The AMODEUS Project
ESPRIT Basic Research Action 7040

Delivering HCI Modelling to Designers: A Framework, and Case Study of Cognitive Modelling

Simon Buckingham Shum and Nick Hammond

July, 1993—revised 14 April, 1994

Amodeus Project Document: TA/WP13

***Interacting with Computers*, 6 (3), 311-341 (1994).**

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)
Dept. 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)

Interacting with Computers, 6 (3), 311-341 (1994).

Delivering HCI Modelling to Designers: A Framework, and Case Study of Cognitive Modelling

Simon Buckingham Shum and Nick Hammond

Human-Computer Interaction Group
Department of Psychology
University of York
York, YO1 5DD
UK
E-mail. <SJBS1, NVH1>@UNIX.YORK.AC.UK
Tel. 0904-433165 or 433154
Fax. 0904-433188 or 433181

Abstract

The human-computer interaction (HCI) research community is generating a large number of usability-oriented models and design frameworks. However, a critical factor which will determine whether any of these achieve significant penetration into the real world of software design is the effort required by practitioners to understand and apply them. In short, analytic tools for usability design must themselves be usable. In response to this challenge, we present a framework which identifies four different 'gulfs' between user-centred modelling and design approaches, and their intended users. These gulfs are potential opportunities to support designers if a given analytic approach can be encapsulated in appropriate forms. We then illustrate the framework's application with a concrete example. An evaluation is reported which investigates gulfs associated with an approach which uses an expert system to automate cognitive modelling for human factors designers. An early prototype was evaluated in order to assess the knowledge required to use it. The study demonstrates that whilst this tool does shield users from the complexities of the underlying modelling, they need to understand the way in which it builds its description of the task and user interface. Implications for bridging the different gulfs are then considered.

Keywords

theory-based design; cognitive modelling; evaluation; ICS

Introduction

The human-computer interaction (HCI) research community is generating a large number of usability-oriented models and design frameworks. However, a critical factor which will determine whether any of these achieve significant penetration into the real world of software design is the effort required by practitioners to understand and apply them. In short, analytic tools for usability design must themselves be usable.

In the analysis and design of human-computer interaction (HCI), three key components are the human, the computer, and the interaction which takes place. Within a project in which we are involved[1], models of all of these have been, and continue to be developed, together with usability-oriented design representations and methodologies. Our goals are to develop, relate, and communicate to designers a range of theoretical modelling and analytic techniques for HCI design.

The approaches with which we are working are modelling user cognition (Barnard, 1987; Blandford and Young, 1993; May *et al*., 1993a; Young *et al*., 1989b), user-system interaction states (Barnard and Harrison, 1989; Harrison and Barnard, 1993), formal interactive system behaviour (Dix, 1988; Duke and Harrison, 1993; Faconti and Paterno', 1990)., and multimodal user interface architectures (Coutaz, 1987; Nigay and Coutaz, 1993). Ways of interrelating these with each other are being explored using fuzzy set theory (Darzentas *et al*., 1993), and semiformal notations for design argumentation (Bellotti, 1993; Bernsen, 1993; MacLean *et al*., 1991; Timpka *et al*., 1993).

Whilst the research concerns of individual modelling groups are, understandably, primarily focused on extending the scope and complexity of HCI phenomena which their models can address, and reducing the cost in time and effort to perform modelling, our particular concern is with the usability of these approaches. Developers of abstract design representations, such as those listed above, should be thinking from the start about the eventual users of their techniques, just as the HCI community urges designers to 'involve the user' from initial conception of the software artifact. Table 1 summarises our position, making the case that principles holding for user-centred design in general should also hold for 'designer-centred' analytic HCI tools in particular; the principle of designing effective, usable tools and training for expert practitioners is just as important for HCI designers as for other kinds of end-users. The modelling questions in Table 1 are strongly motivated by issues raised by participants at a modeller-designer workshop (Shum and Hammond, 1993) held early in our work to assess the 'lay-of-the-land.'

---

1 The AMODEUS-2 project – "Assaying Means Of Design Expression for Users and Systems." (AMODEUS-2, 1992) . This follows the first AMODEUS project (Hammond *et al*., 1991; Young *et al*., 1989a).

User-Centred Technology Design	**Designer-Centred Model Design**
• Are the requirements for technological intervention understood?	• **What kinds of input do designers want or need from HCI modellers?**
• Do prospective users understand what the technology is capable of?	• **How can we communicate what different models do?**
• Similarly, seeing two potential systems compared and contrasted illustrates their respective merits and limitations.	• **How do different models relate to each other?**
• How will current work practices be changed by introducing new technology?	• **How do different models fit into existing design practice?** • **What process is involved in using a modelling/analytic approach?**
• What training is needed to use the new technology?	• **What knowledge and expertise is needed to make effective use of a modelling approach?**
• Does the new technology improve quality and effectiveness of work?	• **Does the modelling improve the quality of usability analysis and decision making?**

Table 1: Parallel concerns between user-centred system design, and the design of designer-centred HCI modelling and design approaches.

This paper serves as a starting point from which to begin to address these questions. In the first section we present a conceptual framework for identifying opportunities for communicating HCI models and design frameworks to designers. This centres around the notion of *encapsulation gulfs* – gulfs of communication which can be bridged by appropriate re-expressions (*encapsulations*) of the process or product of a particular model or design method to meet different needs.

The gulfs framework emphasises the potential of communicating modelling at both 'upstream' and 'downstream' points in the modelling process, as opposed to only communicating once a 'full analysis' of a problem is complete. This conceptualisation highlights the different ways in which modelling/analytic representations are intended to be used by the design community, a critical factor being the level of knowledge about a modelling approach which is required in order to apply it effectively in design. The framework both motivates and sets the context for the results of a study which we report in the second part of the paper. This explores in detail the gulfs to be bridged in encapsulating a cognitive modelling approach for human factors designers to utilise.

'Encapsulation': getting modellers and designers talking

It has been reported (Bellotti, 1988) that designers do not use theoretical HCI modelling techniques in practice. A broad characterisation of the differences between HCI practitioners and researchers might begin with their *work* (design vs. science), the *goals* thereof (finding good-enough vs. optimal solutions), and hence their different *training and conceptual language* (concrete orientation, e.g. artifacts and scenarios vs. abstract orientation, e.g. issues and tasks). The conceptual (and hence representational) apparatus which HCI researchers bring to bear on a problem are thus often very different from those of typical user interface designers in the field. However, if we believe that theory still has something to offer HCI design practice, albeit not based on a simplistic model of directly applying core science practice (e.g. experimental psychology), then we must explore what we shall refer to as *encapsulations* of analytically and theoretically based approaches which can be used by design teams.

It is this gulf of communication which our current work aims to narrow, in the context of communicating and assessing the contribution of HCI modelling to software designers. We use the term *encapsulation* (as a process) to mean:

> Re-expression of the representations with which HCI researchers work, in order that members of the design community can comprehend and apply the results of their analyses, or go on to use the techniques themselves.
>
> Thus, *an encapsulation*, the product of the encapsulation process, will comprise some subset of the modelling[2] process or product, highlighting particular features of interest to designers.

In other words, encapsulations are a point of contact between modelling and design – designers are meant to ‘use’ them in some way. The nature of this usage will vary depending on the approach. An encapsulation may embody elements of the analytic *process* of a particular approach, or the *results* of that approach. The framework introduced below conceptualises the need to communicate something between modeller and designer as a *gulf,* which can be bridged by one or more appropriate encapsulations of some aspect of the modelling (Figure 1). The focus on encapsulation takes to heart earlier work emphasising the need for *application representations* to bridge from theory to practice (Long and Dowell, 1989).

---

[2] Not all of the groups in AMODEUS-2 can be described as ‘modelling’ – some such as Design Space Development (Bernsen, 1993), QOC Design Space Analysis (MacLean *et al*., 1991), and Action Design (Timpka *et al*., 1993) provide design frameworks or methodologies for designers to reason with or follow. The term ‘analytic approaches’ will be used to cover both modelling and design frameworks.

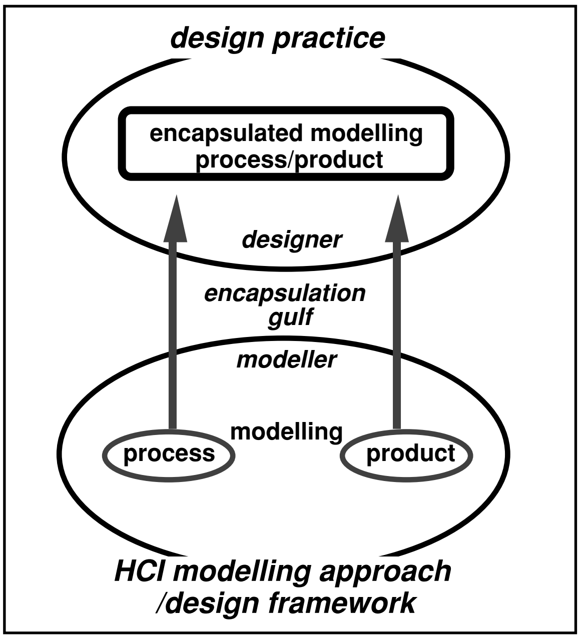
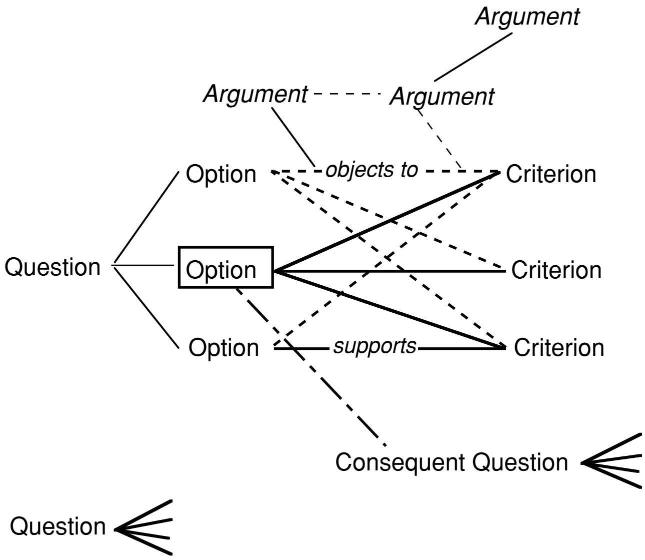


Figure 1: A good encapsulation embodies elements of modelling process or product in a form which can be comprehended and applied by designers.

Let us illustrate the encapsulation concept with two brief examples. One of the cognitive modelling approaches with which we are working is *Cognitive Task Analysis* (CTA), based on *Interacting Cognitive Subsystems*, or ICS (Barnard, 1985; 1987). ICS is an architecture which supports approximate modelling of the cognitive resources and processing contraints in performing a task. It has been demonstrated that this modelling can be automated within an expert system shell (May *et al*., 1993a). The system takes a description of a user interface, users, and tasks, builds a model describing which cognitive resources will be used and coordinated, and then maps this internal state onto report rules predicting key aspects of user behaviour and potential usability problems. This expert system modeller is the primary *encapsulation* of the ICS modelling approach, embodying both *ICS modelling process* (mapping from the real world, to a cognitive model, and back to the real world), and *ICS modelling product* (the modelling reports). We return to this example in the study reported later.

An example of work which is not a 'modelling approach' but rather a representation with which designers can reason, is a notation and method for expressing *design rationale* (DR) (Carroll and Moran, 1991). A DR is a representation of why a design is the way it is – the reasoning behind the decision to implement something in a particular way. The particular approach to DR which we are studying is *Design Space Analysis* using the *QOC* (Questions/Options/Criteria) notation (MacLean *et al*., 1991). Design Space Analysis is a process of identifying key design issues (expressed as *Questions*), *Options* providing possible answers to those Questions, and Criteria for assessing and comparing the Options (Figure 2). Thus, a QOC network clarifies the local design space around different aspects of a design, in order to gain a fuller understanding of the issues, and showing why potential Options have been rejected. This serves as an ongoing rationale which can be subsequently inspected.



Question . . .

Options ***Criteria***	Option	Decision	Option
Criterion	– *Argument... Argument...*	+	–
Criterion	–	+ *Argument...*	?
Criterion	–	+	+

→ **Question . . .**

Figure 2: The QOC notation for representing design rationale, in graphical and matrix formats.

One of QOC's roles in AMODEUS is as a form of encapsulation for *modellers*. A QOC design space summarises many different modelling approaches within a common, semiformal notational framework; it hides the complexity of the respective modelling processes, showing only their products as they relate to key issues (Questions) of concern (e.g. Bellotti, 1993). However, such a QOC design space can also serve as an encapsulation delivered to *designers*, which makes it potentially very useful—if it can be demonstrated that designers can work with QOC analyses, they can serve as a uniform representation through which modellers and developers can communicate. Detailed analyses of QOC in use by designers (e.g. Buckingham Shum, 1994), and of similar 'argumentative' approaches (Buckingham Shum and Hammond, 1994) suggests that this kind of representation has realistic potential, although there are still many issues to resolve.

It should now be clear that ICS and QOC are very different in nature, and the *ways* in which they are intended to be used in design vary correspondingly. However, which aspects of a given analytic approach can be encapsulated? How does each approach map to design practice? The concept of 'encapsulation gulfs', briefly introduced above, is now elaborated, providing a framework for our encapsulation work.

Identifying the gulfs to be bridged

The developers of any analytic approach make a set of assumptions about who will be able use their techniques. The extent to which this is explicit, or has been assessed in any way, varies widely. Our prime concern is to examine more closely what points of contact exist (or could potentially exist) between a given approach, and design practice.

Each modelling approach targets a particular class of end-user ('designer'), in terms of their background and design expertise (e.g. software engineering, computer science, human factors). These individuals or teams work within an existing design practice and culture, employing its own methods and tools. Similarly, each of the modelling approaches utilises its own procedures and representations to inform design decisions. Given these two processes, we are concerned to assess the potential for assisting HCI practitioners by making modelling insights available in appropriate forms.

If one imagines a designer and a modeller working on the same HCI problem, their analysis processes can be represented as two activity streams (Figure 3), separated by an encapsulation gulf which represents the potential for communication.

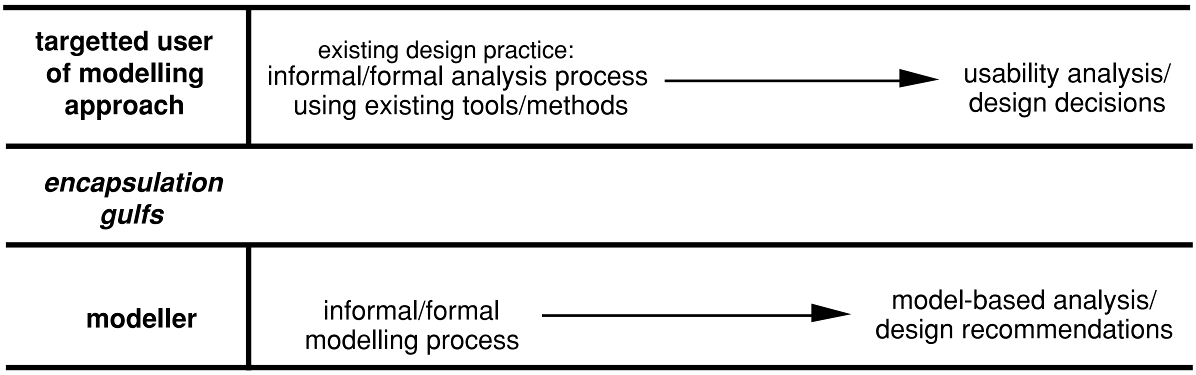


Figure 3: The structure of the encapsulation gulfs framework, showing two design processes: that of existing design practice, and that of HCI modellers. The potential for, but as yet non-existent, communication between the two parties is conceived as a set of dividing gulfs which can be bridged by appropriate encapsulations.

This template is elaborated in Figure 4, which articulates more clearly the potential for encapsulating a modelling approach.

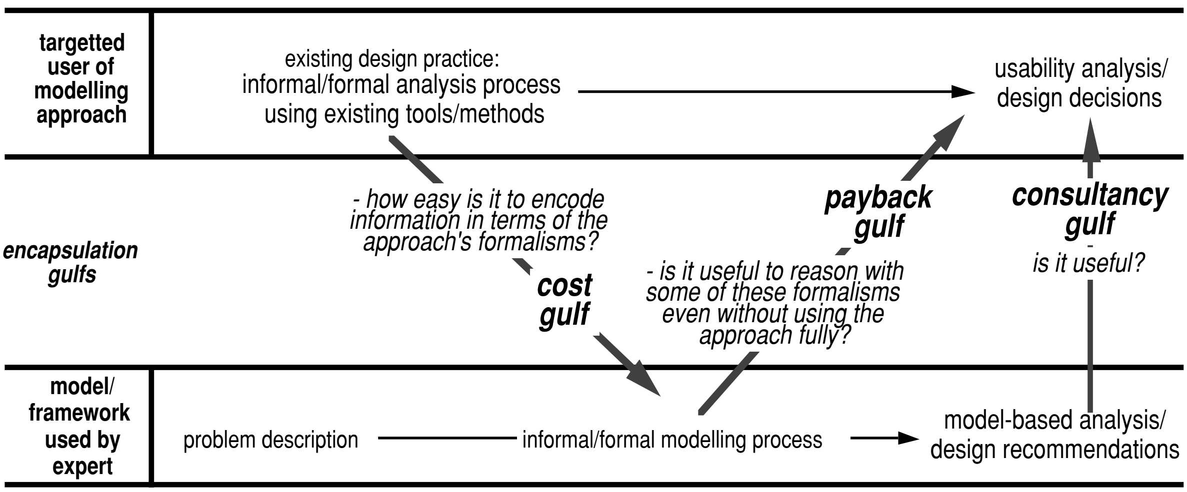


Figure 4: Three key gulfs between designers and modellers. To make elements of the analytic process available to designers, the **cost gulf** *and* **payback gulf** *must be bridged. To communicate the product of a modelling analysis, the* **consultancy gulf** *must be bridged.*

The most commonly envisaged way of assisting designers is via the **consultancy gulf**, so called because the modeller works in the role of an expert 'consultant' who delivers a detailed modelling analysis of the problem. In validating HCI models, one normally thinks of this gulf, asking whether the result of the modelling (e.g. a GOMS analysis) is useful. “Useful” here should mean that the modelling delivered (the encapsulation) must be:

- *intelligible* – can the modelling be delivered in a form which is understood?
- *relevant* – does the modelling address needs, or introduce new useful ideas to the designer?
- *applicable* – can the modelling be delivered in a form which can be acted upon?

There are, however, additional gulfs—although at present new HCI analytic techniques are used by a few expert ‘consultants,’ the long term goal is for mainstream practitioners to be able to use them. In that case, how easy will this be, and for what benefit? The **cost gulf** and **payback gulf** associated with using a representation or method raise these issues. The cost gulf asks about the demands introduced in using these formalisms, and the payback gulf about the potential support for design reasoning which this might provide, independent of any benefit which might accrue from a complete, formal modelling analysis (consultancy gulf). This pair of ‘down-up’ arrows captures the two-way communication or ‘dialectic’ which a good design representation or method should engender; the effort of encoding ideas within a formalism, or of following a particular method, should aid in the formulation of, and reflection on, those ideas.

Modellers often argue that the modelling *process* (the discipline of using a representation) is as valuable for gaining insights as the final *result*. This is an intriguing claim, which requires careful study before it can be substantiated. Within the gulfs framework (specifically the **cost/payback** gulfs), it raises the interesting possibility that for a given analytic technique, there may be elements which could be encapsulated as ‘tools for thought’ by designers; that is, as ‘stand-alone’ techniques independent of the approach which originally motivated it. An example of this may be work on Programmable User Models (Young *et al.*, 1989), in which a step called *Knowledge Analysis* (e.g. Young *et al.*, 1990), which focuses attention on whether the user will know *what* to do next, and *how*, could potentially be conducted by an interface designer without understanding the Soar-based cognitive modelling in support of which the technique was originally developed. We hope to test this hypothesis, in order to examine more closely what, if any, dependency there is between Soar expertise and effective Knowledge Analysis; for instance, are there systematic differences between Soar and non-Soar experts’ analyses?

Finally, the framework makes explicit that the long term goal for some approaches is that the point of contact with designers will be through a software tool which automates the modelling (e.g. a Programmable User Model, or an expert system as described in this paper). Consequently, an additional modelling stream appears for these approaches (Figure 5), to represent the gulfs associated with a tool intended for use by a particular kind of designer.

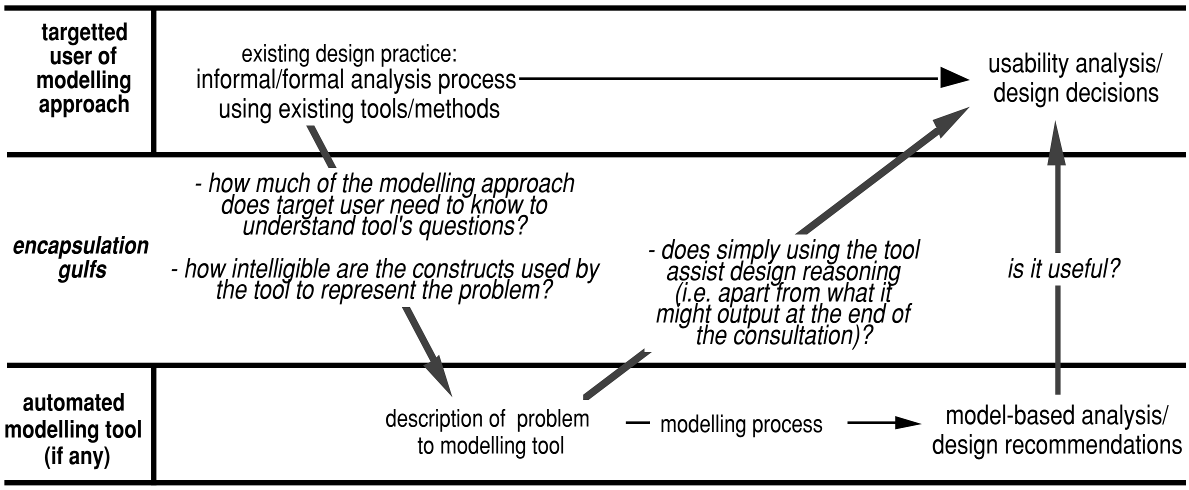


Figure 5: Some modelling approaches have automated modelling tools, which introduce a new gulf to be bridged.

The ‘reflective process’ described in relation to Figure 4 can be related most obviously to a design representation like QOC, which provides a notations for a designer to directly reason with. However, Figure 5 highlights equally the potential of a modelling tool as a cognitive artifact: the claim here is that the process of using such a tool can change the way in which one conceptualises a problem.

The ‘prerequisite gulf’: establishing common ground

A gulf which must be bridged, but which has remained implicit thus far, is the ease of communicating to designers the purpose and (at a general level) the underlying principles, of a given approach. Successfully bridging this gulf is critical to establishing sufficient common ground for any subsequent interaction to be successful, in the same way that two people cannot effectively interact without establishing a shared understanding of relevant concepts. There is also the issue of the extent to which designers will trust any of the other forms in which they encounter the approach if this first gulf is ignored.

“Successfully bridging this gulf” means not only communicating sufficient knowledge, but doing so on a realistic timescale, given the constraints under which professionals normally undergo extra training (from an afternoon, to a few days’ course). A common feature to each gulf analysis, therefore, will be the “prerequisite background gulf.” Thus, the complete encapsulation gulfs framework is as shown in Figure 6.

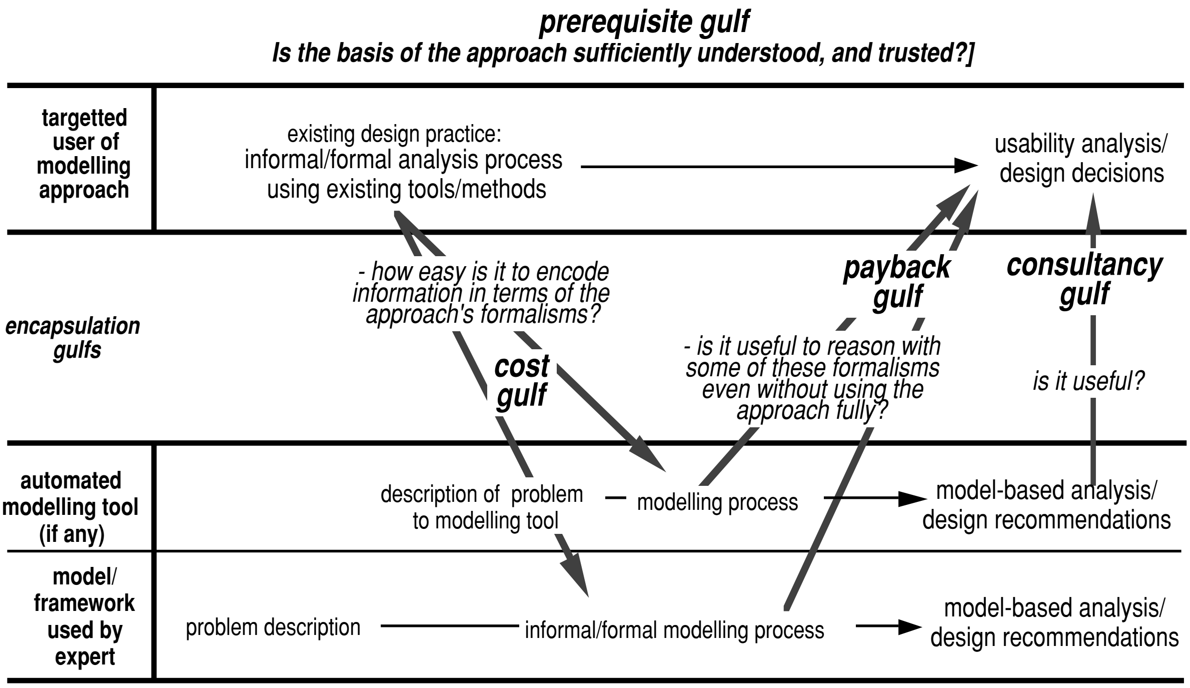


Figure 6: The complete encapsulation gulfs framework, which can be used to analyse the potential for communication between an HCI modelling/analytic approach, and design.

As we investigate encapsulation requirements further, we will gain an understanding of how much a designer needs to know about an approach, and in what forms this knowledge seems to be best presented. Note that according to this gulf analysis, it is possible, in principle, that the basis or rationale of an approach might prove so hard to communicate that attempts to evaluate other encapsulations are subsequently undermined by practitioners' incomprehension, or scepticism. We are currently evaluating one way of encapsulating this kind of 'prerequisite knowledge' through the use of executive summaries and short worked examples (Buckingham Shum *et al.*, 1994a). These are available for evaluation by interested members of the HCI community, as well as serving as background for designers who are collaborating with AMODEUS so that they can better understand the modelling analyses they receive.

Organisational gulfs

Within the gulfs framework presented thus far, encapsulations have been assessed in terms of compatibility with 'design practice' in quite a restricted sense. The scope of analysis has treated design as essentially a cognitive activity, asking questions such as, *Can designers understand the purpose of a modelling approach? How easy is it to encode ideas within a formalism? How does a representation assist design reasoning? Is the modelling intelligible?* Thus, at the cognitive level, an encapsulation will be 'good' if it is representationally *sufficient* (expressing the modelling in appropriate scope and detail), and *cognitively compatible* (e.g. making important information salient).

Critical as these requirements are, satisfying them does not guarantee that an encapsulation will successfully bridge the gulf. Other obstacles need to be seriously considered if the complexity of real design settings is to be acknowledged. These obstacles might be grouped under the heading of 'organisational inertia to change.'

The *context* in which the gulfs analysis sits is characterised by questions such as *How compatible are HCI modelling encapsulations and associated work practices with current representations and practice?* and *What levels (individual/group/project/organisation) will the introduction of a particular encapsulation impact?* Within our own work, we anticipate that it will be impossible to ignore the wider context into which modelling encapsulations will be introduced – indeed, such factors may prove to be significant obstacles to their introduction at all; rather than merely widening the gulfs, these obstacles may prevent them from ever being assessed. Such considerations add weight to the importance of shaping modelling approaches—even as they are being developed—to the characteristics of the environments in which they will be used.

Using the framework to motivate empirical studies

Gulf analyses have been conducted for eight different HCI analytic approaches being developed within the AMODEUS-2 project (Shum, 1993), providing a starting point for evaluating their usability. The gulfs framework motivates several different kinds of investigation, depending on what is being offered to designers by an approach. Two clear categories of gulf can be identified—*process gulfs* and *product gulfs*—which require different methods of investigation:

- **Modelling Product—prerequisite** and **consultancy gulfs** raise the question: *is the modelling analysis comprehensible and useful to designers?* This modelling gulf is being validated by providing 'modelling consultancy' to external design teams, and then documenting designers' reactions at modeller-designer workshops. One exercise of this sort has been completed so far (Buckingham Shum *et al.*, 1994b).

- **Modelling Process—cost** and **payback gulfs** raise the question: *are there any aspects of the analytic process used by experts, which practitioners might find useful?* Encapsulations which fall into this category are evaluated by detailed study of the costs and benefits of systematising one's thinking through the use of a notation or method of some sort. One example of this are studies of design rationale notations in use (e.g. Buckingham Shum *et al.*, 1993). Another is analysis of the ICS cognitive modelling expert system in use, described next.

Evaluating the ICS expert system as a vehicle for encapsulation

The remainder of this paper reports an evaluation of an expert system modeller as a vehicle for encapsulating ICS modelling in a form appropriate for human factors experts (but not modellers) in design practice. This approach is a good example of the encapsulation concept, since it places considerable emphasis on the possibility of delivering complex cognitive modelling in a form appropriate for non-modellers. Let us begin by considering a gulfs analysis of the modelling approach (Figure 7).

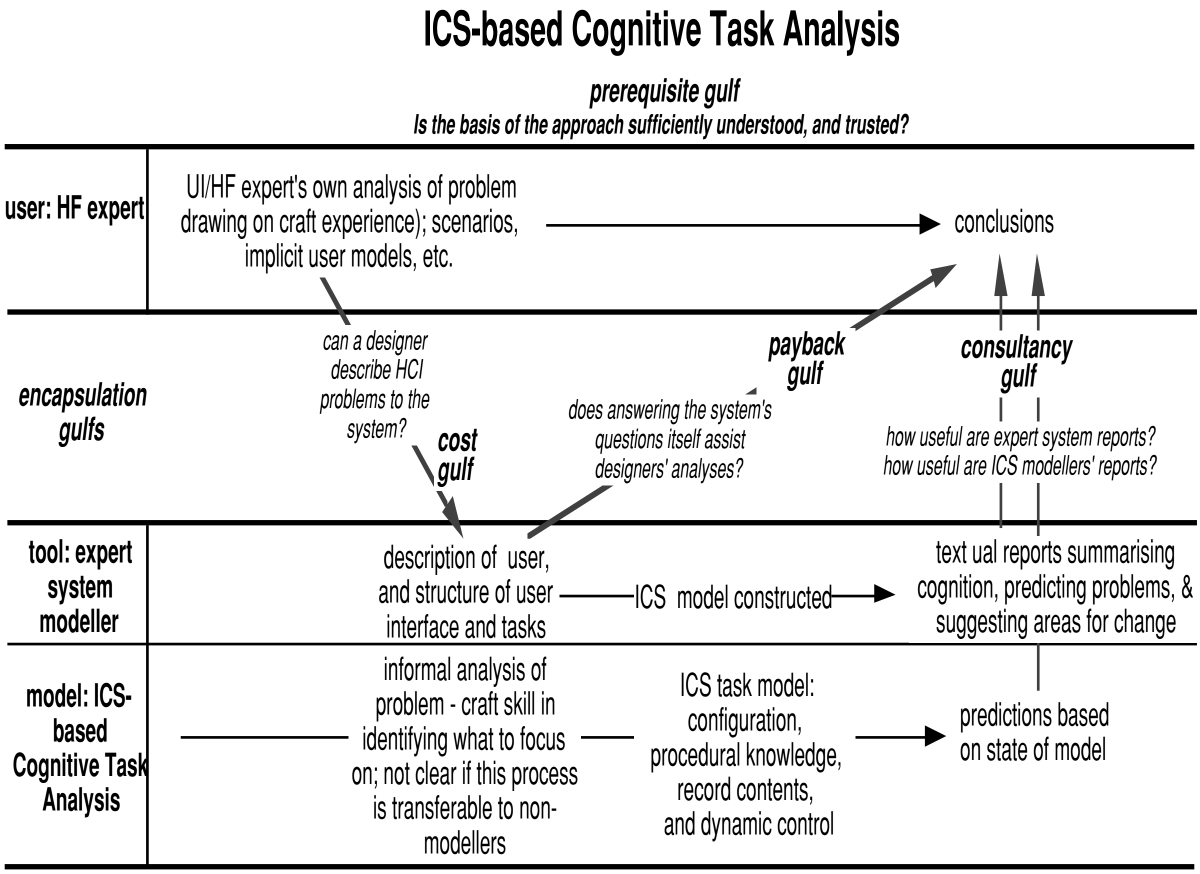


Figure 7: Gulf analysis of 'Cognitive Task Analysis' *based on the* 'Interacting Cognitive Subsystems' *architecture.*

The **targetted user** of the ICS modelling tool is a human factors expert with some training in the use of the system. The evaluation of the ICS expert system's usability reported here focuses primarily on the communication of background knowledge about ICS (**prerequisite gulf**), and on the **cost gulf**: can non-modellers with a human factors background successfully provide the tool with the information it requests about a problem? How can this gulf be more effectively bridged? To a lesser extent, the results address the **payback** and **consultancy** gulfs, which concern potential benefits to designers' understanding of the problem which accrue from answering the tool's questions.

Thus, the gulfs framework motivates (and sets in context with other possible studies) an investigation of the question, 'To what extent does the existing demonstrator version of the ICS expert system achieve the developers' long-term goal of producing a tool which does not require the user to know about the underlying modelling approach?' This concern can be focused by asking, 'What knowledge is needed to use the tool?' This issue must be confronted at some point in the tool's development, and we argue that it is both desirable, and possible, to conduct early evaluations of a proof-of-concept prototype in order to inform this critical issue. However, it is important that we restrict our evaluation to the *knowledge demands on users* which are inherent to the way in which the tool is being developed, and are not distracted by surface usability problems particular to the environment in which the early prototype was running (in fact, the system is now being re-implemented on a different platform).[3] Targetting the knowledge demands on users is also the first step towards answering the question, 'What *kind* of training will a human factors expert need to use the tool?'

In order to understand the results, it helps to have a basic grasp of the relationship between 'manual' ICS modelling and the expert system's 'automated' modelling, outlined next.

The relationship between ICS modelling and the expert system modeller

The ICS architecture and its application to HCI is presented in detail elsewhere (Barnard, 1985; 1987; 1991), and its automation as an expert system in Barnard *et al.* (1987; 1988) and May *et al.* (1993) . We shall summarise here only certain aspects of the modelling approach.

Firstly, ICS builds up a four-component model based on:

- **which cognitive transformations** (i.e., from one subsystem's representation to another) are involved in performing the task (representations can be transformed from raw acoustic and visual, to more structured linguistic and visuospatial, to 'higher order' propositional and schematic);
- to what degree these parallel transformations between subsystems are **proceduralised;**
- the usefulness for these transformations of the **memory records** availability to each subsystem;
- the subsystem(s) whose processing places the most **constraints** upon the smooth flow of information through the configuration of transformations.

'Families' of such models may be developed in a complete analysis, by instantiating the above components for each activity phase.

A proof-of-concept expert system has been developed, which demonstrates that the ICS modelling process can be automated. This prototype, written in an off-the-shelf expert system shell, consists of around 400 production rules, three-quarters of which control the operation of the system, presentation of questions, and generation of output. The remaining hundred rules perform the modelling. The way in which the expert system automates ICS modelling is shown in Figure 8.

Two key concepts need to be introduced at this point: *phases of user activity*, and *structural representations* of tasks and displays.

Firstly, Hutchins *et al.* (1986) have presented the concept of seven 'stages of action' in human-computer interaction. Of these, ICS-based Cognitive Task Analysis draws on the three stages for communicating with the system, termed *Goal Formation, Action Specification* and *Action Execution*, as a way of structuring its analysis of cognition during the performance of a task. The phases capture certain changes in cognition which will occur in performing the task, corresponding to deciding *what* to do (goal), *how* to do it (in terms of which commands do I need), and actually *doing* it (e.g. through a specific typing or mousing procedure). These shifts may occur very rapidly, in an opportunistic manner.

---

[3] However, some user interface refinements were made to the basic prototype, described shortly.

Secondly, Cognitive Task Analysis, the process of modelling the world using ICS, uses a method of *structural description* which can be used to represent both tasks and displays. May *et al*. (1993) introduce this representational technique as follows:

> This ... representation is based upon the structure of the mental information that must be transformed by the user's cognitive resources in the interaction with the interface. It is the ease with which these structures can be parsed and transformed into other mental representations that determines the ease or complexity of the cognition that is required to use the design which is being examined. Furthermore, since it is the structure which is important, and not the content of the structure, this approach to deriving the acquisition representation means that it can be used across a wide range of design questions. (p.29)

Figure 9 shows an example of how different kinds of interfaces can be represented structurally.

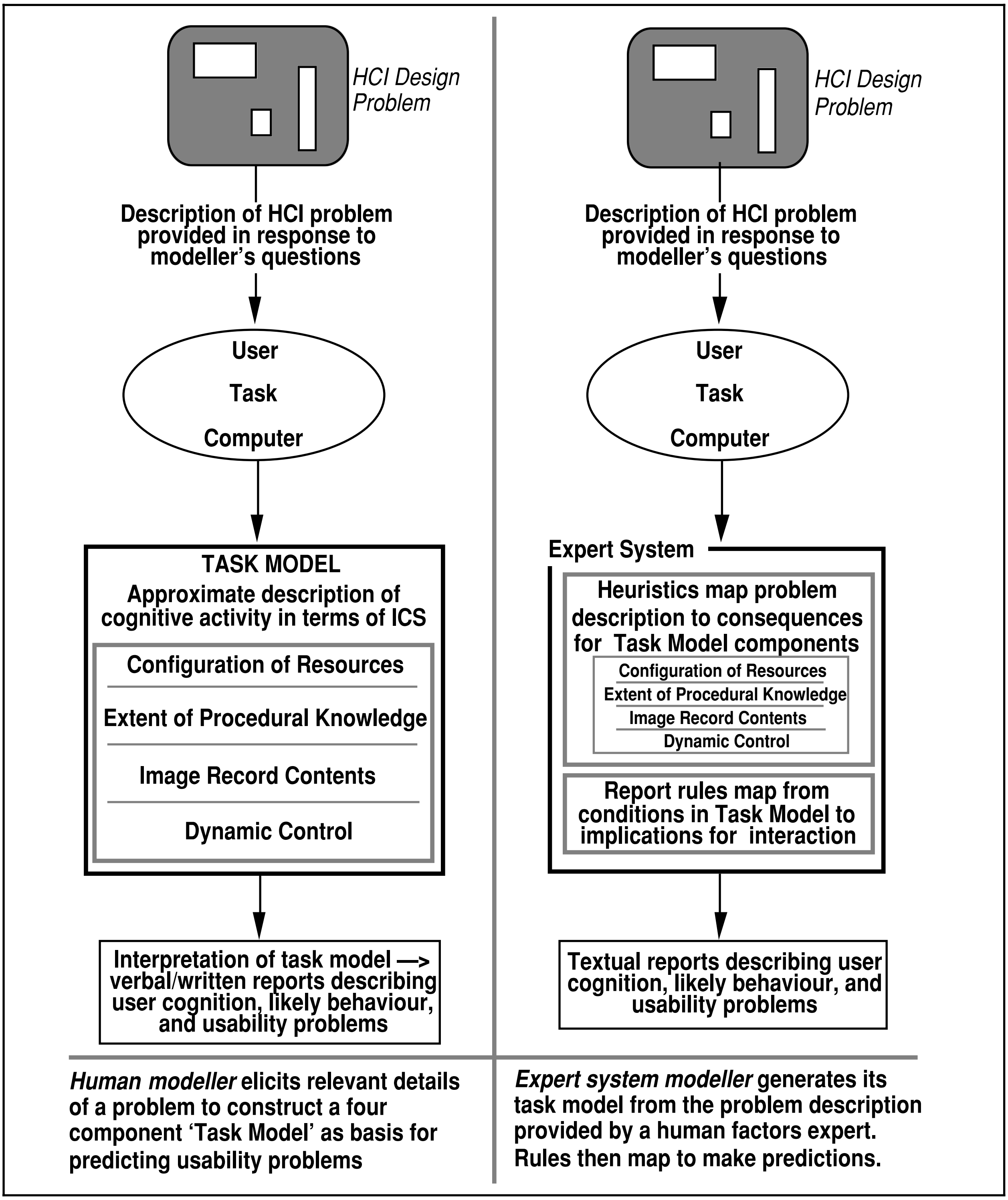
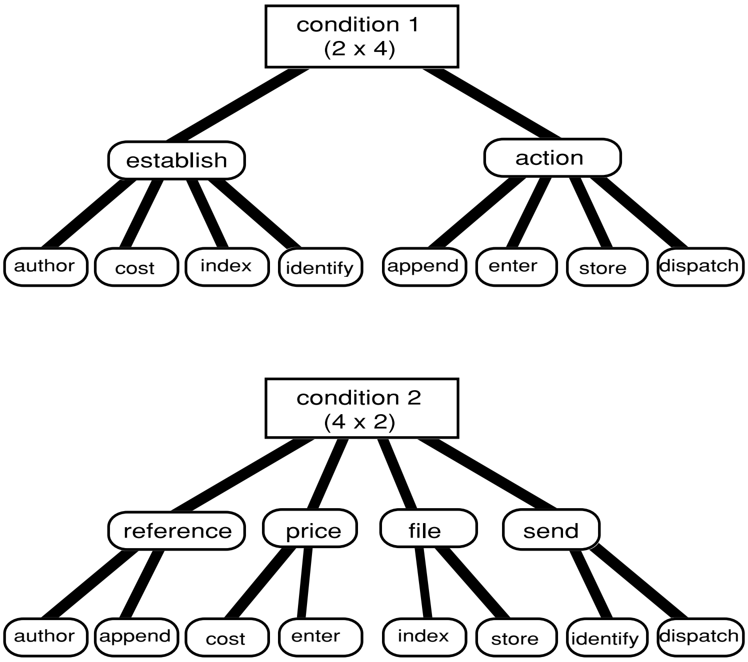
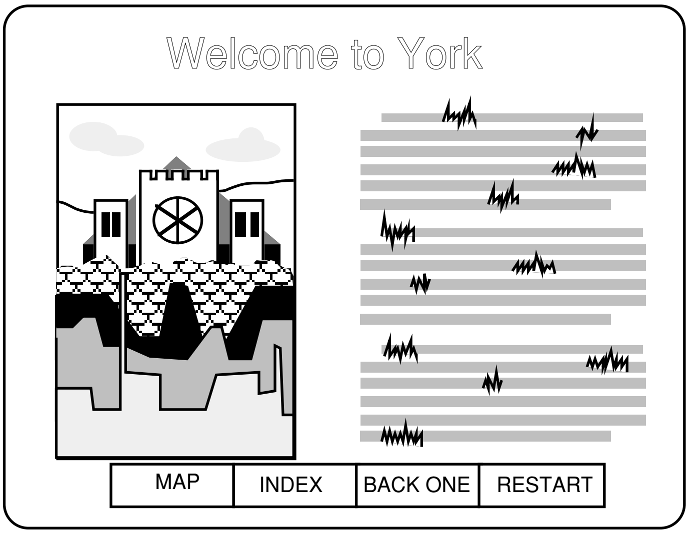


Figure 8: ICS-based Cognitive Task Analysis, as performed by a human modeller, and as performed by the ICS expert system modeller.



(a) Command structures from two conditions used in an e-mail task (Barnard *et al.*, 1984). *Commands were grouped into two groups of four, or four groups of two.*



hypertext screen	***consists_of***	**title : picture : text : buttons**
buttons	***consists_of***	**map : index : back one : restart**

(b) A hypertext screen, and part of a structural representation (May *et al.*, 1993a).

Figure 9: Examples of how both command and visual structures can be represented using the same method of structural representation. The visual structure of individual icons has also been analysed using this technique (May et al., 1993b).

The first step shown in Figure 8 for acquiring a description of the HCI problem is now shown in more detail in Figure 10, clarifying the role of the structural representation.

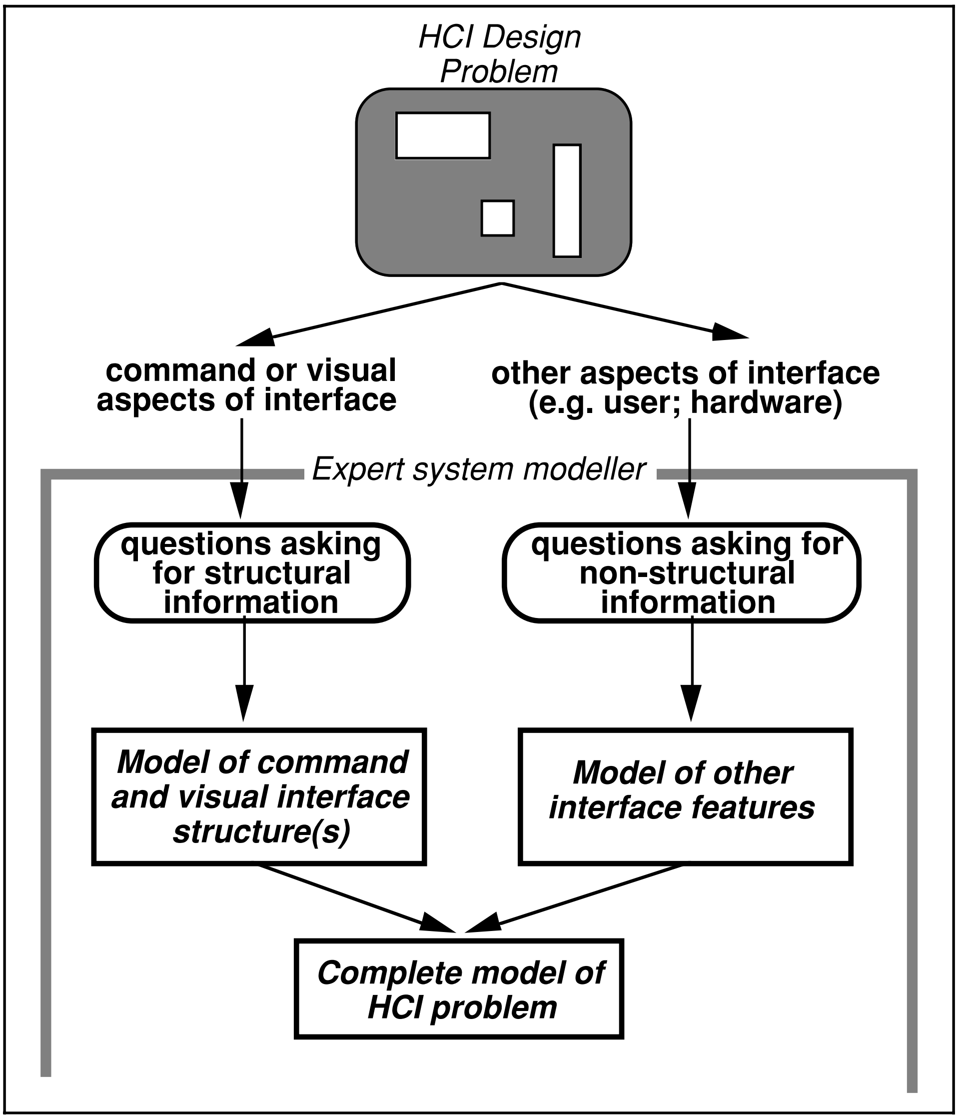


Figure 10: The tool acquires its representation of a user interface's command and visual features in terms of structural relationships. (Other aspects are non-structural).

To summarise, the primary goal of this particular modelling encapsulation is that a human factors expert has to answer questions about users, tasks, and the user interface—matters on which he or she already has considerable expertise—and is not drawn into the modelling. As described, the tool's questions elicit information from the user in order to construct representations of the user interface. The effectiveness of the user-system dialogue in this context should be evaluated in terms of whether the user understands what information is being asked for, and is able to select from the options offered by the system the one which most closely describes the problem in hand.

Evaluation method

Subjects

Twelve Ergonomics Masters students specialising in HCI were recruited. Whilst these users had relatively little design experience compared to the professional human factors experts at which the modelling tool is aimed, it was considered that their ability (i) to understand the issues at stake in the experimental design problem, and (ii) to interpret the expert system's questions, was sufficiently good to yield useful, generalisable data. This also took into account experiences gained from an earlier evaluation of the expert system (Shum, 1988), in which there was broad overlap in the usability problems encountered by students (on the same Ergonomics course as the present subjects), and more experienced HCI researchers.

Experimental task

Working in pairs, subjects were asked to use the expert system to evaluate aspects of a user interface. The interface features of interest were the organisation of *Move, Copy,* and *Delete* commands in a direct manipulation text editor (a real application). In this editor, *Delete* was one of a row of buttons on a permanently displayed menubar, whilst *Copy* and *Move* were located on a pop-up menu.

Subjects were asked to describe this design in the imaginary context of comparing it with another design (from a later version of the same editor) in which all three commands were together on the pop-up menu. The introduction of an alternative design to the experimental scenario meant that subjects now had to understand which phases of cognitive activity they needed to analyse to contrast the two designs. (For example, in this scenario, the *Goal Formation* phase was identical for the two designs, and so modelling it would not have yielded analyses of their differences.)

Expert system version

The version of the expert system modeller which was evaluated is built within the *Xi Plus* expert system shell (Inference, 1991) running on an IBM compatible PC. However, it was judged that before it would be possible to gain insights into issues such as the level of knowledge required, the basic quality of user-system interaction had to be improved so that relatively trivial usability problems did not obscure other phenomena. A version of the system was therefore developed for this evaluation, tailoring the user interface in two ways.

As described above, many questions asked by the tool are based on a structural analysis of the user interface. Whilst this supports a powerful form of generalisation for cognitive modelling, in the current proof-of-concept demonstrator, these questions are expressed in generic, domain-independent language, which can be difficult to contextualise to the domain of concern. An example of a generic question, which attempts to establish the distinctiveness between items at the same level in the structural hierarchy, is:

> `How distinguishable from the other superordinate groupings is the one that` *`Delete`* `belongs to?`

For the purposes of this study, the questions and Help text for each question were tailored to the WIMP interface paradigm in which the experimental HCI problem occurred, in order, it was hoped, to pre-empt some of the more obvious sources of incomprehension during the evaluation. Thus, the above question now read:

How distinguishable from the other command groups is *menubar*, the one that *Delete* belongs to?

(Note the addition of the command group's name to improve the question's readability).

Secondly, in the introductory text and Help screens, additional emphasis was placed on the importance of the three cognitive activity phases (*Goal Formation, Action Specification,* and *Action Execution*) which structure ICS modelling and the expert system's dialogue. In many contexts, modelling a particular phase is irrelevant to what one wishes to know about a design, and so the expert system asks in turn if the user wishes to model each phase in detail. In order to answer this, the user must understand what the three phases mean (although whether they can decide for themselves to answer Yes or No is one indication of their level of understanding).

In order to convey the style of dialogue in which the expert system is consulted, below is an extract in which the user is describing the *Copy* command in the pop-up menu (user's responses shown in bold):

What is the name of the group that *Copy* belongs to?
pop-up

Is *Copy* fixed in sequence or position with respect to *pop-up*?
yes
no

How well does the meaning which *pop-up* comes to stand for encompass *Copy*? **better than other** groups do
as well as other
worse than other

Does the meaning or function of *Copy* bear a closer resemblance to the commands in *pop-up* (its own group) than to commands in other groups? **yes**
no

On processing *pop-up*, will users think of or see it before the other commands? **yes**
no

During use, is *pop-up* located away from users' previous focus of processing? yes
no

Subjects' introduction to the expert system

The expert system was introduced to all subject pairs as "an expert system-based human factors design tool" which required information about users, tasks and the user interface being evaluated. Subjects were shown how to select answers to questions. Three pairs were run before a one-day seminar on ICS and the expert system, presented by two of its developers, whilst the other three were run after this seminar. These conditions were introduced to take advantage of the seminar which was part of the student's course, in order to pick up gross differences which might emerge due to different levels of knowledge about ICS. However, none emerged, possible reasons for which are discussed in the results. Before the evaluation, subjects from the group who had already attended the seminar were taken through a brief reminder of what they had been taught about ICS.

Data collection

All discussion between subject pairs, and between subjects and the experimenter, was audiotaped. Subjects were asked to check with the experimenter before entering their answer to a question; this provided an opportunity for the experimenter to gain clarification about subjects' reasoning, and also enabled 'wrong' answers to be pre-empted. This avoided: (i) cumulative errors over the course of the session in which misunderstandings from one question would be carried over into subsequent reasoning, and (ii) the triggering of different questions, following wrong answers, which would result in the dialogue taking different directions for different subject pairs.

After attending both the seminar and expert system consultation, subjects completed a brief questionnaire asking about their understanding of the ICS approach from the seminar, and their experiences with the expert system.

Results

In any usability evaluation, and in all the encapsulation work of which this study forms a part, problems will arise which reflect a lack of knowledge or understanding at a variety of different levels. In assessing the data from this evaluation exercise, it is important for ongoing ICS modelling and encapsulation work, both of which are at early stages in their development, to focus on the right level of problem. As explained earlier, the emphasis is on identifying the knowledge demands placed on users by the modelling dialogue with the tool, and the long term implications for the tool's development.

Knowing which task phases need to be modelled

For each phase, the system asked the following question:

```
Do you wish to describe the commands for analysis in the current
phase?            yes

                  no
```

Although the meaning, and optional nature of the three phases for analysing activity was emphasised in the expert system's introductory and Help screens, and additionally in the 'revision sheet' which the second subject group were talked through, this concept still proved to be a source of confusion for every pair.

A human factors expert should have no difficulty in understanding these activity phases, which are relatively well known concepts. However, it is not enough to grasp these phases on their own: they must be understood in the context of providing *structural descriptions* – for each phase (goal formation; action specification; action execution), and how the *basic unit*[4] being analysed and its *grouping* changes (from *goals* and *goal groups*, to the *commands* provided in the user interface and their *command structure*, to the *particular form* of those commands, and the *visual* or *procedural* structure, depending on the modalities involved). Details can be found in May *et al*. (1993).

---

[4] A basic unit varies according to the phase being modelled; for instance, it can be a goal, a command, or mousing action.

When faced with the above question, therefore, over and above a general grasp of the concept of activity phases, users need to be able to map from the particular problem they have in mind (e.g. "where should *Delete* go?"), to the corresponding level of *basic unit* in the structural description ("this has implications for *command procedures*"), and from there infer the activity phase ("I need to conduct an analysis in the *action execution* phase").

It will be clear that to perform this series of mappings, the designer needs a good understanding of how the tool acquires its description of the interface, which the subjects clearly lacked. Moreover, the 'informative' text embedded within the tool (specifically added for the purposes of this evaluation) failed to make up for this missing knowledge. If the above analysis is correct, we conclude that this failure was largely because the Help text tried to relate the phases to the questions being asked without mentioning the structural description method used by the tool; indeed, the original goal in designing the interface had been to shield the user as far as possible from having to consider the underlying mechanisms. Implications of this finding are discussed in the final section.

Knowing how to define command groups

```
How many command groups are there?
                       more than one
                       one
Please give a name to the group which Delete belongs to:
                       menubar
```

"Command groups" in the above questions referred to the way in which the user interface organised commands. Barnard (1987) for example describes an ICS analysis demonstrating the usability implications of different command groupings in an electronic mail interface (see Figure 9). In the present study, there were two groups associated with the editing commands in question, namely, the menubar (location of the *Delete* button), and pop-up menu (*Copy* and *Move* menu options).

Simple as this concept might seem, user debate about how to answer the above two questions reflected unforeseen complexities in defining groups. Four of the six pairs wondered whether despite their spatial organisation on screen, groups should be defined on a *functional* basis – i.e. what the commands' *effects* were. Consequently, subjects suggested names such as *Editing; Modifiers; Destroyers; Destructive vs. Non-destructive; Block-text move manipulation.*

The confusion caused by this question seemed to derive from insufficient understanding of the way in which the tool acquires its description of the interface. The structural description models only the spatial or temporal relationships between basic units and their groups. Information about what commands actually *do* (and hence how they relate functionally) must be estimated by the *user* and entered in response to general questions about "similarity between members of groups" and "distinctiveness of groups" – functional attributes cannot be inferred like structural ones. Thus, the tool will detect inconsistencies between structural and functional groupings by defining all groups in structural terms, and then adding information about functional attributes in response to specific questions; this is in contrast to defining groups initially on a functional basis in the way that subjects tended to.

In summary, the task of command group definition served to illustrate very clearly the user's need to understand more about the underlying workings of the tool than originally anticipated.

'We're doing the tool's job...'

Several interesting problems arose which seemed to derive from users' expectations about what the tool could do, which in turn derived from their conceptual model of how the tool worked. Clearly this has implications for the prerequisite gulf (Figure 7) in terms of the level of training users require. This problem arose in three forms.

The first problem seemed to derive from the *kinds of questions* which were expected. This issue arose for three pairs, for the same question:

```
How well does the meaning which menubar comes to stand for,
encompass Delete?  better than other groups.
                   same as
                   worse than
```

This was asking whether the *Delete* command's positioning in the menubar was appropriate, seemed arbitrary, or if it would in fact be better located in another group (like the pop-up menu).

Three pairs of subjects were surprised at this question, because it asked them to directly address what appeared to be a large part of the design question they were consulting the expert system about – 'where should commands go?'

> *"This expert system is asking me to make some fundamental judgements about the user interface." [Pair 1]*

> *"It's asking if Delete really belongs to its group, but we don't know that yet." (i.e. that's precisely what we're trying to find out) [Pair 3]*

> *"[responding to this question] is answering the question we're trying to find the answer for – I'd have thought the system was trying to help us to make a decision on that." [Pair 5]*

This led to a suspicion as to how the tool was working, and what it would report. It was felt that having to make a decision on such an issue allowed the user to 'tell' the tool what s/he thought the analysis should conclude. One subject expressed quite strong reluctance to answer the question:

> *I've already formed a judgement about where I think Delete would be better [located], independent of the system. I'm trying to step back and not say where I think it should be, 'cos then the system will say ok, and regurgitate what it's been told... I get the impression it's going to tell me what I've already told it.*

> *E: Because you've already made lots of decisions?*

> *Yes. [Pair 5]*

Clearly, the modelling tool cannot make predictions about structural or functional properties of the user interface unless it is given information to work with, and subjects expected to be asked for relevant information. However, it seemed to be the direct, high level nature of the above question which caused concern. This level of question is, however, unavoidable for functional relationships between user interface entities ('basic units') since the system cannot infer these in the way that it can infer structural relationships. Given that analysis of user interface semantics is beyond the scope of this tool, therefore, appropriate user expectations need to be set about the system's capabilities, and about the level of detail to which it will analyse different aspects of the design.

Secondly, subjects commented on the vagueness of some of the guidance they received from Help, feeling that sometimes they were left to make decisions for which they were unqualified without more specific guidance. For example, the question:

```
Do you want to describe the commands for analysis in the current
phase?
```

offered the following Help text:

```
(...)
Providing this information can take four or five minutes, though,
and so if you do not think that these units are relevant to your
inquiry, or if they do not vary across the design options you are
considering, you do not have to describe their structure.
(...)
```

It was not clear to subjects what "relevant" meant in this context, and to decide whether or not the variable varied across the two editor designs seemed to require analyses of their own. Methods for communicating sufficient background knowledge are considered shortly in the general discussion.

Finally, the third way in which user concern that the tool was not 'doing enough' manifested itself, was the view, expressed by some, that it was not providing HCI insights which they themselves did not already know. This issue is examined more closely next.

'I could have told you that anyway...'

Over the course of this and another study (Shum, 1988), as well as in public presentations of the modelling approach by its developers, a reaction by some to the expert system's reports is that "I could have told you that anyway without the modelling." In one sense, this might be taken as a good quality check that the tool is not outputting obviously erroneous human factors advice. However, the purpose of the modelling tool is of course also to augment the user's ability to analyse a problem, rather than merely confirm intuitions.

There is at present no data on which to compare HCI analyses with and without the modelling tool (nor are we aware of systematic comparative studies of other modelling approaches). However, we would argue that the *process* of using the tool can be beneficial, not just the final report. For example, results from other empirical studies in which the tool has been either used by students (Shum, 1988) or demonstrated to researchers and practitioners (Shum and Hammond, 1993) indicate that consultation with the tool has the potential to make cognitive analyses more rigorous by prompting users to think in more detail about a design.

One interesting consequence of this is that by the time the system reports back at the end of the consultation, the issues addressed and suggestions made can seem all too obvious. To take a concrete example, to what extent did the subject quoted in the last section, who expressed reluctance to bias the tool with his own opinion, reach his conclusions about which design was better by answering the system's questions? Whether or not he would have come to the same conclusion unaided is an empirical question which at some stage needs to be investigated.

Implications for the ICS expert system encapsulation

The expert system demonstrates that ICS modelling can be automated. Given that this property of the modelling approach distinguishes it from many others, the key issue in evaluating it must be the level of knowledge actually required to use the tool. This study set out to assess the size of the gulf which remains to be bridged. In drawing our conclusions, we begin by returning to our attempt to contrast two conditions.

As briefly reported earlier, the subjects were divided into two groups of three pairs, using the tool either before or after an ICS one-day seminar. There proved to be no clear differences in the nature of the problems encountered by the two conditions during the consultation (although with only three pairs in each group, at most, any differences would have served to highlight issues for further investigation). The fact that no obvious differences emerge is also of value, however, as we consider why this might have been so, and reflect on the direction in which the results seem to point.

The one-day seminar on ICS covered a lot of ground: (i) introducing the ICS approach in the context of other approaches to delivering HCI theory into practice, (ii) stepping through several examples of how structural descriptions had been used in different domains to support ICS modelling, and (iii) demonstrating the expert system analysis of one of those examples. However, in the post experimental questionnaire none of the subjects reported that they understood the relationship between ICS and the expert system. Unfortunately, nor would they have been able to infer this relationship from the tool itself, since the original goal in designing its user interface had been to *shield* the user as far as possible from having to consider the underlying mechanisms.

This study indicates that subjects suffered by not knowing about the relationship between the modelling and the tool, or more precisely – and crucially – about *how* the two are linked, namely, via *structural descriptions* of the design, as described earlier (Figure 10). It proved impossible to design questions, even with a lot of attention to 'friendly' wording and Help, which could be answered without grasping the tool's underlying model.

It is hypothesised that most if not all of the problems reported would be resolved if the user understood that the tool acquires its representation of the user interface in these structural terms. Communicating that user interface features can be represented structurally would appear to be critical to establishing sufficient common ground (bridging the prerequisite gulf) for any subsequent interaction to be successful. It explains why an understanding of ICS as a cognitive modelling approach per se will not transfer to ability in using the tool – ICS *is* successfully hidden from the user in the expert system, but the acquisition of the structural description is all too obvious at present, dominating the dialogue. According to this interpretation, our attempt to find an effect of attending an ICS tutorial on using the modelling tool was in fact misplaced, since the crucial link between them is the structural description method.

Surface level refinement of questions and Help texts (enhanced in the system used in this study) should certainly continue, and can improve the tool's accessibility. In addition, work is currently under way to improve the knowledgebase structure, to provide alternative ways (not based around the three activity phases) to describe user interfaces. However, unless the user has grasped the approach to structural description, then these other ways of enhancing communication will be merely patching over symptoms of a deeper problem. An analogy might be the problems novices will have in using 'intuitive' graphical user interfaces if they still lack a basic model of underlying constructs such as disk, directory, file, or RAM.

In presenting the vision for the modelling tool, May *et al.* (1993) anticipated that a user would require the following skills:

> The end user of the modeller, operating in the design context, may require some relevant theoretical skills but need not be particularly skilled in the mechanics of cognitive modelling. Their skills lie more in the anticipation of likely sources of difficulty in a design, so that they know what aspects of the interface to describe to the expert system, and the appropriate level of precision with which to answer its questions. They would also be skilled in suggesting viable alternatives for the designers to consider. The role of the expert system modeller in the design process is to support the human factors specialist by automating the application of cognitive theory. (p.53)

We are now in a position to unpack some of the implications of the crucial second sentence. What do these user skills entail specifically? Firstly, May *et al.* note that users must do some preliminary analysis of their own, in order to decide which aspects of the problem may be fruitful to analyse. Clearly, any tool is dependent to some extent on the discretion of its user; however, this study illustrates the importance of training the user to specify what to analyse if the way in which the tool defines the world is unfamiliar, or requires extra work on the part of the user. Thus, in the prototype which was evaluated one initially focuses the analysis by specifying the *Phase of Activity*. Can users determine which phases are implicated? With no training, the introductory and question-related Help screens proved insufficient in the present study. Subjects had difficulty in deciding which phases (*goal formation*, *action specification*, or *action execution*) they should analyse in order to contrast the two designs.

Secondly, users need to know, "the appropriate level of precision with which to answer its questions." Within a given phase, one specifies *how detailed a structure* to analyse. For instance, whether one wished to analyse 'screen navigability' or 'icon distinctiveness' in a graphical interface would determine whether the structure one described was at the level of screens, screen-areas, windows, groups of icons, or individual icons. Users therefore need to understand (a) the structural approach which the tool employs in acquiring its representation of the user interface, and (b) how to identify and express the structural relationships associated with those parts of the design on which they wish to focus. Once that description has been provided, it is indeed true that users "need not be particularly skilled in the mechanics of cognitive modelling" to understand the tool's subsequent questions or reports.[5]

To summarise, this study provides evidence that whilst users need not concern themselves with ICS, other constructs which are used by the tool to represent aspects of the HCI world introduce non-trivial tasks. Based on this analysis, the following conclusions are drawn about the training which a human factors expert will require.

Training needs to focus on understanding and using the structural description language:

- Firstly, to answer the tool's questions, a prospective user of the expert system needs *a sufficient grasp of the generic structural description approach* to understand how to interpret and contextualise questions to their own problems. Understanding that a model of cognition is constraining how the tool interprets and resolves the structural descriptions will be helpful, but is not strictly necessary.

---

5 We recognise that we have yet to address the issue of how much human factors experts actually want to know about the ICS cognitive architecture before they genuinely trust the tool (prerequisite gulf). Answers to this question may emerge only as the tool and underlying theory mature, and designers test it in practice.

- Secondly, we propose that a key element in this training should be *extensive examples illustrating how to answer questions*—demonstrations showing alternative paths through the dialogue for different tasks, users, user interfaces, and domains. With practice, they should develop the ability to generalise from these concrete examples as they interpret questions in the context of their own designs.

Given these requirements, options open up as to where to locate this missing knowledge – with the system or the user? On the one hand, more online Help information could be introduced, whilst on the other, users could be trained so that they come to the system with a deeper understanding, and need less detailed help. These solutions could be complementary of course. In the latter case, a possibility to explore would be computer-based learning as an alternative to resource expensive human instruction. (One might even envisage ICS modelling being delivered as an integrated instructional and modelling package). However, whether or not this proved to be feasible, this study provides an empirical baseline against which to compare future 'prototype bridges' between ICS modelling and the HCI practitioners it aims to support.

To conclude, this paper argues that just as designing the user interface should not be delayed until a system's functionality has been designed, implemented and tested, nor should the process of envisioning, early prototyping and validation of potential modelling *encapsulations* be delayed until the models themselves have reached some threshold of 'readiness' for exposure to practitioners. Whilst this tendency is to some extent only natural given the 'advanced research' status of much work to date, we believe that as the field continues to mature, practitioners' requirements should shape analytic approaches earlier in their development than has been the case to date. We would encourage members of the HCI community who are developing new approaches – particularly the more abstract ones which may in the long term be the most powerful, but which may also require more training – to critically evaluate the four encapsulation gulfs which have been highlighted in this paper, in order to consider how their own models, methods, and notations may be delivered in a usable form.

Acknowledgements

Our thanks to Jon May for modelling advice throughout this study.
This work is funded by Esprit Basic Research Action 7040 (AMODEUS-2).

References

Note: All AMODEUS-2 documents are available at: http://www.mrc-apu.cam.ac.uk/amodeus

AMODEUS-2 (1992). Esprit Basic Research Action 7040: Assaying Means Of Design Expression for Users & Systems. Amodeus-2 Project, *Technical Annex* , June, 1992.

Barnard, P. (1985). Interacting Cognitive Subsystems: A Psycholinguistic Approach to Short Term Memory. In A. Ellis, Ed. *Progress in the Psychology of Language*, 197-258. London: Lawrence Erlbaum Associates.

Barnard, P. (1987). Cognitive Resources and the Learning of Human-Computer Dialogs. In J. M. Carroll, Ed. *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, 112-158. Cambridge: Mass.: MIT Press.

Barnard, P. (1991). Bridging Between Basic Theories and the Artifacts of Human-Computer Interaction. In J. M. Carroll, Ed. *Designing Interaction: Psychology at the Human-Computer Interface*, 103-127. Cambridge: Cambridge University Press.

Barnard, P. and Harrison, M. (1989). Integrating Cognitive and System Models in Human-Computer Interaction. In A. Sutcliffe and L. Macaulay, Ed. *People and Computers V: Proceedings of HCI'89*, 87-103. Cambridge: Cambridge University Press.

Barnard, P., Wilson, M. and MacLean, A. (1987). Approximate Modelling of Cognitive Activity: Towards an Expert System Design Aid. In *Proceedings of CHI+GI'87: ACM Conference on Human Factors in Computing Systems and Graphics Interface*, 21-26.

Barnard, P., Wilson, M. and MacLean, A. (1988). Approximate Modelling of Cognitive Activity with an Expert System: A Theory-Based Strategy for Developing an Interactive Design Tool. *The Computer Journal,* **31**, 445-456.

Barnard, P. J., MacLean, A. and Hammond, N. (1984). User representations of ordered sequences of command operations. *Proceedings of Interact'84: First IFIP Conference on Human-Computer Interaction, Volume 1*, London, 434-438, IEE: London.

Bellotti, V. (1988). Implications of Current Design Practice for the Use of HCI Techniques. *Proceedings of HCI'88: People and Computers V*, University of Manchester (5-9 Sept.), 13-34, Cambridge University Press.

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

Bernsen, N. O. (1993). The Structure of the Design Space. In P. F. Byerley, P. J. Barnard and J. May, Ed. *Computers, Communication and Usability: Design Issues, Research and Methods for Integrated Services*, Amsterdam: Elsevier: North Holland Series in Telecommunication.

Blandford, A. and Young, R. M. (1993). Developing Runnable User Models: Separating the Problem Solving Techniques from the Domain Knowledge. *HCI'93: People and Computers VIII*, Cambridge University Press.

Buckingham Shum, S. (1994). Analyzing the Usability of a Design Rationale Notation. In T. P. Moran and J. M. Carroll, Ed. *Design Rationale: Concepts, Techniques, and Use*, Hillsdale, NJ: Lawrence Erlbaum Associates. (Earlier version available as Amodeus-2 Working Paper TA/WP6)

Buckingham Shum, S. and Hammond, N. (1994). Argumentation-Based Design Rationale: What Use at What Cost? *To appear in: International Journal of Human-Computer Studies.*

Buckingham Shum, S., Jørgensen, A., Hammond, N. and Aboulafia, A. (1994a). (Eds.) Amodeus HCI Modelling & Design Approaches: Executive Summaries & Worked Examples. Amodeus-2 Project, *Working Paper TA/WP16*, 25 March, 1994. *(Available by anonymous ftp from: ftp.mrc-apu.cam.ac.uk::/pub/amodeus /SummariesAndExamples/SummariesAndExamples<.hqx, .rtf, .ps.Z>)*

Buckingham Shum, S., Jørgensen, A., Hammond, N. and Aboulafia, A. (1994b). Communicating & Evaluating HCI Modelling: The ISLE Hypermedia System. Amodeus-2 Project, *Working Paper TA/WP22*, 16 March, 1994.

Buckingham Shum, S., MacLean, A., Bellotti, V. M. E. and Hammond, N. (1993). Learning to Use Argumentation-Based Design Rationale. Amodeus-2 Project, *Working Paper TA/WP19*, December, 1993. (Submitted for publication)

Carroll, J. M. and Moran, T. P. (1991). *(Eds.) Human-Computer Interaction (Special Issue on Design Rationale),* **6** (3 & 4).

Coutaz, J. (1987). PAC: an Object-Oriented Model for Dialog Design. In *Proceedings of IFIP INTERACT'87: Human-Computer Interaction*, 431-436. Elsevier Science Publishers B.V: Amsterdam.

Darzentas, J., Darzentas, J. and Spyrou, T. (1993). Towards a Design Decision-Aiding System. Amodeus-2 Project, *Working Paper TA/WP9*, June, 1993.

Dix, A. (1988). Abstract, Generic Models of Interactive Systems. *Proceedings of HCI'88: People and Computers V*, University of Manchester (5-9 Sept.), 63-75, Cambridge University Press.

Duke, D. and Harrison, M. (1993). Abstract Interaction Objects. *Computer Graphics Forum,* **12**, 25-36.

Faconti, G. P. and Paterno', F. (1990). An Approach to the Formal Specification of the Components of an Interaction. *EuroGraphics'90*, 481-494, Elsevier Science Publishers.

Hammond, N., Barnard, P., Coutaz, J., Harrison, M., MacLean, A. and Young, R. M. (1991). Modelling User, System and Design: Results of a Scenarios Matrix Exercise. In *Proceedings of CHI'91: ACM Conference on Human Factors in Computing Systems*, 377-380. ACM: New York.

Harrison, M. and Barnard, P. (1993). On Defining the Requirements for Interaction. *RE'93: IEEE International Symposium on Requirements Engineering*, January, 1993, 50-54, IEEE Society Press.

Hutchins, E. L., Hollan, J. and Norman, D. A. (1986). Direct Manipulation Interfaces. In D. A. Norman and D. S.W., Ed. *User Centered System Design*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Inference (1991). *Xi Plus*. (Inference Corporation, 550 North Continental Blvd., El Segundo, CA 90245, USA)

Long, J. and Dowell, J. (1989). Conceptions of the Discipline of HCI: Craft, Applied Science, and Engineering. In A. Sutcliffe and L. Macaulay, Ed. *People and Computers V: Proceedings of HCI'89*, 9-32. Cambridge University Press.

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

May, J., Barnard, P. J. and Blandford, A. (1993a). Using Structural Descriptions of Interfaces to Automate the Modelling of User Cognition. *User Modelling and Adaptive User Interfaces* **3**, 27-64. *(Also available as Amodeus-2 Working Paper UM/WP1)*

May, J, Tweedie, L A, & Barnard, P J (1993b). Modelling User Performance in Visually Based Interactions. *HCI'93: People and Computers VIII*. Cambridge University Press: Cambridge, pp. 95-110 *(Also available as Amodeus-2 Working Paper UM/WP3)*

Nigay, L. and Coutaz, J. (1993). A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion. In *Proceedings of InterCHI'93: Human Factors in Computing Systems*, 172-178. ACM: New York.

Shum, S. (1988). Construction and Validation of an Expert System Design Tool for the Evaluation of Icons. *Masters Thesis* , Ergonomics Unit, Faculty of Engineering, University College London

Shum, S. (1993). Encapsulation Gulfs: A Framework for Communicating HCI Modelling to Designers. Amodeus-2 Project, *Working Paper TA/WP3*, May, 1993.

Shum, S. and Hammond, N. V. (1993). The 'EnEx1' Modeller-Designer Workshop: Implications for Encapsulation. Amodeus-2 Project, *Working Paper TA/WP4*, May, 1993.

Timpka, T., Nyce, J. N., Sjöberg, C. and Johansson, M. (1993). Action Design: From Modelling to Support of the Software Process. *Proceedings of Software Engineering and Knowledge Engineering'93*, San Francisco, IEEE Press.

Young, R. M., Barnard, P., Simon, T. and Whittington, J. (1989a). How Would Your Favourite User Model Cope with These Scenarios? *ACM SIGCHI Bulletin* **20** (4), 51-55.

Young, R. M., Green, T. R. G. and Simon, T. (1989b). Programmable User Models for Predictive Evaluation of Interface Designs. *Proceedings of CHI'89: ACM Conference on Human Factors in Computing Systems*, 15-19, ACM: New York.

Young, R. M., Howes, A. and Whittington, J. (1990). A Knowledge Analysis of Interactivity. *Proceedings of IFIP INTERACT'90: Human-Computer Interaction*, Cambridge, UK, 115-120, Elsevier Science Publishers B.V.: Amsterdam.