the Walter demonstrator, we aim at developing a representation for the information base that would allow for persistent storage of design information and the subsequent construction of manipulation and argumentation tools as illustrated in Jackie. The requirements on the representation are derived from the need to support all the argumentative notations involved, as well as from the range of design support that the tools are intended to offer.

In the analysis of aRD, DSD/DR and QOC, a "least common denominator" set of primitive argumentative categories was identified: questions, options, criteria, argumentation and annotation. The relationship between four of the primitives and the three DR representations used within Amodeus is shown in Table 8.1, below. The fifth category, *annotation* , is more general in the sense that it denotes information not allocated to any of the other four roles.

Primitive	**QOC**	**aRD**	**DSD**
Question	question	need	problem
Option	option	measure, decision	option
Criterion	criterion	goal	commitment
Argumentation		consequences	justification

Table 8.1: The Relationship between general argumentative categories and specific DR notations.

Different link types were identified to provide the necessary semantics for relations between argumentative categories. Examples of link types are links for relating questions and options (e.g., the QO link for

connecting an option to a question, and the OQ link for connecting a subsequent question to an option) and judgement links that represent the assessment of one object by another.

The argumentative categories and link types, together with an enumeration of the required information types, form the basis for the information representation in the Walter demonstrator. To implement Walter, an object-oriented database was used. The overall architecture is presented in the following figure.

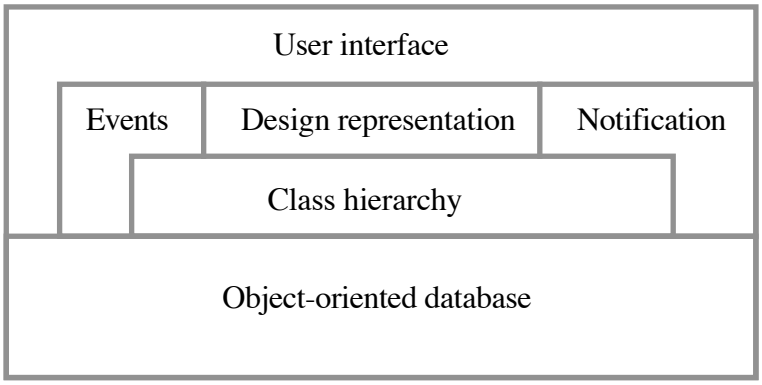

Figure 8.3: Architecture of the Walter Demonstrator

The underlying database stores objects of the different types. Action propagation is achieved by means of an event broadcasting mechanism, approximating the message passing or method encapsulation of conventional object-oriented frameworks. The current implementation fully supports the three notational views on the design information, including simple reasoning about the information. For example, if a DSD/DR view is created, the Justification slot is filled automatically by computing the transitive coverage of the information base following positive assessment links. The user interface currently consists of text-based command-line interaction. In its stead, various graphical tools for information manipulation and argumentation as illustrated by the Jackie demonstrator can be developed and plugged in.

8.1.3 Tentative conclusions

As mentioned above, the purpose of the work on technical integration was twofold: to illustrate technical means for supporting integration, and to explore the inter-relationships between the different integrational approaches involved.

Regarding the first goal, the Jackie demonstrator serves to illustrate the range of design support possibilities arising from regarding integration as a matter of different views onto the same design information. The Walter demonstrator illustrates the feasibility of implementing the necessary information representation.

The second goal was addressed in part through finding a common representation for implementation. The argumentative categories identified above highlight that the different notations are similar as regards argumentative structure. It is, however, important to realise that this similarity cannot be used simplistically in an integrative tool to present any information through any view. Modelling information is created in a certain context, and its interpretation is not always clear outside that context (as witnessed by the QOC and DSD integration experiments). The approach reported here may partially address the problem of providing appropriate interpretative context, primarily by the convenient use of links back to the structure where a certain modelling result was originally inserted.

8.2 DDAS: A Designer' Decision Aiding System

Given the wealth of techniques and approaches available, a means for helping design practitioners decide what would be the most useful for them in a particular situation was devised - DDAS (Designer's Decision Aiding System). The challenges facing the creation of DDAS were threefold:

- to find a common language and use it to describe what each approach can do;
- to describe the design problem in such a way that it could be correlated to the relevant abilities of the modelling approaches, using that language;
- to find a way for the system to evaluate the appropriateness of each modelling approach to a design problem.

In order to meet these challenges an approach was developed for the elicitation of expert knowledge about the potential of the modelling approaches; for representing the meaning of that potential to the designer and from there recommending to the designer the most suitable technique(s). It is based primarily on principles of Systems Thinking, and in particular Soft Systems Methodology (SSM) [c81, cs90], for eliciting and structuring knowledge and on fuzzy sets through test score semantics [z89] for representing the meaning of relationships between the modelling techniques and the design problem and for reasoning about them. [dds94a, dds94b, dds94c, dds94d].

The representation of the expert knowledge needs to be capable of accommodating the multi-disciplinary character of the techniques. It should represent the elements that are common in the sense that they share similar goals, or they explain the same phenomena. It should represent the differences between modelling techniques by capturing their strengths and weaknesses in relation to specific design problems. And finally it should represent the relationships of the various components within a technique and those between different techniques. This could be accomplished by representing the expert knowledge about the modelling techniques and their use in design, through their identified relationships to parts of the possible problems (sub problems) they can address. This in turn means that the representation of the expert knowledge is such that the designer can express his problem within it.

Problem understanding, already a complex process, is made more difficult by the tendency for large parts of design practice to be carried out in an unstructured, almost haphazard, way rendering problem understanding secondary. For a designer aiding system, it is more important to express the problem in ways that are compatible with the representation of the expert knowledge. Thus formulating the problem becomes a question of restricting the designer to descriptions derived from the expert knowledge.

The product of the interaction of the designer with the expert knowledge results in an expression of the design problem against which the modelling techniques have to be evaluated. This evaluation is the aid to the designer and constitutes the reasoning which due to the nature of the problem and the representation of the expert knowledge, will have to be made on the basis of linguistically expressed as opposed to quantifiable statements.

Systems thinking offered a means to operationally define the subsystems relevant to the generic design space. The links of those subsystems to the modelling approaches were elicited primarily from relevant texts and communications with experts. In this way the primary purpose of translating the design space into a mode where the role of the modelling approaches in the design space is described in an operational way was achieved. Then the application of test score semantics can evaluate that role in the part of the design space selected by the designer. Finally the system makes recommendations to the designer on the basis of the evaluations.

The evaluation of the links between the subsystems and the modelling approaches by treating them as fuzzy constraints allows for greater flexibility in representation and also sustains considerably more of the designers understanding of his problem in order to be able to aid him in his decision for choice.

9 Transfer prospects for integration techniques: Usability and utility for designers?

Given that within AMODEUS, most of the individual modelling techniques are still at a basic research status, modelling-*integration* techniques are even more embryonic in development. We suggest that only the semiformal 'global' approaches, QOC/DSA, DSD, aRD and (possibly) Co-modelling can be considered as candidates for transfer to HCI practitioners (i.e. ready for substantive training and subsequent serious use). Thus, it is too early to assess the cost/benefit trade-off of training designers to use the IF notation or syndetic modelling for example. For this reason, the focus of this section is on issues related specifically to use of the global integration techniques.

9.1 Global integration techniques as a 'modelling-designer user interface'?

Most of the analysis of integrational techniques addresses their respective merits for modellers seeking to understand the interconnections between modelling approaches, since this is the original purpose for which they were developed. However, during the course of the project, the possibility has arisen of using integrational techniques to communicate modelling to designers. From a modelling transfer perspective, we therefore ask the question, *Can integrational techniques be used to communicate modelling to designers, and what is the range of possible ways in which this may be done?* In this section, we briefly outline the rationale underlying this particular question, and a framework which may assist in exploring answers.

It is currently not realistic to think of design practitioners trying to integrate their own multi-disciplinary modelling using an integrating technique, since the individual modelling techniques (with the possible exception of the FSM and LIM formal system modelling) are themselves not yet sufficiently developed for use by practitioners. We suggest that at present, the most likely form of contact between designers and integrational techniques is in terms of how the global integrational techniques (DSD, QOC, and co-modelling) can be used to communicate modelling which has been carried out by modelling experts, acting in a consultancy role as we have done in AMODEUS. (In the future, one might envisage such experts as members of design teams).

Elsewhere [bbm95], we have drawn the analogy between user interfaces, and integrational techniques (indeed all encapsulations of modelling)—integrational techniques can be used as user interfaces to present complex modelling analyses to design practitioners. However, only the global integration techniques can be considered as candidate 'modelling-designer interfaces'; the local techniques are primarily for internal project purposes, and are at a very early stage of development.

Our concern is therefore to identify the most effective ways in which global integration techniques can assist designer-modeller communication. In order to make the analysis more systematic, we propose several dimensions which together describe a space of possible ways in which integrating expressions can be used in designer-modeller meetings. Firstly, this space clarifies how the different kinds of integration process

which have been explored within AMODEUS differ from one another. Secondly, we can introduce 'designers' into this space, allowing us to consider the implications of the different integration approaches for modeller-designer collaboration.

Before introducing the dimensions and the space they describe, let us briefly remind ourselves of two particular processes by which global integration techniques have been used in AMODEUS. These concrete examples, described in detail in earlier sections, serve to highlight the fact that they differ in several important respects.

9.2 The process of global modelling integration: two examples

QOC integration of modelling on ECOM. In this exercise, modellers initially worked separately, but were asked to *focus their analyses on an existing QOC design space* (describing the designers' rationale); they were also required to *re-express their analyses in terms of QOC*, and these contributions were then *re-worked and integrated* by a QOC specialist into a single design space. In this process, inter-modeller communication was sometimes *face to face* (on the fly modelling sessions at project meetings), and by *e-mail and document exchanges*. (The latter case also applies to the DSD integration work on the modelling of CERD).

Co-modelling integration of modelling on CERD. In this exercise, modellers initially worked separately, but *met face to face* to build on each others' modelling, and *construct an 'integrated' account together*. This was facilitated by a collator, who then reflected on, and *wrote up the results of the meeting*. A variation on this process was where *communication was by e-mail*, which introduced different dynamics.

9.3 Four process dimensions to global integration

We propose that the ways in which global integration has been achieved can be usefully distinguished according to where they lie on four salient dimensions. The first two dimensions are:

(1) How do the modellers communicate with each other?

- *Asynchronously*: modellers worked separately without close coordination.
- *Partially synchronously*: modellers initially worked separately without close coordination, but may later coordinated work by e-mail for example.

(2) When is the integration work done?

- *Retrospectively*: integration after modelling by a specialist, scribe or coordinator; modellers do not express their analyses in the mediating representation, or try to build on a design space
- *Concurrently*: the integration process occurs in tandem with the modelling, each feeding the other; although there may be a scribe or collator, all participants assume responsibility for constructing an integrated analysis.

Visualised spatially, these two dimensions describe a plane within which it is possible to locate different modelling integration exercises which have been conducted within AMODEUS (Figure 9.1). One aim of the dimensions (and diagram) is to clarify what properties of co-modelling distinguish it from other methods of integration used to date, why in principle use of QOC or DSD could also draw on these properties, and what forms these unexplored possibilities might take. These points are taken up shortly.

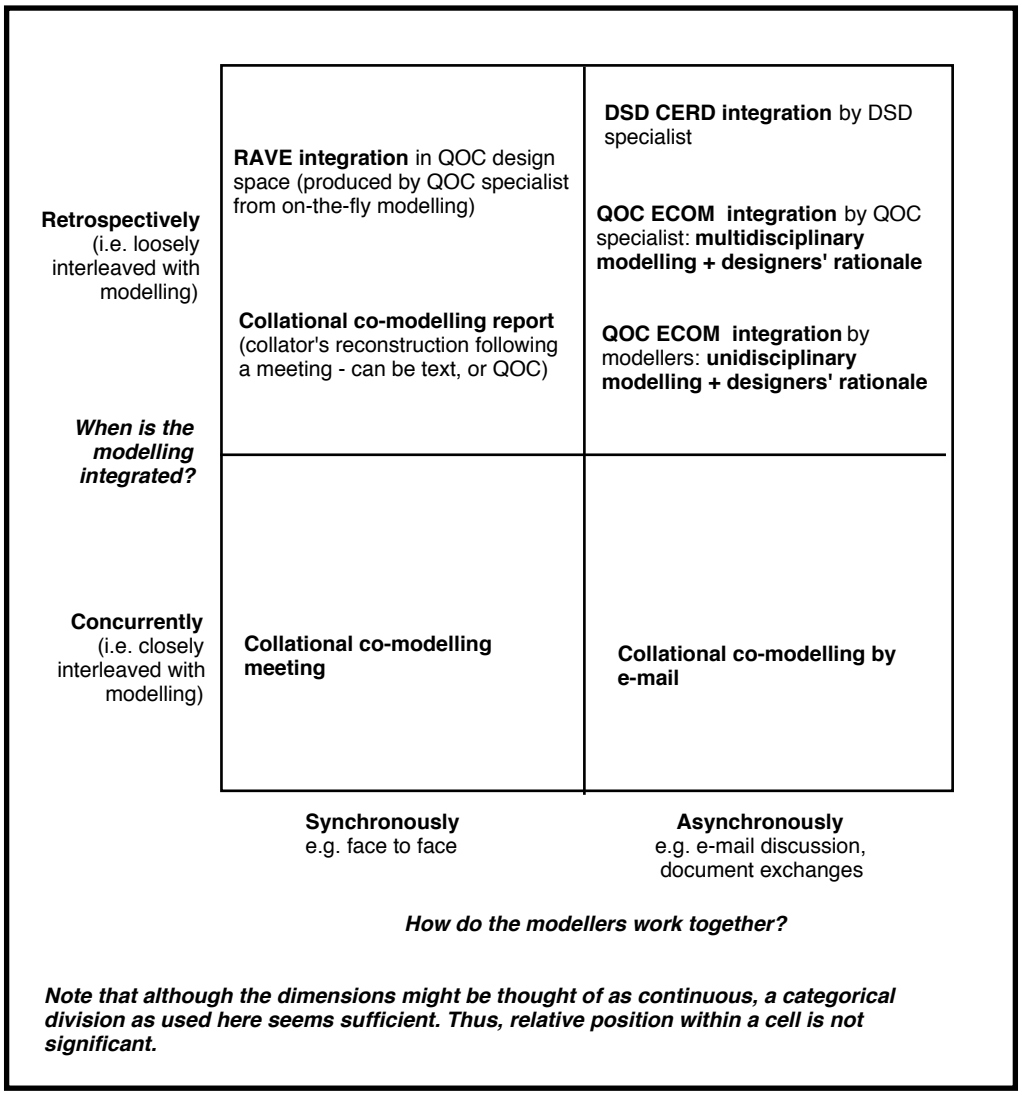

Figure 9.1: Two dimensions for classifying integration strategies: *When is the modelling integrated* relative to the actual modelling conducted by different groups, and *How closely do the modellers work together?*

While the first two dimensions address the communication between modellers, the next is concerned with the way that the integration scheme might structure or guide both the communication and the organisation of the integrated modelling.

(3) How is the integration scheme used?

- *To set the agenda:* In the case of the ECOM exemplar, a QOC design space was used to describe the problem space and modelling issues.
- *purely retrospectively :* DSD was used to summarise and collate modellers' analyses of the CERD.
- *as a weak structure to chair meetings:* In a co-modelling meeting both the collator and the participants are aware of the structure of a co-modelling report. This shared knowledge has the effect of shaping the overall structure of the design meeting.
- *as an explicit structure in meetings:* An integration scheme such as QOC that provides a notation for expressing the structure of an evolving design space can be used to summarise key modelling points as they arise, and hence feed directly into subsequent rounds of modelling.

The four options listed above are selected points within a range of possibilities and are not exclusive. In fact it is possible that one integration scheme (for example QOC) might be used to set the agenda while another, for example co-modelling, provides a weak structure to the meeting. In the latter context one further dimension is important for understanding the use of integration techniques:

(4) How formal is the integrational scheme?

- *Relatively informal*: for example, headings from a co-modelling report
- *Relatively formal*: DSD tables or a terse QOC graph

There is clearly a trade-off between a loose, informal but flexible scheme such as a co-modelling report and a more structured but specific scheme like a QOC graph. As the experiences reported in Sections 3 and 4 suggest, the structure of an integrational scheme may sometimes impede modellers by making it difficult for them to express salient aspects of their analyses. In contrast, more formal representations have the virtue of focusing attention on the key aspects of a design problem, and can provide a more succinct account and record of the rationale leading into resolution of design issues. The tension between these two concerns does suggest that it may be profitable to investigate a hybrid scheme:

QOC-based co-modelling face to face

[partly synchronous modelling] modellers initially work separately without close coordination, but then extend this in meetings

[concurrent integration] the integration process occurs in tandem with the modelling, each feeding the other; although there is a QOC scribe responsible for maintaining the QOC, all participants assume responsibility for constructing a QOC space.

[integrational structure is explicit] the collaborative construction of QOC during co-modelling shapes its focus and depth, e.g. through defining initial Questions to be discussed, and by feeding back the state of the design space as it is changed (a multi-author electronic whiteboard/graphical editor would facilitate such a meeting).

[integrational structure is formal] the structure of QOC provides a well defined framework for recording the design decisions or assessments made during the meeting. *However*, the co-modelling meeting itself provides an informal integrational structure in which the more formal QOC can be called upon as and when appropriate.

A similar framework could be defined just by replacing QOC by DSD. These two suggestions build on co-modelling as it has been tried to date. Rather than leave the emerging integrational account implicit (a function of the individuals' memories), which is only made explicit retrospectively by the collator, the semiformal structure of QOC or DSD could be used as a resource to assist the co-modelling. This suggests that design rationales themselves can fill a spectrum of roles within multi-disciplinary integration, as set out in Figure 9.2.

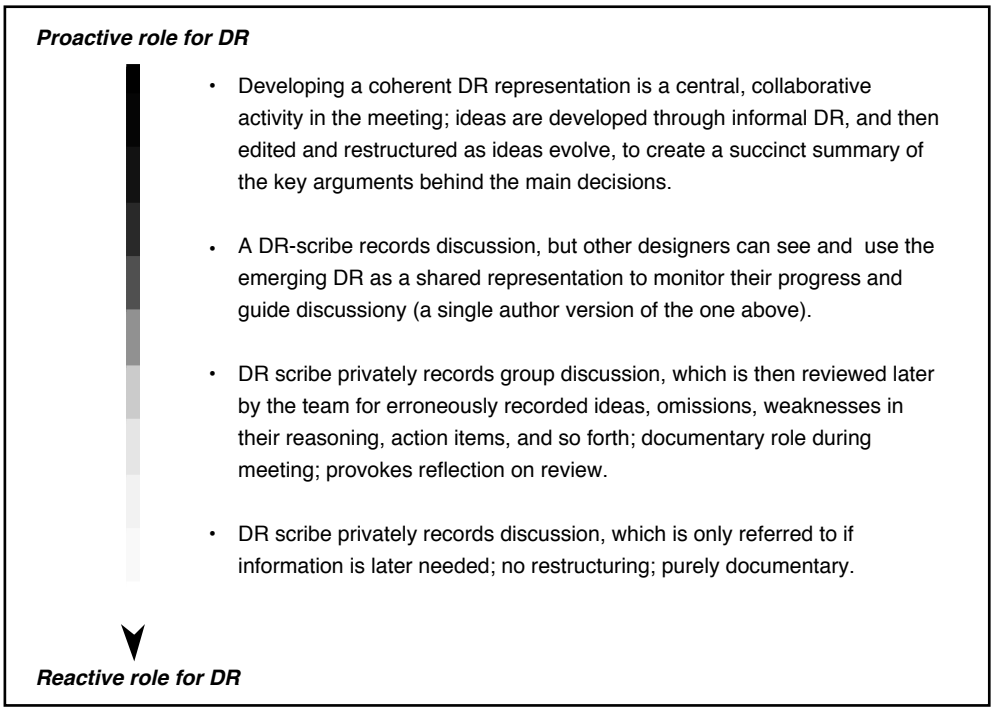

Figure 9.2: Different role for design rationale within the integration process (from [bs95b]).

An argument against using QOC or DSD in the context of a co-modelling meeting might be that this would sacrifice the freedom which modellers have to model and integrate without worrying about encoding their ideas semi-formally as QOC/DSD. However, this issue is already well understood from research on the use of QOC by designers [bs95a; bs95b]. Use of more structured representations of the integrated analysis is not meant to straitjacket modellers into interacting purely in terms of Questions, Options and Criteria, for instance. The goal is to achieve cycles of 'free talking' (as occurs in co-modelling at present), and reflection, by checking the state of the QOC/DSD design space, and adjusting co-modelling accordingly, in order (for example) to cover unaddressed Questions, or weaknesses in Options. The more evolving, more structured picture of the integrated analysis is meant to serve as an external, group memory, which can 'talk back' to the co-modellers, and augment their collective awareness of the state of the 'multi-faceted story' emerging from the co-modelling.

9.4 Introducing designers into the integration space

Having considered how *modellers* work together within the integration processes which we have followed within AMODEUS, let us now consider the question of *How do modellers and **designers** work together?*

Although the ECOM exercise [bbm95, bm94, bbm94] showed that designers can get to grips with QOC, on its own it was a terse representation which required more detailed explanation (provided by referencing the modelling reports). In contrast, initial analysis [bsh95] of the CERD co-modelling sessions with designers indicates that, once the designers had some background information on the modelling techniques, this was an effective way to convey the potential of the approaches, and in particular, the way in which modellers work.

In sum, the evidence from these two large scale studies is that initially, the most effective strategy is for modellers themselves to explain and discuss their analyses with designers, ideally face to face, rather than by phone or e-mail. Meeting to work together on a specific design problem is more productive, since the

dynamics of a face to face meeting allow for rapid clarification and adjustment by each party to the responses of the other. Although we have been unable to conduct longitudinal studies of modeller-designer collaboration, we hypothesise that written modelling documents or paper/online QOC design spaces may only be an attractive, effective 'modelling-designer user interface', once designers have developed good working relationships with the modellers (as consultants or members of the design team), and they are relatively familiar with the basic ideas behind a given modelling technique.

9.5 Envoi

When investigating the research question "what relationships are there between modelling approaches?" , if the insights gained from using a particular integrating expression are substantial, then substantial costs (in terms of effort to use the integrating expression) may be judged worthwhile, and tolerated by researchers. However, the cost/benefit trade-off is substantially different for designers if they are expected to use the integrating expression in some way (as in ECOM), since their tolerance of costs will almost inevitably be lower than that of researchers, and their tasks will be different. Investigation of the cost/benefit trade-off for integration techniques in design practice remains an open area of research.

10 Conclusions

Human-computer interaction has developed from considering the layout and operation of simple graphical or textual interfaces to examine technologies such as media-spaces, gesture, and multi-modal interfaces. These touch on a broad range of issues ranging from 'hard' system constraints, through human perception and cognition, to broad social issues concerning the use of appropriate systems to support workers needs. It is clear that insight into the design of interactive systems needs to draw, at times, on any or all of these disciplines. What we have begun to address in this paper is the way in which those insights can or should be brought together to provide a coherent picture of the design problem. In setting out various approaches to integration, we have had to confront three issues:

1 *What* do we mean by integration?
2 *When* can integration be carried out?
3 *How* is it carried out?

We have chosen to answer these by taking integration to mean the embedding of the results, process or underlying theory of two or more modelling techniques into a common space. For integration of results, the common space is a shared representation, for example QOC or DSD frames, within which modellers can locate and assess design options. Integration of process, through co-modelling, supports the iterative analysis of design issues, where modellers can request from or provide to each other evidence that supports a design claim. Integration of theory brings together the foundations of specific modelling techniques within some common framework. Here, both the insight and the supporting theory are available to answer questions both about the design and about the reason or process through which the modelling technique suggests that insight.

A number of factors that might influence the choice of integration technique have been suggested. These include the progress of the artefact through some design process, the modelling techniques that are employed, and of course the kind of questions that are being asked. The mapping between actual people and modellers' roles may be significant: where for instance a single human-factors analyst is conducting both system and user modelling, the situation may favour a process-based integration technique.

Orthogonal to these questions are issues of utility: is the expected gain from integration commensurate with the expected insight? Does it significantly add to the cost of doing the modelling in the first place?

From the viewpoint of Amodeus, one of the most challenging problems has been how modelling and integrative techniques can be transferred into design practice. The assessment of individual modelling approaches has been a non-trivial problem in itself, and apart from the ECOM study using QOC, it has not been possible to conduct any systematic exercise involving both modellers and designers. Such assessment of the other integrative techniques must wait for the modelling approaches to be taken up within the design community, and for the integrative techniques themselves to mature.

Acknowledgements

This work was carried out as part of the Amodeus-2 project, ESPRIT Basic Research Action 7040 funded by the Commission of the European Communities.

References

Most of the technical reports produced within Amodeus-2, including those referenced below, are available electronically via FTP and the world-wide web. The appropriate resource locators are given below.

`ftp://ftp.mrc-apu.cam.ac.uk/pub/amodeus`
`http://www.mrc-apu.cam.ac.uk/amodeus/amodeus.html`

[acn92] Abowd, G., Coutaz, J. & Nigay, L. Structuring the Space of Interactive System Properties, *IFIP Transactions A-18, Engineering for Human-Computer Interaction, Proceedings of the IFIP TC2/WG2.7 Working Conference*, Ellivuori, Finland, J. Larson and C. Unger Ed., North-Holland Publ., p. 113-128.

[abc91] Arango, G., Bruneau, L., Cloarec, J.-F. and Feroldi, A. A Tool Shell for Tracking Design Decisions. IEEE Software, March, 1991, pp. 75-83.

[a90] Andersen, P.B. A theory of computer semiotics: semiotic approaches to construction and assessment of computer systems, Cambridge: University Press, 1990.

[bh89] Barnard, P.J. & Harrison, M.D. Integrating cognitive and system models in human-computer interaction. In A. Sutcliffe & L. Macauley (eds), *People and Computers V: Proceedings HCI'89*. Cambridge University Press, 1989.

[bm95] Barnard, P.J. & May, J. Interactions with advanced graphical interfaces and the deployment of latent human knowledge. In *DSV-IS'94: Proc. Eurographics Workshop on Design, Specification and Verification of Interactive Systems,* Springer Verlag, 1995. To appear.

[b93] Bellotti, V. Integrating Theoreticians' and Practitioners' Perspectives with Design Rationale. In *Proceedings of InterCHI'93: Human Factors in Computing Systems*, 1993, ACM Press: NY, pp. 101-106.

[bb+95] Bellotti, V., Blandford, A.E., Duke, D.J., MacLean, A., May, J. & Nigay, L. *Controlling Accessibility in Computer Mediated Communications: a Systematic Analysis of the Issues*. Submitted for publication.

[bbm94] Bellotti, V., Buckingham Shum, S. and MacLean, A. Assaying Multidisciplinary Modelling Using QOC as a Mediating Design Expression: The 'EuroCODE' Design Exemplar. Amodeus-2 Project, *Working Paper ID/WP30 [Amodeus ftp-site: /design/id_wp30.ps.Z]*, 1994.

[bbm94] Bellotti, V. Buckingham Shum, S. and MacLean, A. Assaying multidisciplinary modelling using QOC as a mediating design expression: The EuroCODE AV exemplar. Amodeus-2 Technical Report ID/WP30. 1994.

[bbm95] Bellotti, V., Buckingham Shum, S., MacLean, A. and Hammond, N. Multidisciplinary Modelling In HCI Design...In Theory and In Practice. In *Proceedings of ACM CHI'95: Human Factors in Computing Systems*, 1995, ACM Press: New York

[bm94] Bellotti, V. and MacLean, A. Integrating and Communicating Design Perspectives with QOC Design Rationale. Amodeus-2 Technical Report ID/WP29,1994.

[b94] Bernsen, N.O. Foundations of multimodal representations: A taxonomy of representational modalities. *Interacting with computers*, Vol.6, No.4, pages 347-371, 1994.

[br94] Bernsen, N.O. & Ramsay, J. *An executive summary of the DSD framework illustrated by two worked exemplars.* In Shum, S.B., Jørgensen, A., Hammond, N. and Aboulafia, A. (Eds.): Amodeus-2 HCI Modelling and Design Approaches: Executive Summaries and Worked Examples.

[bv95a] Bernsen, N.O. & Verjans, S. Designing interfaces by Information Mapping - A case study of the Cerd design. Amodeus-2 Technical Report TM/WP13.

[bv95b] Bernsen, N.O. & Verjans, S. Information mapping: Knowledge-based support for user interface design. *CHI'95 Workshop on knowledge-based support for the user interface design process*. 1995.

[bhb95] Blandford, A.E., Harrison, M.D. & Barnard, P.J. Using Interaction Framework to guide the design of interactive systems. *International Journal of Human-Computer Studies*. To appear; also available as Amodeus-2 technical report ID/WP42.

[by95a] Blandford, A.E. & Young, R.M. Separating User and Device Descriptions for Modelling Interactive Problem Solving. In *Proceedings INTERACT'95*. To appear.

[by95b] Blandford, A.E. & Young, R.M. Applying Programmable User Models to Real Design Problems. Submitted for publication.

[b91] Booch, G. Object Oriented Design - with Applications. Benjamin-Cummings. 1991.

[bh94] Bowen, J.P. and Hinchey, M.G. Seven More Myths of Formal Methods: Dispelling Industrial Prejudice. In *FME'94: Industrial Benefit of Formal Methods*. M. Naftalin, T. Denvir and M. Bertran (Eds). pp. 105-117. Lecture Notes in Computer Science, Vol 873. Springer-Verlag, 1994.

[bs94] Buckingham Shum, S. (Ed). Executive Summaries of Amodeus Modelling Approaches. Amodeus-2 Technical Report TA/WP12, 1994.

[bs94a] Buckingham Shum, S. (Ed). Preliminary modelling of the CERD flight sequencing tool. Amodeus-2 Technical Report TA/WP23, 1994.

[bs95a] Buckingham Shum, S. Analyzing the Usability of a Design Rationale Notation. In *Design Rationale: Concepts, Techniques, and Use,* Moran, T.P. and Carroll, J.M., (Ed.), Lawrence Erlbaum Associates: Hillsdale, NJ, (in press).

[bs95b] Buckingham Shum, S. 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. 1995, (in press).

[bsd+94] Buckingham Shum, S., Duke, D.J., Hammond, N., & Jørgensen, A. CERD: Functionality, Design Process, and Modelling Issues. Amodeus-2 Technical Report TA/WP30. 1994.

[bsh94] Buckingham Shum, S. & Hammond, N. Argumentation-Based Design Rationale: What Use at What Cost? *International Journal of Human-Computer Studies*, 40, 4, 1994, pp. 603-652.

[bsh95] Buckingham Shum, S. & Hammond, N. Transfer & Evaluation of HCI Modelling for the CERD Air-Traffic Control Tool. Report in: AMODEUS Project Deliverable D12: Transferring and Assaying HCI Modelling and Design Approaches (Year 3), 1995.

[cc92] Carey, T.T. and Crentsil, J. Integrating Widget Design Knowledge with User Interface Toolkits. In *Proceedings IEEE CASE Workshop*, 1992, Department of Computing and Information Science, University of Guelph, Canada

[c81] Checkland, P.B. Systems Thinking, Systems Practice, Wiley New York, 1981.

[cs90] Checkland, P.B.and Scholes J. Soft Systems Methodology in action, Wiley, New York. 1990.

[c87] Coutaz, J. PAC, An Object-Oriented Model for Dialog Design. In H. Bullinger & B. Shackel, Eds, *Human-Computer Interaction - Interact'87*, pages 431-436. North-Holland, 1987.

[c89] Coutaz, J. Architecture Models for Interactive Software. In *ECOOP'89: European Conference on Object-Oriented Programming*. Lect. Notes in Computer Science. Springer-Verlag, 1989.

[c93] Coutaz, J. Software Architecture Modeling For User Interfaces. In *The Encyclopedia of software Engineering*. pages 38-49. Wiley, 1993.

[cns93] Coutaz, J., Nigay, L. & Salber, D. The MSM Framework: A Design Space for Multi-Sensory-Motor Systems. In L. Bass, J. Gornostaev, & C. Unger (eds) *Lecture Notes in Computer Science 753 (EWHCI'93) Selected Papers*. pages 231-241. Springer-Verlag, 1993.

[cns+95] Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J., and Young, R. Four Easy Pieces for Assessing the Usability of Multi-modal Interaction: The CARE Properties. In Proceedings INTERACT'95. To appear, 1995.

[cms91] CMSI (1994) CM/1 Release 2, Corporate Memory Systems, Inc., 11824 Jollyville Road, Austin, TX 78759, USA. (organizational memory hypertext groupware for MS Windows)

[cb88] Conklin, J. and Begeman, M.L. gIBIS: A Hypertext Tool for Exploratory Policy Discussion. *Transactions on Office Information Systems*, 6, 4, 1988, pp. 303-331.

[dds94a] Darzentas J., Darzentas J.S., Spyrou T. Defining the Design Decision Space: Rich Pictures and Relevant Subsystems. Amodeus Project Document: TA/WP 21, 1994.

[dds94b] Darzentas J., Darzentas J.S., Spyrou T. An Architecture for Designer Decision Aiding in Proceedings of the 5th Meeting of the European Working Group on Decision Support Systems, Finland, 1994.

[dds94c] Darzentas J., Darzentas J.S., Spyrou T. Fuzzy Reasoning and Systems Thinking in a Decision Aid for Designers in Proceedings of Second European Conference on Intelligent Techniques and Soft Computing, Aachen, pages 1609-1614, 1994.

[dds94d] Darzentas J., Darzentas J.S. & Spyrou T. DDAS: a Designers Decision Aiding System, submitted to Journal of Decison Systems, Hermes, 1994.

[db92] Dourish, P. & Bly, S. Portholes: Supporting Awareness in Distributed Work Groups. In *CHI'92: Proc. ACM Conference on Human Factors in Computer Systems*. Addison-Wesley, 1992.

[dh93] Duke, D.J. & Harrison, M.D. Abstract Interaction Objects. *Computer Graphics Forum*. 12 (3). NCC/Blackwell, 1993. Conference Issue: Proc. Eurographics'93.

[dh94a] Duke, D.J. & Harrison, M.D. A Theory of Presentations. *FME'94: Industrial Benefit of Formal Methods*. M. Naftalin, T. Denvir & M. Bertran. Lect. Notes in Comp. Sci. Vol. 873. pp.271-290. Springer Verlag, 1994.

[dh94b] Duke, D.J. & Harrison, M.D. From formal models to formal methods. In R.N. Taylor & J. Coutaz (eds), *Software Engineering and Human-Computer Interaction*. Volume 896 of Lecture Notes in Computer Science, pages 159-173, Springer Verlag, 1995.

[dh95] Duke, D.J. & Harrison, M.D. Interaction and Task Requirements. In *DSV-IS'95: Proc. Eurographics Workshop on Design, Specification and Verification of Interactive Systems,* Lecture Notes in Computer Science, Springer Verlag, 1995. To appear.

[d95] Duke, D.J. Reasoning about Gestural Interaction. *Computer Graphics Forum*. 14 (3). NCC/Blackwell. Conference Issue: Proc. Eurographics'95.

[dbd+95] Duke, D.J., Barnard, P.J., Duce, D.A. & May, J. Syndetic Modelling. Amodeus-2 Technical Report ID/WP49, 1995.

[e76] Eco, U. A Theory of Semiotics. Indiana University Press, 1976.

[ffz94] Faconti, G. P. Fornari, A. & Zani, N. Visual Representation of Formal Specification: an application to Hierarchical Logical Input Devices. In *DSV-IS'94: First Eurographics Workshop on Design, specification and Verifaction of Interactive Systems,* pages 293-316. Springer-Verlag, 1995.

[gmm+92] Gaver, B., Moran, M., MacLean, A., Lovstrand, L., Dourish, P. & Carter, K.: Realising a Video Environment: EuroPARC's RAVE System. In *CHI'92: Proc. ACM Conference on Human Factors in Computer Systems.* Addison-Wesley.

[hbb94] Harrison, M.D., Blandford, A.E. & Barnard, P.J. The requirements engineering of user freedom. In F. Paterno', Ed, *DSV-IS'94: First Eurographics workshop on the design, specification and verification of interactive systems*, pages 181-194. Springer Verlag, 1995.

[kma86] Karat, J., McDonald, J.E. & Anderson, M. A comparison of menu selection techniques: touch panel, mouse and keyboard. In: *Int. J. Man-Machine studies*, **25**, pages 73-88, 1986.

[l90] Lee, J. SIBYL: A Tool for Managing Group Design Rationale. In *Computer Supported Cooperative Work*, 1990, ACM Press: New York, pp. 79-92.

[mbs93] MacLean, A. Bellotti, V. & Shum, S. Developing the design space with Design Space Analysis. . In P.F. Byerley, P.J. Barnard, & J. & May (eds) *Computers, Communication and Usability: Design issues, research and methods for integrated services*. North Holland Series in Telecommunication, pages 197-219. Elsevier, 1993.

[mby90] MacLean, A., Bellotti, V. and Young, R. What Rationale is There in Design? In *Proceedings of IFIP INTERACT'90: Human-Computer Interaction*, Elsevier Science Publishers B.V.: Netherlands: , 1990, pp. 207-212.

[myb+91] MacLean, A., Young, R.M., Bellotti, V. and Moran, T. Questions, Options, and Criteria: Elements of Design Space Analysis. *Human-Computer Interaction*, 6, 3 & 4, 1991, pp. 201-250.

[mym89] MacLean, A., Young, R.M. and Moran, T. Design Rationale: The Argument Behind the Artifact. In *Proceedings of CHI'89: Human Factors in Computing Systems*, 1989, ACM: New York, pp. 247-252.

[mcr90] Mackinlay, J., Card, S. & Robertson, G. A semantic analysis of the design space of input devices. In: *Human-Computer Interaction*, **5** (2&3), pp. 145-190, 1990.

[mb94] May, J. & Barnard, P.J. A cognitive task analysis of the CERD exemplar material. Amodeus-2 Technical Report UM/WP23, 1994.

[mb95] May, J. & Barnard, P.J. Cinematography and user interface design. In *Proceedings INTERACT'95*. To appear.

[msb95] May, J., Scott, S. & Barnard, P.J. Structuring displays: a psychological guide. Eurgographics Tutorial Notes Series, EACG, 1995 (in press).

[m77] McCall, J. Factors in Software Quality; General Electric Eds, 1977.

[mm94]. McKerlie, D. and MacLean, A. Reasoning with Design Rationale: Practical Experience with Design Space Analysis. *Design Studies*, 15, 2, 1994, pp. 214-226.

[nc95] Nigay, L. & Coutaz, J. A generic platform for addressing themultimodal challenge. Proceedings of CHI'95. ACM Press, 1995.

[ncs93] Nigay, L. Coutaz, J. & Salber, D. A Multimodal Airline Travel Information System. Amodeus-2 Technical Report SM/WP10. 1993.

[pf92] Paterno', F. and Faconti, G. On the Use of LOTOS to Describe Graphical Interaction. In *People and Computers VII, Proc HCI'92*. A. Monk, D. Diaper and M.D. Harrison (Eds). Cambridge University Press. pp 155-173, 1992.

[p93] Paterno', F. Definition of Properties of User Interfaces Using Action-Based Temporal Logic. *Proceedings of the Fifth International Conference on Software Engineering and Knowledge Engineering*, pages 314-318. 1993.

[pm94] Paterno', F. Mezzanotte, M. Analysing Matis by Interactors and ACTL. Amodeus-2 Technical Report SM/WP36. September, 1994.

[pm95] Paterno', F. Mezzanotte, M. Formal analysis of user and system interactions in the CERD case study. Amodeus-2 Technical Report SM/WP48. February, 1995.

[rb94] Ramsay, J. and Bernsen, N.O. AMODEUS II Shared Modelling Exercise - DSD Modelling Technique Report on the CERD Exemplar. Amodeus-2 Technical Report ID/WP33, 1994.

[rb95] Ramsay, J. & Bernsen, N.O. Traceability Support for Modelling in the Design Process. Amodeus-2 Technical Report ID/WP45.

[r92]. Ryan, M. An Interactive Tool for the Construction, Modification and Browsing of Design Rationales of Smalltalk Programs. *Masters Thesis* Queen Mary and Westfield College, University of London, 1992.

[rfm91] Ryan, M., Fiadeiro, J. & Maibaum, T. Sharing Actions and Attributes in Modal Action Logic. in *Theoretical Aspects of Computer Software.* T. Ito and A.R. Meyer, Eds. Volume 526 of Lecture Notes in Computer Science. Springer-Verlag, 1991.

[snc94] Salber, D., Nigay, L. & Coutaz, J. Extending the scope of PAC-Amodeus to cooperative systems. Position paper at the CSCW'94 workshop on Software architectures for cooperative systems. Also published as Amodeus-2 Technical Report SM/WP45.

[s94] Sjöberg, C. Argumentative Design (aRD): Visualisation and support of design as an argumentative process. Amodeus-2 Technical Report ID/WP23. 1994.

[s95] Sjoberg. C., & Timpka. T. Inside multi-disciplinary design in medical informatics: experiences from the use of an argumentative design method. Accepted to the tri-annual World Conference in Medical Informatics, MEDINFO'95, Vancouver.

[s91]. Shum, S. Cognitive Dimensions of Design Rationale. In *People and Computers VI: Proceedings of HCI'91,* Diaper, D. and Hammond, N.V., (Ed.), Cambridge University Press: Cambridge, 1991, pp. 331-344.

[s92] Spivey, J.M. The Z Notation - A Reference Manual. Prentice-Hall. Second Ed, 1992.

[v92] Vernant, D. Approche actionelle et modèle projectif du dialogue informatif. *Du Dialogue: Recherches sur la philosophie et le langage*, No. 14, 301-319, 1992.

[v94] Verjans, S. Computer semiotics as a basis for integrating different views on Human Computer Interaction. *Esprit BRA7040 AMODEUS Working Paper TM/WP10, 1994.*

[ya94] Young, R M. & Abowd, G D. Multi-perspective Modelling of Interface Design Issues: Undo in a Collaborative Editor. In *People and Computers IX: Proceedings of HCI'94.* G. Cockton, S. W. Draper & G. R. S. Weir (Eds), Pages 249-260, Cambridge University Press, 1994.

[ybd+94] Young, R.M., Blandford, A.E., Coutaz, J., Duke, D.J., & May, J. Collational Co-Modelling: An Introduction and an Example from the CERD Interface. Amodeus-2 Technical Report ID/WP36. 1994.

[ygs89] Young, R.M., Green, T.R.G. & Simon, T. Programmable User Models for Predictive Evaluation of Interface Designs. In K. Bice & C. Lewis (Eds), *Proceedings of CHI'89: Human Factors in Computing Systems*, pages 15-19. ACM Press, 1989.

[z89] Zadeh L.A. Knowledge Representation in Fuzzy Logic, IEEE Transactions on Knowledge and Data Engng 1 no 1, pp. 89-100, 1989.