FROM TASK DOMAIN TO HUMAN-COMPUTER INTERFACE

Exploring an Information Mapping Methodology

Niels Ole Bernsen

Centre for Cognitive Science
Roskilde University
PO Box 260, DK-4000 Roskilde, Denmark
phone: +45 46 75 77 11, fax: +45 46 75 45 02
e-mail: nob@cog.ruc.dk

Steven Verjans

Limburgs Universitair Centrum
Studycentre for Multimedia Interactive Learning
Wetenschapspark 2, B-3590 Diepenbeek, Belgium
phone: +32 11 26 84 57, fax: +32 11 26 84 66
e-mail: sverjans@alpha.luc.ac.be

Summary: Human-computer interface designers are able to select from a rapidly increasing number of input/output modalities for a given application. Intuitively, the selection process can be viewed as the mapping of information from the task domain of the artefact to be built onto a set of input/output modalities for this artefact. Inherent to the mapping process is the goal of optimising the representation and exchange of information between user and system during task performance. It might be useful if ways could be found to support optimisation of information mapping. One of the authors has developed a theory called modality theory which provides a theoretical basis of information mapping. The crucial step in applying modality theory to practical interface design is to turn the above, intuitive notion of information mapping into an operational methodology. We have explored this problem in a series of design case studies. The paper reports from the explorations and presents the current version of the information mapping methodology. This version builds heavily on our most recent case study which is an information mapping analysis of the CERD interface. CERD is a flight sequencing tool for use by air traffic control officers. The paper concludes by considering open issues for future work.

Keywords: Multimodal Systems, Interface Design, HCI Design Methodology, Information Mapping, Modality Theory.

1. INTRODUCTION

It is an exciting fact of interface design that thousands of different combinations of input and/or output *modalities of information representation* are currently becoming available to designers of interfaces for human-computer interaction, from unimodal spoken language input or static graphic image output to complete multimodal virtual reality interactive systems. Each single modality or multimodal combination has its own specific capabilities and limitations for representing or conveying information and it is obviously important to be able to select the right combination of modalities for a given application. The crucial question is how this can be done in a principled manner so as to optimise the usability of the interface, given the specific purpose of the artefact to be designed. Answering this question involves addressing the research agenda of modality theory whose development formed part of the Esprit Basic Research project AMODEUS-2 [Barnard 1993]. The agenda is as follows [Bernsen 1993a]:

(1) to establish sound foundations, both conceptually, in terms of an operational taxonomy, and through actual modality analysis, for describing and analysing any particular type of multimodal output representation relevant to human-computer interaction;
(2) to create a conceptual framework for describing and analysing interactive computer interfaces so as to cover both input and output of information;
(3) to apply the results of steps (1) and (2) above to the practical problems of information mapping and information transformation between work/task domains and human-computer interfaces in information systems design.

The main problem raised by agenda item (1) is how to build the theoretical foundations for addressing the information representational capabilities of thousands of different, potentially useful combinations of output modalities or multimodal representations. The only viable approach, we argue, is through the generation and analysis of a limited set of elementary or *unimodal* modalities from which any particular *multimodal* representation or modality combination can be built. The taxonomy of unimodal output modalities which resulted from adopting this approach is presented in [Bernsen 1994a,c]. A full account of all levels of the taxonomy is in publication [Bernsen 1995b, cf. Figure 1]. Output modality theory proper, i.e. an analysis of the capabilities of information representation of each of the unimodal output modalities in the taxonomy, is currently being implemented for WWW presentation at the Centre for Cognitive Science. Recently, we have begun to address input modalities (agenda item 2). An exploratory working paper and a state-of-the-art survey are available [Bernsen 1995c, Verjans 1995].

Modality theory is not new in substance. Much work, both empirical and semantical, has been done on the information representation capabilities of static graphic graph modalities, often combined with typed natural language [Bertin 1983, Tufte 1983, Tufte 1990]. However, solid and useful taxonomic work is still in its infancy [Twyman 1979, Lohse et al. 1991, Bernsen 1992, Karagiannidis 1995]. Taxonomies of multimodal systems from a systems engineering point of view are emerging [Mackinlay et al. 1990, Carbonell 1995, Martin and Béroule 1995, Nigay and Coutaz 1995]. However, most results on output modalities have been achieved by piecemeal work on the information representation capabilities of selected families of output modalities, such as static or dynamic graphics, language and speech, touch screen interfaces, gesture, etc. [Lefebvre et al. 1993, Maybury 1993, Neuwirth et al. 1994, Ogozalek 1994]. Selective studies of the information representation and exchange capabilities of input modalities are emerging [Lefebvre et al. 1993, Bellalem and Romary 1995, Siroux et al. 1995], but input modalities still receive less theoretical treatment [Maybury 1993, Johnson 1994, Verjans 1995].

Agenda item (3) of modality theory calls for the development of an operational 'bridging' representation [Barnard 1991] between the science base of modality theory and design practice. We call this bridging representation the *Information Mapping Methodology* or IMAP. The design of a user interface can be considered an information mapping task. The task consists in mapping (a) information on the task domain of the artefact being designed as well as other relevant constraints on the design process, onto (b) a set of input/output modalities which, when fully specified, will constitute the user interface of the artefact. Interface designers thus essentially solve information mapping problems whether or not they think of themselves in those terms. Much work is currently being done on various aspects of information mapping, e.g. from task domains onto interface modalities, from the facilities offered by available software packages (such as OSF Motif) onto task domains, or from tasks or information onto input devices [Maybury 1993, Paterno 1994, Verjans 1995]. The Namur group has worked on AI knowledge-based support of information mapping in the domain of business oriented applications for nearly a decade [Bodart et al. 1995, Vanderdonckt 1995]. Again, however, the impression is that of piecemeal approaches rather than larger-scale theory-based efforts. The approach which is closest to the one adopted in this paper is the seminal work of Hovy and Arens [1990] which called for a more general approach to information mapping. This work, however, did not address taxonomy, input modalities or the haptic medium, and only partly addressed the crucial question of how to establish realistic links with design practice.

Previous investigations have shown that information mapping is a hard problem. The current version of our information mapping methodology is far from complete and mature as a practically applicable methodology. We want to highlight what we view as being perhaps the crucial issue in determining the direction of any further development of IMAP. At the CHI'95 workshop on "Knowledge-Based Support for the User Interface Design Process" [Bernsen and Verjans 1995, Vanderdonckt 1995], it was proposed that current human-computer interaction research adopt a weak sense of the term 'knowledge-based support' according to which not only rule-based AI systems but also systematically developed hypertext and hypermedia training material and guideline sets, walkthrough methodologies and other similar approaches would count as knowledge-based support. In this broad sense, modality theory will obviously be able to provide *some* knowledge-based support of information mapping in design practice. The crucial issue, then, is which kind of knowledge-based support it is realistic to aim for given the complexity of the information mapping problem. We shall return to this question in the concluding section.

IMAP was initially proposed in a case study in which a first version of the methodology was applied to early in-house design of a spoken language dialogue system prototype and to a toy 'water bath' monitoring and control system [Bernsen and Bertels 1993]. Since then, IMAP has been further developed through case studies of two full-scale, realistic design processes. In the first of these, IMAP was applied to PaTerm, an interactive tool for adding lexical databases to the commercial machine translation system PaTrans [Verjans and Bernsen 1994]. In the second, the methodology was applied to CERD, a flight sequencing tool to be used by air traffic control officers. The case studies have yielded significant insight in the merits, limitations and problems of the methodology. We report on these in the present paper, focusing in particular on the CERD case study. Section 2 describes IMAP as conceived of prior to the CERD study, including lessons learned and open problems from previous case studies. Section 3 introduces CERD and contextualises the CERD analysis. Sections. 4 and 5 present the CERD IMAP analysis. Results and lessons learned from this case study, including a revised IMAP methodology, are discussed in Section 6 which also suggests issues for future work.

Modality theory development and IMAP development are not independent of one another. A central purpose of modality theory is to support IMAP and interface design. The IMAP case studies, therefore, while each using the most recent version of modality theory for information mapping purposes, at the same time play an important role in interpreting, evaluating and shaping modality theory from the point of view of practical usability.

2. THE INFORMATION MAPPING METHODOLOGY

The intuitive idea behind IMAP is that of an operational methodology for mapping *Information1* into *Information2* via a *theory T*. Information1 is task domain information, i.e. relevant information on the artefact to be designed, such as on the tasks it should enable, its users, the work environment, the information to be exchanged between user and system during task performance, etc. Information1 is primarily being expressed linguistically as part of standard requirements capture but may also be represented as interface sketches or in other ways. Information2 is a set of possible input/output interface modalities, such as typed static graphic text, dynamic graphic images, acoustic compositional diagrams or haptic keywords for the representation and exchange of information between user and system. T is a body of knowledge which (i) states which modalities exist (i.e. the taxonomy, cf. Figure 1); (ii) analyses the properties of each modality; and (iii) expresses the suitability and/or unsuitability of a certain modality for representing certain kinds of Information1. For instance, given task domain information {a,b}, T may state that (unimodal) modality M1 is excluded from consideration whereas (unimodal) modalities M2 and M3 are suitable interface components. In this example, {a} may state the need for a user alert function, {b} may state the presence of strong ambient noise in the work environment, M1 may include the entire family of acoustic modalities, M2 may point to solutions using repetitive graphics, such as blinking or flashing representations, and M3 may point to solutions involving repetitive haptics. The claims to completeness and orthogonality of the underlying theory of output modalities mean that, in terms of our simplified example, there will not exist an entirely different modality M4, say, which has so far been overlooked by the theory nor a modality, say, M2* which is a hybrid of several of the modalities M1-M3 [Bernsen 1994a].

2.1 From Intuitive Idea to Operational Methodology

The task of IMAP development is to turn the intuitive idea of information mapping into an operational and practical interface design support methodology. How, for instance, is Information1 to be collected and represented if IMAP is to be made to work? Which types of information does, or should, it include? T cannot just be an abstract scientific theory but must be made suitably operational to serve the applied task of supporting practical interface design: how should T be expressed for this purpose? Which solutions are feasible given the complexity of the information mapping problem? What are the results of T-based IMAP? Are they complete interface specifications or something less than that? If the latter, what are they? How should IMAP results be represented - linguistically or in some other way? Until the CERD case study, we conceived of IMAP as consisting in five operational steps divided into two main phases, as follows [Bernsen and Bertels 1993, Verjans and Bernsen 1994]:

Phase 1 Step 1: Identification of information and tasks

The first problem is to identify the information to be exchanged between user and system during task performance in the application domain of the artefact to be designed. So the aim of Step 1 is to obtain that information from the task domain, which is needed to do a reasonable and possibly optimal mapping from task domain information to interface input/output representation. [Bernsen and Bertels 1993] demonstrated that the nature and variety of the information relevant to this end should not be underestimated. The fact, for instance, that, unlike France, Denmark had not introduced a Minitel-like home computer networking system at the time, turned out to support the commercial realism of building a spoken language dialogue system in the air travel information systems domain. The general problem of information mapping between task/work domain and human-computer interface cannot, therefore, be reduced to a problem about matching (a) the narrow information contents of the interactive messages between user and system during task performance with (b) the capabilities of information presentation and communication of different interface modalities. Rather, the types of information that are relevant to the definition and resolution of a particular information mapping problem may derive from any part of the design space, including information on task domain, intended users, task environment, task performance, user preferences, standards, resource constraints etc.

Still, a core part of the information needed for solving an information mapping problem is information on users' tasks. As any reasonably versatile or powerful IT artefact can be used in performing a multitude of different tasks, it is obviously not always possible during practical systems design to consider each and every such task as to its information mapping requirements. In other words, it will often be necessary during IMAP to be selective as to the tasks to be considered in any detail. The ideal way to be selective is to identify a limited set of *representative* tasks or scenarios to be performed through use of the intended artefact, and carry out the information mapping analysis on these. The problem, of course, is that no guaranteed method for generating an appropriate set of representative scenarios currently exists in HCI. In some interface design cases, such as the CERD which is a safety-critical system, complete task analysis is mandatory.

The results of Step 1 would normally be (i) high-level information relevant to solving the information mapping problem and (ii) a set of representative tasks which users should be able to carry out on or with the intended artefact. These results constitute an operationalisation of the information mapping problem, which will be completed in Step 2.

Phase 1 Step 2: Selective task analysis

In Step 2, either the selected representative tasks or the full set of tasks to be performed are each analysed in as much detail as possible in order to identify their goals and initial states, their preconditions, the activities and procedures involved, how they might go wrong, the task (work) environment, the intended users and their experience etc. The analysis should primarily aim at revealing the input/output information representation needs of the tasks. That is, whereas more or less complete task analysis may be performed either formally or informally, not all of the information it produces needs to be explicitly represented for IMAP to work. Jointly, Steps 1 and 2 prepare for Step 3.

Phase 1 Step 3: Information representation

In Step 3, the relevant information acquired through Steps 1 and 2 is represented explicitly and succinctly. We use the Design Space Development (DSD) notation for representing design space structure. The DSD notation has been developed to allow explicit representation of all the types of constraint which need to be taken into account during design reasoning and decision-making, including design decision-making on interface properties [Bernsen 1993b,c, Bernsen and Ramsay 1994a,b] (see Section 3.2). In principle, the explicit representation should include everything of relevance to the input/output modality choices to be subsequently made using IMAP. Moreover, the explicit representation should preferably be expressed in the technical terms of modality theory in order to facilitate the mapping of information from the requirements analysis (Steps 1 and 2) onto input/output modalities. Step 3 makes explicit the requirements on interactive information to be satisfied by the interface to be designed and concludes the first phase of IMAP.

Phase 2 Step 4: Information mapping

Step 4 consists in applying modality theory to the results of Step 3 above in order to map the collected task domain information onto a suitable set of input/output modalities. This is where modality theory should be able to act as a bridging representation (Section 1). As said above, modality theory is a body of knowledge which (i) states which modalities exist (Figure 1); (ii) analyses the properties of each modality; and (iii) expresses the suitability or unsuitability of a certain modality for representing certain kinds of information. Our approach to the bridging process has been that IMAP would only make use of (iii) and that the knowledge included in (iii) could be expressed in terms of rules. Thus, from the point of view of IMAP, modality theory consists in a large set of *information mapping rules,* such as the following:

> Visualise specific information in 1D, 2D or 3D spatial, temporal development being important to the visualisation <->
> Consider using dynamic analogue graphics.

Visualise specific information such that freedom of visual inspection is less important than development, movement or change <->
Consider using dynamic analogue graphics.

The expression '<->' is read, from left to right, as the 'if-then' of production rules. From right to left, rules are read 'modality X (e.g., dynamic analogue graphics) is [good/bad] at representing [left-hand side of rule]'. Technical terms occurring in the rules, such as 'specific' (or 'specificity') or 'dynamic analogue graphics', are terms of modality theory. Such technical terms are of two types, (a) terms designating modalities and (b) supporting theoretical terms. Modality theory includes 70 different unimodal modalities or modes of representation in the media of graphics, acoustics and haptics. These are presented in Figure 1 which shows the main screen of the 'Taxonomy Workbench' which is used for analysing and presenting the properties of individual modalities and their combinations [Bernsen and Lu 1995]. In modality theory, modalities are analysed at up to four different levels of abstraction, the three highest of which are shown in Figure 1. Modality properties identified at one level are inherited by that modality’s daughter nodes, their daughter nodes etc. The supporting theoretical terms

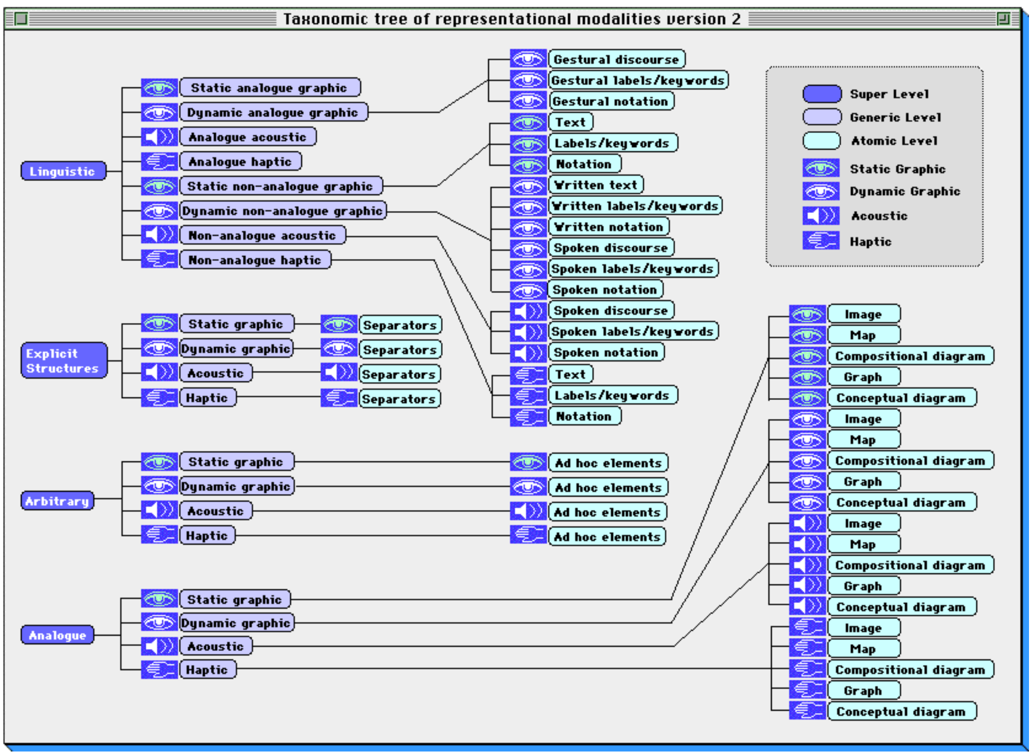


Figure 1. The taxonomy of unimodal output modalities. The three top levels are shown, from left to right: Super Level, Generic Level and Atomic Level.

are the terms, such as 'information channel', 'saliency', 'dimensionality' or 'freedom of perceptual inspection', which are needed to express the analysis of individual modalities. One result of analysing the characteristics of individual modalities are the rules of modality theory. Thus, the two information mapping rules exemplified above both derive from the analysis of the (unimodal) modality *dynamic analogue graphics*. This modality is one of the modalities belonging to the generic level of the taxonomy of unimodal modalities (see cluster in the bottom left-hand corner of Figure 1).

In this approach, the application of modality theory to information mapping can be viewed as an application of rules such as those just illustrated. Rules 'fire' when triggered by appropriate information about the task domain of the artefact which is being designed. Note that this assumes (i) that the results of Step 3 of IMAP have been expressed in such a way that the rules can actually be triggered and (ii) that modality theory includes a significant set of the rules that are relevant to deciding which input/output modalities should be used for the artefact. The result of information mapping will be sets of possible input/output modalities and modality combinations which are capable of representing the information required of the artefact. If only one solution remains after the conclusion of Step 4, IMAP terminates. It seems likely, however, that information mapping will often produce several alternative solutions which subsequently have to be compared and traded off against one another as described in Step 5 below.

Phase 2 Step 5: Trade-offs

In Step 5, a 'higher level filtering' is performed to trade off potential solutions against one another given the results of Steps 1 through 4. *Ex hypothesis,* Step 5 cannot consist in applications of the rules of modality theory but must involve other kinds of reasoning. These may range from designer craft skill argumentation through usability testing to the application of science-based approaches other than modality theory, such as user modelling techniques. The trade-off process may be explicitly represented in some form of design rationale representation as was done in [Verjans and Bernsen 1994]. The result of Step 5 is a solution to the task domain/interface mapping problem together with its justification as provided by modality theory and trade-off reasoning. Note that the final solution is not necessarily a good one. The information mapping problem may have been posed by the requirements specifications in such a way that no good solution exists. In these cases, reconsideration of the requirements specifications would seem advisable.

The iterative nature of IMAP was demonstrated in [Verjans and Bernsen 1994]. The five steps of IMAP described above may be analytically distinct but iteration turns out to occur quite frequently in practice. Typically, the information that was initially collected and represented during the first phase of IMAP turns out to be insufficient for carrying out a complete information mapping process in the second phase. The use of modality theory for information mapping often raises additional questions about aspects of the artefact to be designed, which can be answered only through further analysis of tasks, task domains, intended users etc., yielding additional explicit constraints on the design process. Since IT artefact design is generally of an iterative nature, this aspect of IMAP does not create any particular difficulty. However, because both of the two large-scale IMAP case studies to date, i.e. the PaTerm and CERD studies, addressed the results of already completed design processes done by teams other than our own, iterativity did pose practical difficulties with respect to the retrieval of additional information. Furthermore, because those design processes had not been documented using tools such as DSD, creating the explicit basis for the use of IMAP in the first place (IMAP Step 3) proved to be a reconstructive exercise of some difficulty and with little correspondence with real-life design.

2.2 Lessons Learned before the CERD Study

Our first case study had shown that information mapping problems arise at very different levels of generality during design, from early design deliberations about issues such as commercial realism or development (software) platform selection, through to the design of low-level interface details [Bernsen and Bertels 1993]. One objective of the PaTerm study was to investigate this aspect in more detail in order to derive further requirements on modality theory support of IMAP. It seemed obvious that a theory of modalities capable of supporting information mapping at all levels of design detail, would be very different from a theory which focused on supporting IMAP during some limited phase or aspect of the design process.

The PaTerm study followed to the letter the IMAP methodology presented in Section 2.1 above. We produced a complete, fully specific lexical entry tool interface together with its explicit design rationale and in the process came up with several suggestions for improvements in the existing design. The large and diverse amount of reasoning needed to achieve this result made it clear that, in general, modality theory cannot be expected to provide enough rules for solving design problems as detailed as many of those raised by the PaTerm interface [Verjans and Bernsen 1994]. For instance, modality theory will not in the foreseeable future, if at all, be developed to address low-level issues such as the size and shape of static graphic dialogue boxes. The scope of the theory remains restricted to higher levels of abstraction, such as whether the interface should have static graphic dialogue boxes in the first place. We later learned that the Namur group had reached a similar conclusion [Vanderdonckt 1995]. In retrospect it may be argued that the difficult issues of interface design in the expanding world of multimodal interfaces, lie at higher levels of abstraction whereas lower-level design details can be left to designer craft skill and standardisation. A key issue for the CERD study, therefore, became that of determining the level of generality or abstraction at which modality theory, as a science-based design support tool, might be able to provide *limited* interface design support. In the context of contemporary HCI research, the rise to prominence of this issue is not surprising as the days are long gone when HCI expected unrestricted scope of its supporting basic theory [Carroll 1991, 1993]. In other words, in addition to the contributions of modality theory, the PaTerm IMAP exercise had made us put in a large amount of designer craft skill unsupported by explicit theory. In a more mature application of IMAP, it should be possible to clearly distinguish between those information mapping results which are due to modality theory and those results which, on the other hand, are due to designer craft skills or at least are not explicitly supported by the theory. The latter results are irrelevant to the development and evaluation of IMAP as a science-based interface design support methodology.

In the PaTerm IMAP study, we did a special-purpose DSD representation which made explicit all and only the information relevant to the information mapping problem at hand. However, in view of the extent and variation of the requirements specification information which carried potential implications for information mapping, the hypothesis was proposed that a ‘full’ application of DSD to the design process might serve better as a basis for IMAP [Verjans and Bernsen 1994]. This means that *any* comprehensive design space representation which captures design commitments to the same extent as does DSD, would probably work as well. From an IMAP point of view, the advantage of this hypothesis, if true, is that no special-purpose design requirement specification has to be developed or used. Testing this hypothesis became another goal of the CERD study. In general, science-based design support always imposes extra overheads on the design process. These overheads should be minimised because they limit the practical usefulness of theory. In view of the description of IMAP in Section 2.1, two problems about the hypothesis follow. Firstly, it is not obvious that sufficiently detailed, representative task analysis information will be present in a general-purpose design commitment representation. This creates a problem for carrying out Step 2 of IMAP. Secondly, in a general-purpose representation, such as DSD, no attempt will have been made to express information in the technical terms of modality theory in order to facilitate the mapping of information from the requirements analysis onto sets of input/output modalities. This creates a problem for carrying out Step 3 of IMAP.

We now turn to the CERD case study. Due to the availability of a general-purpose CERD DSD representation (Section 3.2), Steps 1, 2 and 3 of IMAP, as described above, did not have to be performed. Moreover, Step 5 of IMAP is also absent from the information mapping analysis below. In fact, it will be concluded at the end of the paper that Step 5 does not form part of IMAP (Section 6).

3. CONTEXT OF THE CERD STUDY

3.1 What is the CERD?

The CERD (Computer Entry Readout Display) is one component in a large information system called CDIS (Central-control-function Display Information System), which is part of an even larger computer system that is being used to manage British air space. The CERD, which was developed by the British software engineering company Praxis Systems Ltd., is used to display and manipulate information about the arrival sequence of aircraft at Major Airport Complexes (MACs) such as London Heathrow. Each runway at a MAC has a Stable Approach Sequence (SAS), i.e. a sequence of aircraft waiting to land at that particular runway.

Air traffic control for a major airport involves the sequencing of incoming flights, i.e. deciding in which order the aircraft will be permitted to land, allocating aircraft to holding stacks, and allowing each aircraft to leave its stack and commence its landing approach. The arrival sequence of aircraft is illustrated in Figure 2.

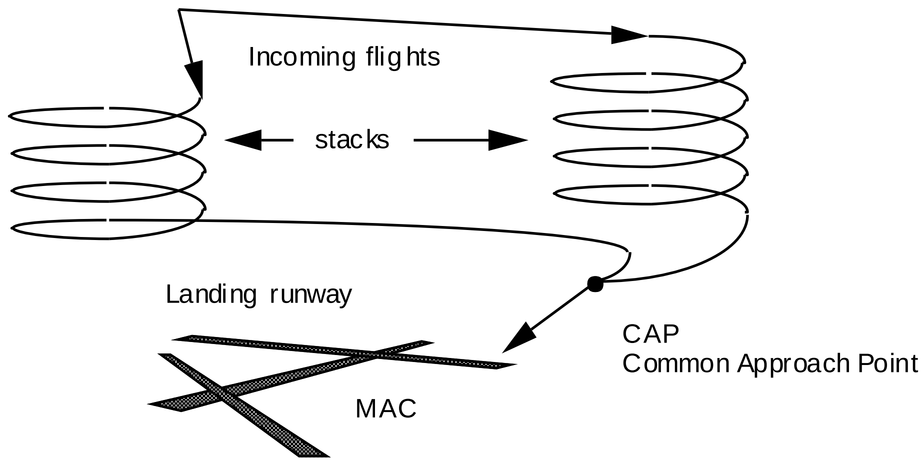


Figure 2. Aircraft approach and landing.

3.2 CERD in the Context of AMODEUS-2

In the AMODEUS-2 project, several interface designs were studied jointly by the project's different modelling and design representation teams. The CERD interface is one of these common design exemplars. For the purpose of analysing the CERD, the teams had access to documents describing the prototype interface built by Praxis. The documents provided a description of the functionality of the CERD together with an introduction to the larger framework of which the CERD device forms a small part (Section 3.1). As the Design Space Development (DSD) approach to design space representation was among the techniques investigated in AMODEUS-2, the CERD information was transferred to a DSD frame which provided a comprehensive representation of the information concerning this particular artefact [Ramsay and Bernsen 1994]. The CERD documents and the DSD representation formed a starting-point of the CERD modelling exercises by the AMODEUS-2 teams [Buckingham Shum 1994].

An IMAP analysis must make explicit all design constraints relevant to information mapping (Section 2.1). Our analysis of CERD departed from a DSD representation which had been done by Ramsay and Bernsen with no particular regard to the requirements of IMAP. This DSD describes the entire CERD including its interface as designed by Praxis. As we wanted to simulate a process in which the CERD interface is being designed through use of IMAP, we had to ‘bracket’ or suspend part of the DSD representation in order to arrive at a reduced-information DSD which could be considered a reasonable reconstruction of the knowledge available to the designers of the CERD interface. We thus suspended all the specific interface design choices that were made by the CERD designers, including aspects of layout, design of the message area, the use of scroll keys and particular sub-divisions of tasks. Another constraint that we chose to disregard, was the user preference for using a specific hardware panel, i.e. a touchscreen plasma device which has limited representational and interactional capacities. The implications of using this panel, such as limitations on screen space, minimum size of interactors, and saliency, were also suspended. The reduced-information DSD which results from the suspending exercise just described, roughly corresponds to the result of Step 3 of IMAP (Sections 2.1 and 2.2).

The IMAP approach to CERD is not primarily an evaluation of the existing CERD interface. IMAP is primarily an interface design support tool rather than a tool for the evaluation of existing interfaces. In fact, as we shall see, IMAP seems to yield both less and more than standard interface evaluation techniques: less, because IMAP does not address all the details of a completed interface; more, in so far as IMAP systematically generates an abstract interface sketch (Section 3.3) from high-level specifications by using the rules of modality theory. The generated sketch can be compared with the actual interface. Inconsistencies between the two interfaces will automatically constitute points on which the actual interface is subject to criticism from modality theory. In this way, IMAP may in principle be used for partial interface evaluation through re-doing of the interface design, i.e. by restarting from the initial constraints on the design and applying the rules of modality theory.

3.3 Novel Aspects

Compared to previous IMAP studies, the CERD case study represents several novel developments:

Firstly, the theoretical basis of IMAP, modality theory, contributes rules which are applied as part of IMAP [Bernsen 1995b]. Thus, many of the IMAP rules applied in the CERD analysis below have been independently developed as part of modality theory. The remainder of the rules have been either (a) constructed from output modality theory in order to handle the CERD problem or (b) generated ad hoc to handle problems of interactor selection. As we have only recently begun to address input modality theory, a systematic treatment of interactors does not yet form part of modality theory.

Secondly, the concept of an *interactor* plays an important role in IMAP analysis. Interactors are the parts of an interface representation through which the user can make the computer execute. An output representation in some output modality, say, a static graphic menu bar, may be either an interactor or a non-interactive part of the interface. If the representation is an interactor, users will have to use some input device to make the computer execute via the output representation. Non-interactive parts of the interface can often be turned into interactors if necessary. We shall see a number of examples of this below. The concept of interactors, in other words, appears to provide a bridge between the theory of output modalities and the unsolved problem of providing a systematic understanding of the nature of possible input modalities. We do not claim to have a fully clarified concept of an interactor. However, what we do have seems to work reasonably well in handling the CERD IMAP problem.

Thirdly, the CERD study explores the hypothesis that a general-purpose design commitments representation, such as DSD or other requirements documentation schemes, might be what best serves as a basis for IMAP (Section 2.2). If successful, this idea would save cost and effort in the practical use of IMAP.

Fourthly, we want to determine the level of generality or abstraction at which modality theory is able to provide interface design support and to be able to clearly distinguish between the contribution of modality theory and IMAP, on the one hand, and the contribution of designer craft skill and non-IMAP design support techniques, on the other (Section 2.2). The CERD study suggests a clear solution to this problem based on the notions of an *abstract interface sketch* and *interface object templates* (Section 5).

4. THE CERD IMAP ANALYSIS

The CERD interface design task involves the following four sub-tasks [Praxis document 1990]:

Flight data presentation and manipulation.
Actions on flights and supporting actions.
Functions for paging and scrolling of data.
Display and interact with selected messages.

These sub-tasks will be addressed, in the order stated, in Sections 4.1 to 4.4 below, sometimes being further sub-divided into sub-sub-tasks. Section 4.5 considers the issue of functional grouping of information and concludes the IMAP analysis. The analysis uses the template: (a) *What to represent,* i.e. relevant task domain information drawn from the DSD representation or other CERD documentation; (b) applicable *IMAP rules* from modality theory or otherwise (Section 3.3); and (c) the resulting *information representation* as expressed in terms of modality theory. Whereas (a) and (c) are mandatory, (b) is not and has, in fact, sometimes been left out, i.e. when (a) directly implies (c). It should also be noted that a particular piece of (a)-information may sometimes both directly imply (c)-information and imply (c)-information via (b). This is illustrated in the very first IMAP template below. As IMAP progresses through tasks and sub-tasks, (c)-information accumulates. In this way, a characterisation of the interface of the intended artefact is gradually being built. The process continues until IMAP has no further contribution to make because no more modality theory rules will 'fire' and no more straight logical derivations are possible from task domain information to interface properties.

In Section 5, we present two results of IMAP in addition to the *information representation* results, namely the abstract interface sketch and the idea of interface object templates. These additional results should be understood as expressions, in two different formats, of part or whole of the *information representation* results produced by IMAP. In other words, once some *information representation* result has been produced by IMAP, the abstract interface sketch and the interface object templates will incrementally incorporate this information. It may be useful to refer to Figure 3 when reading the CERD IMAP analysis.

The CERD IMAP analysis to follow is the first one to apply modality theory rules in a principled way. Many unsolved questions still remain, among which the issue concerning the level of detail at which to represent the IMAP process itself. Below, we have presented IMAP as applied to *chunks* of task domain information one after another rather than to task domain information individual piece by individual piece. Chunking means brevity of presentation at the cost of not achieving ultimate transparency and traceability of the reasoning represented. It is a matter for future investigation to decide in which way the IMAP process may be most efficiently represented. As said in Section 3.3, IMAP rules concerning the properties of interactors still lack sufficient theoretical foundation. These rules therefore should be taken with a grain of salt in what follows and their exact formulation does not represent a priority in this paper. Thus, one of the CERD designers, Anthony Hall (personal communication), has already remarked that IMAP Rule No. 9 in Section 4.1 may not possess the generality intended for IMAP rules.

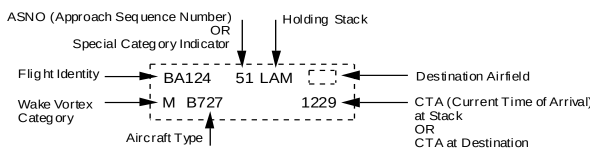
The modality theory rules used during IMAP have been enumerated for ease of reference. Unless otherwise specified, information represented under the *What to represent* -heading derives from the DSD representation of CERD. Individual *What to represent - IMAP Rules - Information representation* cycles are closed by a horizontal line. The following acronyms from the CERD task domain will occur during IMAP:

CAP Common Approach Point.

SAS Stable Approach Sequence.
NAS National Airspace System.
ATCO Air Traffic Control Officer.

4.1 Flight Data Presentation and Manipulation

What to represent
Flight information must be represented in authorised linguistic notation on the CERD rest menu (or main screen). The notation is as follows [Buckingham Shum et al. 1994, p.5]:



Information on many (approximately 20) flights must be simultaneously represented in expected landing sequence.
Allow back-and-forth search on, and comparison of, flight information.
The authorised representation of the temporal landing sequence is in terms of columns of 'flight information groups'. Columns are to be interpreted such that 'bottom right is to land first' [Buckingham Shum et al. 1994, p. 5].

Information representation
Flight information is represented in alphanumeric and time-indicating linguistic notation + explicit grouping of information per flight.
IMAP Rules
1. Allow freedom of perceptual inspection of many items of information <->
Use static graphic or static haptic modalities.

2. Static representation of information <->
Unless intended for special users (the blind or visually deficient) or unless intended for use in circumstances excluding visual access: use static graphic modalities rather than static haptic modalities.

3. Static graphic language representation of information <->
Unless intended for special purposes, such as ‘aesthetic’ display: use static graphic typed language rather than static graphic written language.

4. If explicit grouping of static graphic tokens is required <->
Include explicit static graphic structures which are as unobtrusive as possible in context.

Information representation
Flight information is represented as static graphic typed language notation (Rules 1&2&3) with unobtrusive explicit static graphic structures for grouping of information per flight (Rule 4). Flights are represented in columns indicating their expected landing sequence. Columns are to be interpreted such that 'bottom right is to land first'.

What to represent
Allow manipulation of approach sequence advisory data on at least 12 flights simultaneously.
The system requires safety-critical interaction.

IMAP Rules
5. If the information to be selected is already being represented and can be made an interactor <->

Make this information an interactor by adding interactive features.

6. Reduce error rate in safety-critical interaction with static graphic interactors <->
Interact through pointing gesture or touch.

Information representation
Create a static graphic interactor out of each representation of flight information (Rule 5). Represent all (approximately 20) flights in the SAS simultaneously. Interaction is through pointing gesture or touch (Rule 6).

What to represent
Large amounts of flight data information arrive around the clock and must be represented.
Landed flights are no longer relevant.

IMAP Rules
7. Represent large amounts of dynamically arriving data in a static modality <->
Include a Tidy interactor.

8. Create interactors <->
Use static graphic modalities or static haptic modalities by default.

9. Clear shorthand iconic representation of abstract functionality (interactors) or declarative information <->
Use static linguistic labels or keywords by default rather than dynamic icons or analogue icons. If possible, avoid abbreviating the labels/keywords.

Information representation
Create a static graphic (Rules 2&8) Tidy interactor (Rule 7). Label the interactor 'Tidy' (Rule 9). Interaction is through pointing gesture or touch (Rule 6).

What to represent
Change of CAP, i.e. the common approach point to the landing strip. CAP Change is a moment in time as of which all flights will have a different approach to the airport. Holding stacks are not influenced by CAP changes but the scope of flight actions is: no flight can be moved across this point in time.

Information representation
Create a representation to indicate the point in the SAS at which the CAP changes. SAS being represented as static graphic tokens, this important point must be represented as a static graphic structure in a way which affords sufficient saliency (Rule 4). The details of presentation of this object are unspecified.

4.2 Actions on Flights and Supporting Actions

What to represent
There are 4 different types of action to be performed on the represented incoming flights using the CERD rest (or main) menu:

a. Assign.
b. Reposition.
c. Resequence.

d. Swap.

IMAP Rules
10. Allow alternative types of action to be performed on the same represented data <->
Create interactors which clearly indicate the alternative types of action.

Information representation
Create 4 static (Rule 7) graphic (Rule 2) interactors (Rule 10) which clearly indicate alternative types of action: an Assign interactor, a Reposition interactor, a Resequence interactor and a Swap interactor. Label the interactors ‘Assign’, ‘Reposition’, ‘Resequence’ and ‘Swap’, respectively (Rule 9). Interaction is through pointing gesture or touch (Rule 6).

4.2.1 Assign

What to represent
A two-letter code is assigned to a particular flight thereby modifying its flight information representation. Codes are to be selected from up to 24 different codes to be input in authorised linguistic notation.

IMAP Rules
11. Include complex notation <->
Represent the notation in an accessible legend list.

Information representation
Code information for up to 24 different codes is presented as static graphic typed language notation (Rules 1&2&3). Each notation token is an interactor (Rule 5). Interaction is through pointing gesture or touch (Rule 6).

What to represent
Once selected and verified through user inspection, the assigned code information on a particular flight is sent as a message to NAS for confirmation. If the selected code information is falsified as a result of user inspection, the user may backtrack or cancel.

IMAP Rules
12. Selection in static graphics <->
Use available graphic information channels to indicate distinction between what is not selectable (disabled), what is selectable (enabled) and what has been selected. Use, in relative terms, lowest level of saliency for disabled functionality, higher level for enabled functionality and higher level still for selected functionality.

Information representation
Indicate, in some order, selected flight interactor, Assign interactor and selected code for verification (Rule 12). Create 3 static graphic interactors which clearly indicate the types of action they cause: a Send Message interactor, a Backtrack interactor and a Cancel interactor (Rule 10). Label the interactors ‘Send Message’, ‘Backtrack’ and ‘Cancel’, respectively (Rule 9). Interaction is through pointing gesture or touch (Rule 6). Verify selection and send message to NAS using Send Message interactor or falsify and use Cancel interactor or Backtrack interactor.

4.2.2 Reposition

What to represent

Flights are selected for repositioning in the SAS. Only 1 flight can be repositioned at a time. A flight can move max. 10 positions in the SAS. Once selected and verified, the flight information on repositioned flights is sent as a message to NAS for confirmation. If the selected flight information is falsified as a result of user inspection, the user may cancel or backtrack.

Interactors are available for this task.

Information representation
Indicate, in some order, selected Reposition interactor, flight interactor to reposition, flight interactor position in flight sequence to which repositioning should be made, verify and send message to NAS using Send Message interactor or falsify and use Cancel interactor or Backtrack interactor.

4.2.3 Resequence

What to represent
Flights are selected for resequencing in the SAS. The user indicates a list of flights to be resequenced (e.g. by selecting the first and the last flight to be resequenced) and then resequences the indicated flights. The resequence command allows the user to change the order of up to 10 adjacent flights in the SAS. Once selected and verified, the flight information on resequenced flights is sent as a message to NAS for confirmation. If the selected flight information is falsified as a result of user inspection, the user may backtrack or cancel.

Interactors are available for this task.

Information representation:
Indicate, in some order, selected Resequence interactor, up to 10 adjacent flight interactors to be resequenced, establish new flight sequence, verify and send message to NAS using Send Message interactor or falsify and use Cancel interactor or Backtrack interactor.

4.2.4 Swap

What to represent
Two flights can be selected for swapping (exchange of position) in the SAS. Once selected and verified, the flight information on swapped flights is sent as a message to NAS for confirmation. If the selected flight information is falsified as a result of user inspection, the user may backtrack or cancel.

Interactors are available for this task.

Information representation
Indicate, in some order, selected Swap interactor and the two flight interactors to swap, verify and send message to NAS using Send Message interactor or falsify and use Cancel interactor or Backtrack interactor.

4.3 Functions for Pageing and Scrolling of Data

The current CERD interface has a rest menu consisting of more than one screen and therefore raises design issues of pageing and scrolling of data. As will be shown in Section 5, IMAP implies that the interface objects needed for the CERD rest menu can in principle be presented in one screen, thus obviating the need for pageing and scrolling of data.

More generally speaking, IMAP is unable to address issues of pageing and scrolling of data in so far as those issues are to do with interface *layout.* Methodologically, layout issues presuppose that more or less all interface objects have been identified and grouped. Since IMAP stops when the identified interface objects have been grouped according to functionality, layout issues will not be addressed by IMAP. The reason why IMAP stops when the interface objects have been grouped is that modality theory has nothing to say about the details of interface layout. Moreover, application of IMAP cannot guarantee that subsequent, more detailed task analysis will not lead to the creation of additional interface objects.

4.4 Display and Interact with Selected Messages on CERD from NAS

A third main purpose of CERD, in addition to those of displaying flight information in the SAS and allowing appropriate manipulation of this information, is to receive messages from, and compose request messages to, NAS. Unfortunately, apart from the implemented interface, we had almost no information on the constraints governing the ATCO-sends-messages-to-NAS function.

What to represent
Hundreds of system-generated (NAS) messages normally arrive at the CERD and must be displayed. The messages come in three priority categories: warnings, special events and data changes. Warnings explicitly require some action on the part of the user, although these actions may not involve the CERD itself. Special events require some user action on the CERD. Data changes do not require user action on the CERD. The information that is presented to the user may, however, occasion a user action on the CERD.
Messages should be ordered by class and then by arrival within class. Warnings take priority. Warnings should have sufficient saliency. The content of messages is very diverse and may be abstract and expressed in non-descriptive linguistic acts. Messages that relate to a flight are automatically removed if that flight leaves the SAS.
Whether or not a message requires user action can only be decided by the user upon having inspected and analysed the message. When the message has been analysed and dealt with, it is no longer relevant. The user can remove messages.

IMAP Rules
13. Allow freedom of perceptual inspection <->
Include static modalities.

14. Represent abstract states of affairs (conditionals, negation, disjunction, universals, variables, concepts, etc.) <->
Include linguistic modalities.

15. Represent non-descriptive linguistic acts (orders, instructions, questions, explanations, etc.) <->
Include linguistic modalities.

16. If compact representation of specialised information for expert users is required or desirable <->
Consider using notation.

17. Remove represented static information <->
Create interactor with the function: removal of represented information.

18. Represent priority levels among data <->

Use different levels of saliency, *or*
Use temporal ordering, *or*
Use a different representation for each level.

Information representation
Represent messages as static (Rule 13) graphic (Rule 2) typed linguistic (Rules 3&14&15) notation (Rule 16). Create an interactor to allow the user to remove messages (Rule 17).

There are several alternatives for the representation of priorities among messages (Rule 18): Saliency, temporal ordering and different representations may all be used to represent the precedence of Warnings over Specials and of Specials over Data changes. Again, order of arrival within each class may be indicated using either of these mechanisms.

What to represent
Assuming that priorities among messages are to be represented in terms of levels of saliency: Represent levels of saliency in a high-acoustic environment (ATCOs wear headphones and are constantly in acoustic contact with aircraft pilots).
User preference: ATCOs dislike movement in peripheral vision.

IMAP Rules
19. In high-acoustic environment <->
Avoid, if possible, the use of acoustics for the representation of information.

Information representation
If priorities among messages are to be represented in terms of levels of saliency: avoid using acoustics for the representation of saliency (Rule 19). Avoid using dynamic representation of saliency. I.e. represent saliency levels in static graphics or static haptics.

4.5 Grouping of Information

An important graphic interface design requirement is that the screen's functionality be logically grouped. The above IMAP analysis has generated the following five groups of information representations and interactors at the interface:

Group 1: Flight data representation.
Group 2: Flight data interactors.
Group 3: Assign code interactors.
Group 4: Representation-linked interactors.
Group 5: Message area and message interactor.

In addition, the IMAP analysis has generated requirements on the nature of the interface, i.e. that it must be either a pointing gesture screen or a touch screen, and that there should be differentiated representation of disabled, enabled and selected information, respectively.

5. THE ABSTRACT INTERFACE SKETCH AND INTERFACE OBJECT TEMPLATES

This section presents two additional results of IMAP, the abstract interface sketch and the idea of interface object templates. Both are generated from the *information representation* advice produced by IMAP (Section 4). None of them contain relevant additional information. Their advantages lie elsewhere, namely in representing IMAP results in formats which are useful to subsequent stages in the design process. The abstract interface sketch provides a first *analogue* representation of the interface and may be used for detailed analysis of how the intended user tasks can be performed at the human-computer interface of the artefact being designed. The linguistic *information representation* of Section 4, being non-analogue, is not suited for this purpose [Bernsen 1995a]. The idea of interface object templates is to make fully explicit for each interface object, and to standardise, the information produced by IMAP in such a way as to bridge between IMAP and interface implementation.

5.1 The Abstract Interface Sketch

The *abstract interface sketch* which resulted from the CERD IMAP analysis is presented in Figure 3. This is as far as IMAP goes in the CERD case. A more detailed specification and layout of the CERD interface is beyond the IMAP methodology. From the point of view of modality theory, Figure 3 presents a somewhat self-contradictory entity, i.e. an 'abstract sketch'. There is an important sense in which sketches cannot be abstract, they are always perfectly specific [Bernsen 1995a]. Figure 3 pre-

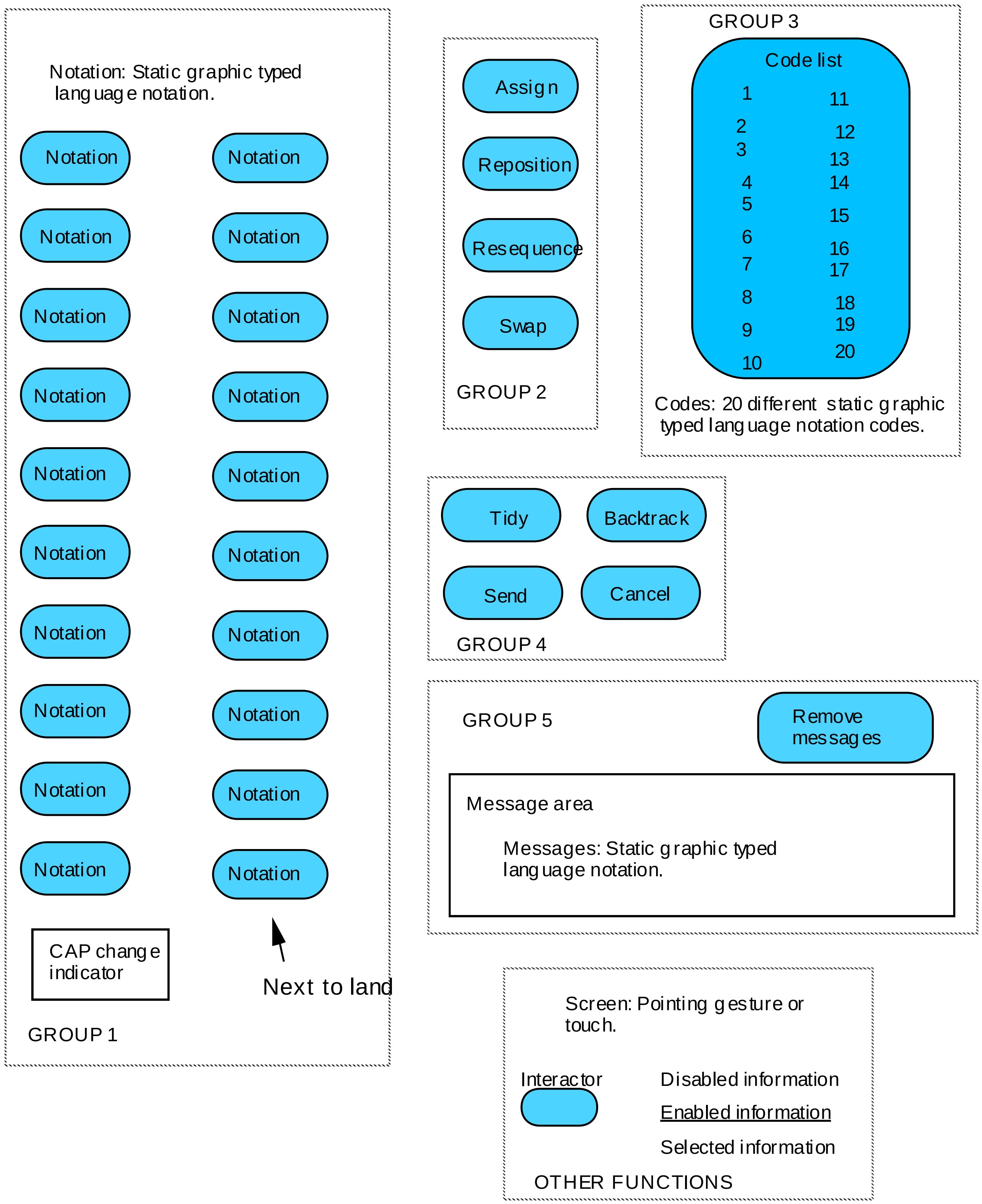


Figure 3. The abstract sketch of the CERD interface resulting from the CERD information mapping analysis. The flight information notation is presented in detail in Section 4.1.

sents an abstract sketch in the following sense: the number of interface objects in the figure follows from the IMAP analysis in Section 4 and so do their functionality and grouping. However, it does *not* follow from the IMAP analysis that the interface objects have the size, shape, colour, position, saliency differences and font labelling that they have in Figure 3, nor are there any constraints on screen size other than those imposed by current technologies. Furthermore, the screen may be a touch screen or one accessible to pointing gesture using some appropriate device. The way in which all these properties are represented in, or suggested by, the abstract interface sketch in Figure 3, can be changed without any conflict with IMAP. It also follows from the incompleteness of the IMAP sketch that further analysis, of user task details or otherwise, may lead to revision of the number of interface objects established through IMAP. The abstract interface sketch contains the functionality, objects and groupings which are necessary, as a minimum, to enable the performance of the intended tasks according to the *What to represent* part of IMAP. This minimum, of course, may not be an optimum from, e.g., a usability point of view.

To summarise, an IMAP analysis based on modality theory leaves open issues of detailed task analysis, low-level interface details, and layout.

On the other hand, and this is one of the unique advantages of representations such as the (analogue) sketch in Figure 3, the abstract interface sketch enables the performance of more detailed task analyses and user interface evaluations which may help answer the many questions unresolved by IMAP concerning, i.a., the detailed design of the complex *Resequence* function which, in the original CERD interface, requires an auxiliary screen, of the *Message area* and the procedures for ATCO's messages to NAS, of user's order of selection of action (such as *Swap)* and flight, of the detailed representation of the CAP-change indicator, or of how to represent landed flights (e.g., are they 'greyed out' or not?). The abstract interface sketch provides, as it were, the first usable representation of the user's task world as defined in terms of the interface of the emerging artefact. In addition, the sketch is useful to the study of layout because it can be experimented with through direct manipulation.

Figure 3 shows that +20 flights can be simultaneously represented on a screen whose interactors are accessible through pointing gesture or touch. All flights in the SAS, therefore, can be simultaneously represented on one screen provided that the screen is being tidied sufficiently often. As there is no need to represent landed flights, the representations of these flights may simply disappear when the screen is being tidied. The *Tidy* operation will allow representation of all flights in the current landing sequence. It follows that one (non-scrolling) main screen or 'Rest Menu' is sufficient (Section 4.3).

5.2 Interface Object Templates

As shown in Section 5.1, the abstract interface sketch is incapable of adequately representing the information that was acquired during IMAP. On the one hand, the sketch provides *incomplete* information. In particular, the sketch does not include information on the abstract functionality of representations and interactors, or on crude sequences of user operations, which, although derived during IMAP, cannot easily be represented in analogue sketch form. For instance, the CERD sketch does not list the operations which are required for repositioning a flight in the SAS (Section 4.2.2). On the other hand, the sketch provides *too much* information, i.e. more than was generated through IMAP. This is because the sketch, being analogue, cannot but specify aspects of representations and interactors which were left unspecified by the IMAP analysis, such as the shape, size, position, shading and colour of an interactor. When it comes to implementing the interface, we must be fully clear and systematic about its properties in all respects, and more so than the IMAP *information representation* parts of Section 4. The linguistic interface object template representation is a first attempt at addressing these problems. An example template from the CERD study is presented in Figure 4.

The terminology of Figure 4 is from Andersen's introduction to computer semiotics [Andersen 1990] and from [Verjans 1994]. The object template has been designed to represent the following information: the type of representation (sign), its output and input medium, the user action(s) related to the interface object (its functionality), and a number of permanent and transient features. The interface object template summarises the results of IMAP. In addition, the template specifies which properties of an object were left open by IMAP and which, therefore, still need to be determined. In contrast to both the abstract interface sketch and the IMAP representation in Section 4, the template is fully explicit and not overly specific. Moreover, an object is a concept that most present-day programmers are familiar with from object-oriented analysis and programming. As such, the object template representation may be a promising representation of the results of information mapping. The representation in Figure 4 is a first draft, however. More research is needed to determine whether and how interface objects and related functionality can be best represented in such a way as to support subsequent implementation work.

OBJECT 1: Flight data key

Sign type	interactor
Output medium	graphic
Input medium	type: direct manipulation
Action	user action = selection result = system action: system remembers selection and provides feedback (e.g. changes level of selectability)
PermanentFeatures	size (unspecified) shape (rectangular structure) possibleMeaning (linguistic notation items) perceivable reachable
TransientFeatures	3 levels of selectability (unselectable/selectable/selected) (reflected in saliency) 3 levels of saliency (arbitrary) position moveable (indirectly through activation of TIDY interactor = OBJECT3)

Figure 4. Example of an interface object template from the CERD IMAP analysis. There is a dynamically changing number of instances of OBJECT 1 in the CERD interface.

6. CONCLUSION AND FUTURE WORK

In this section, we present some tentative generalisations from the findings of the CERD IMAP case study including a revised IMAP methodology, followed by some issues for future work.

6.1 IMAP Located in the Design Process

The first conclusion to be tentatively drawn is that through the IMAP methodology, modality theory is capable of offering *some* amount of support of interface design from the beginning of the design process through to some level of specificity short of interface layout, interface properties that depend on detailed task analysis and lower-level interface details. It follows that the analysis of representative tasks on which IMAP is partly based (Section 2.1), only needs to be carried out to a corresponding level of detail. This again means that the fact that general-purpose requirements capture representations such as DSD may lack detailed task analysis information, does not count against using such representations as a basis for IMAP (Section 2.2). In fact, it appears that those representations are bound to be lacking in such details as long as coarse-grained interface design at the IMAP level has not yet been carried out.

What determines the level of specificity and completeness to which modality theory is capable of supporting IMAP? This level is determined by the level of abstraction of modality theory. The theory analyses the representational advantages and disadvantages of individual unimodal modalities at four levels of abstraction, i.e. from the highly abstract Super Level through to the Generic Level, the Atomic Level and, in some cases, the Sub-Atomic Level (Section 2.1). Rules describing what a certain modality is good or bad at representing are tied to the level of abstraction at which the modality is being analysed. When no finer distinctions between different unimodal modalities are being made, modality theory stops producing rules. The CERD IMAP analysis provides a first illustration of the extent of modality theory support of interface design. Roughly, IMAP seems capable of generating all or most interface objects, their groupings and important parts of the interface functionality.

The final results of information mapping are one or more abstract interface sketches and succinct descriptions of the generated interface objects. These results would appear to provide a useful basis for doing the final and detailed interface design involving detailed task analysis on the artefact being designed, detailed specification of the sequences of user operations, information channel selection for the purpose of low-level interface design, and layout design. Indeed, the abstract interface sketch can be used as a vehicle for performing representative task analysis. If true more generally, this means that IMAP only requires 'raw' task analysis rather than, as assumed in [Verjans and Bernsen 1994] and the IMAP methodology presented in Section 2.1, detailed, fine-grained analysis of representative tasks. Many sub-task issues are simply irrelevant to IMAP as modality theory has nothing to contribute to them. Furthermore, it is quite possible that the really detailed task analysis needed for the complete specification of the artefact to be built, *can only be performed when supported by something like our abstract interface sketch.* It is only when one has a first general notion of the interface of the artefact being designed, that is becomes possible to tell how the intended tasks can be carried out in detail through interaction with the artefact. Without the sketch and the objects it affords, it is not possible to go into the details of task performance on the emerging artefact. In that case, IMAP can be claimed to support the crucial step in interface design of going from task notions defined *independently of* the artefact being designed, to task notions defined *in terms of* the designed artefact.

The paper has produced a clear answer to the question of how to distinguish between IMAP results which have been produced from using modality theory as science base and results which have been produced on the basis of individual designer craft skill or through other techniques. Information mapping is inference which leads from abstractly (i.e. linguistically) represented information representation requirements to physically instantiated human-computer interface modalities which express the required information. An IMAP result is based on modality theory as a scientific modelling approach if and only if that result is either due to the firing of a modality theory rule or to a straightforward logical inference from established requirements specifications of how information should be represented at the interface being designed.

6.2 A Revised IMAP Methodology

Given the fact that IMAP always and necessarily produces incomplete interface sketches, it seems doubtful whether Step 5 of IMAP, i.e. the final trade-off reasoning or 'higher level filtering' step can be upheld as presented in Section 2.1. The assumption behind Step 5 has so far been that the purpose was to trade off two or more, in some sense ‘complete’, interface designs. But since IMAP never produces complete interface designs, this assumption is false and the very idea of Step 5 would therefore appear unfounded. The iterative nature of IMAP means that, as long as modality theory is likely to produce further constraints on (or properties of) the interface to be designed, it may pay off to go back to the task domain to identify further information which may lead to the ‘firing’ of additional rules supported by the theory. It is, therefore, no accident that Step 5 was not performed on the CERD interface above.

The requirements specification of the CERD design, which was done without IMAP support in mind, has worked well as a structured representation of design commitments for use by IMAP. The conclusion is that IMAP does not seem dependent on IMAP-specific design representations. If true, this will save cost and effort in the practical use of IMAP. Any comprehensive, explicit representation of the design space around the artefact to be built and created prior to the design of its interface, may be used in conjunction with IMAP. Given that representation, IMAP promises to get interface design started and advanced to the point where modality theory has no more support to offer. In other words, Steps 1-3 of IMAP are not needed in those cases where the design process has already been explicitly represented. One problem remains with respect to this optimistic scenario, however. It concerns the need for translating the collected task domain information into the technical terms of modality theory in order to facilitate the mapping of information from the requirements analysis onto the choice of input/output modalities (Sections 2.1 and 2.2). Clearly, general-purpose representations such as DSD do not embody translations into the notions of modality theory. The translation was done implicitly in each IMAP cycle presented in Section 4 above. This solution is only available at the expense of involving experts on IMAP and modality theory, which would be a major drawback when using IMAP to support interface design practice. An alternative solution could be to automate a set of questions which would elicit from the interface designers the information required for IMAP. In this case, the designers would not have to be modality theory experts. They would merely have to be able to correctly understand the questions posed to them by a question-answering system. Once answered, the information on design constraints would be represented in a form accessible to the IMAP rules.

Summarising, a revised IMAP methodology includes the following iterative steps:

Step 1. System and interface requirements gathering resulting in a comprehensive explicit representation. A high degree of detail must be achieved with respect to artefact functionality and information to be represented and exchanged by user and system during task performance. Less detailed information is required (and possible) with respect to the analysis of task performance on the interface to be designed. The types of information relevant to information mapping may otherwise derive from any part of the design space, including information on task domain, intended users, task environment, task performance, development environments, user preferences, standards, resource constraints etc. (Section 2.1).

Step 2. Translation into the conceptual apparatus of modality theory. This may be done by a modality theory consultant or through a question-answering system which elicits the requirements specification information from the designers.

Step 3. Information mapping through modality theory rule-firing. The initial result is a coarse-grained linguistic specification of the interface. This specification represents the simplest possible, functionally adequate interface relative to the requirements specification.

Step 4. Creating, from (3), (a) an abstract interface sketch and (b) an abstract and partial, linguistic specification of each individual interface object. This specification must be capable of supporting subsequent work on interface implementation.

As viewed from the perspective of the completed interface, IMAP produces many open design options, in particular with respect to interface layout, low-level interface detail and optimisation of interface usability. To rationally select between these options, designer craft skill or additional science-based methods, such as user modelling methods, are required.

6.3 Issues for Future Work

The series of case studies reported in this paper, and in particular the CERD case study, have helped clarify some of the problems involved in providing theory-based interface design support in the form of principles of information mapping between task domains and interface modalities. In particular, we know more about the nature and extent of the interface design support that could be offered by modality theory. At least five lines of inquiry follow more or less straightforwardly from the discussion above. *Firstly,* the revised IMAP methodology must be further developed. Among the issues to be addressed is that of identifying systematic and efficient heuristics for carrying out the information mapping task. For instance, should the interface design task be hierarchically broken down into 'issues' to be sequentially addressed by IMAP? How should the complex rule set of modality theory be searched to enable efficient information mapping? How to translate the task domain information into the technical terms of modality theory in order to facilitate the information mapping process? *Secondly,* modality theory must be generalised to be able to account for input modalities and user-system interaction. *Thirdly,* more case studies are needed both in order to test and enlarge the rule set of modality theory and in order to assess the complexity involved in generating a really comprehensive rule base. Given the scope of modality theory, we might easily be talking about hundreds of rules, which immediately raises issues such as that of checking for rule consistency. Moreover, the task of verifying all these individual rules is a daunting one in itself. It is not clear that the rules used in the CERD IMAP exercise (Section 4) are all valid as they stand. The nature of the evolving rule base will serve as a basis for judging whether, *fourthly,* it will be possible in practice to build a rule-based design support system based on modality theory as well as a question-answering system for eliciting requirements specifications from interface designers. *Fifthly,* we need to investigate the level of detail at which to represent the IMAP process itself. The representation in Section 4 does not render the underlying reasoning sufficiently transparent. We are currently making the first application of IMAP to a commercial monitoring and control systems design project in which we act as consultants and modality theory experts. The issues just mentioned will be addressed when analysing the project results.

As argued in Section 1, it might be advisable for HCI to adopt a weak sense of the term 'knowledge-based support' according to which not only rule-based AI systems but also systematically developed hypertext and hypermedia training material and guideline sets, walkthrough methodologies and other similar approaches would count as knowledge-based support. Given the unsolved problems just outlined, caution seems in order with respect to the prospects of developing an automatic, rule-based IMAP system. It the problems turn out to be too hard, more "lightweight" approaches to making modality theory useful to interface design will have to be sought bearing in mind that modality theory is more than a set of rules. Such approaches might include using the hypertext/hypermedia version of the theory currently under development for designer training, developing a theory-based walkthrough methodology, or limiting automation or other forms of theory application to some segment of the theory.

REFERENCES

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

[Barnard 1991] Barnard, P.: Bridging between basic theories and the artefacts of human-computer interaction. In J.M. Carroll (Ed.): *Designing Interaction. Psychology at the Human-Computer Interface.* Cambridge: Cambridge University Press.

[Barnard 1993] Barnard, P.J.: Modelling users, systems and design spaces (Esprit Basic Research project 3066-AMODEUS). *Proceedings of HCI 93*, Orlando, Florida.

[Bellalem and Romary 1995] Bellalem, N. and Romary, L.: Reference interpretation in a multimodal environment combining speech and gesture. *Proceedings of the First International Workshop on Intelligence and Multimodality in Multimedia interfaces (IMMI-1),* July, Edinburgh, Scotland.

[Bernsen 1992] Bernsen, N.O.: Matching Information and Interface Modalities. An Example Study. *Working Papers in Cognitive Science* WPCS-92-1, Centre of Cognitive Science, Roskilde University.

[Bernsen 1993a] Bernsen, N.O.: A research agenda for modality theory. In Cox, R., Petre, M., Brna, P. and Lee, J. (Eds.): *Proceedings of the Workshop on Graphical Representations, Reasoning and Communication.* World Conference on Artificial Intelligence in Education, Edinburgh, August, 43-46.

[Bernsen 1993b] Bernsen, N.O.: The structure of the design space. In Byerley, P.F., Barnard, P.J. and May, J. (Eds.): *Computers, Communication and Usability: Design Issues, Research and Methods for Integrated Services.* Amsterdam, North-Holland, 221-244.

[Bernsen 1993c] Bernsen, N.O.: Structuring design spaces. *InterCHI '93 Adjunct Proceedings,* 211-212.

[Bernsen 1994a] Bernsen, N.O.: Foundations of multimodal representations: A taxonomy of representational modalities. *Interacting with Computers* Vol. 6, 4, 347-71.

[Bernsen 1994c] Bernsen, N.O.: A revised generation of the taxonomy of output modalities. *Esprit Basic Research project AMODEUS Working Paper* RP5-TM-WP11. *CCI Working Papers in Cognitive Science and HCI*, WPCS-94-7. Centre for Cognitive Science, Roskilde University 1994.

[Bernsen 1995a] Bernsen, N.O.: Why are analogue graphics and natural language both needed in HCI? In Paterno, F. (Ed.): *Interactive Systems: Design, Specification and Verification.* Springer-Verlag 1995, 235-51.

[Bernsen 1995b] Bernsen, N.O.: A toolbox of output modalities (submitted).

[Bernsen 1995c] Bernsen, N.O.: A taxonomy of input modalities. *Esprit Basic Research project AMODEUS-2 Working Paper* RP5-TM-WP22. *CCI Working Papers in Cognitive Science and HCI*, WPCS-95-9. Centre for Cognitive Science, Roskilde University 1995.

[Bernsen and Bertels 1993] Bernsen, N.O. and Bertels, A.: A methodology for mapping information from task domains to interactive modalities. *Esprit Basic Research project GRACE Deliverable 10.1.3. CCI Working Papers in Cognitive Science and HCI*, WPCS-93-10. Centre for Cognitive Science, Roskilde University 1993.

[Bernsen and Lu 1995]. Bernsen, N.O. and Lu, S.: A software demonstrator of modality theory. In Bastide, R. and Palanque, P. (Eds.): *Design, Specification and Verification of Interactive Systems '95.* Springer-Verlag 1995, 242-61.

[Bernsen and Ramsay 1994a] Bernsen, N.O. and Ramsay, J.: An executive summary of the DSD framework illustrated by two worked exemplars. In Buckingham Shum, S., Jørgensen, A., Hammond, N. and Aboulafia, A. (Eds.): AMODEUS-2 HCI modelling and design approaches: Executive summaries and worked examples. *Esprit Basic Research project AMODEUS-2 Working Paper* RP4-TA-WP16.

[Bernsen and Ramsay 1994b] Bernsen, N.O. and Ramsay, J.: Design structure, process and reasoning. The advancement of a tool for the development of design spaces. *Esprit Basic Research project AMODEUS-2 Working Paper* RP3-ID-WP28.

[Bernsen and Verjans 1995] Bernsen, N.O. and Verjans, S.: Information mapping. Knowledge-based support for user interface design. In *Proceedings of the CHI '95 Workshop on Knowledge-Based Support for the User Interface Design Process,* Denver, Colorado May 1995.

[Bertin 1983] Bertin, J.: *Semiology of Graphics. Diagrams. Networks. Maps.* Madison, University of Wisconsin Press.

[Bodart et al. 1995] Bodart, F., Hennebert, A.M., Leheureux, J.-M., Provot, I., Zucchinetti, G. and Vanderdonckt, J.: Key activities for a development methodology of interactive applications. In Benyon, D. and Palanque, P. (Eds.): *Critical issues in user interface systems engineering.* Springer-Verlag 1995.

[Buckingham Shum 1994] Buckingham Shum, S. (Ed.): Preliminary modelling of the CERD flight sequencing tool. *Esprit Basic Research project AMODEUS-2 Working Paper* RP4-TA-WP23.

[Buckingham Shum et al. 1994] Buckingham Shum, S., Duke, D., Hammond, N. and Jørgensen, A.: CERD: Functionality, design process, and modelling issues. *Esprit Basic Research project AMODEUS-2 Internal Report* RP4-TA-IR3.

[Carbonell 1995] Carbonell, N. (Ed.): *Multimodal Human-Computer Interaction. Proceedings of the ERCIM Workshop,* Nancy, France, November 1994.

[Carroll 1991] Carroll, J.M. (Ed.): *Designing Interaction. Psychology at the Human-Computer Interface.* Cambridge: Cambridge University Press.

[Carroll 1993] Carroll, J.M.: Creating a design science of human-computer interaction. *Interacting with Computers* Vol. 5, 1, 3-12.

[Hovy and Arens 1990] Hovy, E. and Arens, Y.: When is a Picture Worth a Thousand Words? Allocation of Modalities in Multimedia Communication. Paper presented at the *AAAI Symposium on Human-Computer Interfaces*, Stanford 1990.

[Johnson 1994] Johnson, P. (Ed.): *Intelligent Multi-Media Multi-Modal Systems, Proceedings of the AAAI Spring Symposium,* Stanford, March.

[Karagiannidis 1995] Karagiannidis, C., Koumpis, A. and Stephanidis, C.: Media/modalities allocation in intelligent multimedia user interfaces: Towards a theory of media and modalities. *Proceedings of the First International Workshop on Intelligence and Multimodality in Multimedia interfaces (IMMI-1),* July, Edinburgh, Scotland.

[Lefebvre et al. 1993] Lefebvre, P., Duncan, G. and Poirier, F.: Speaking with Computers: A Multimodal Approach. In *Proceedings of EUROSPEECH '93.*. Berlin, September, 1665-68.

[Lohse et al. 1991] Lohse, G., Walker, N., Biolsi, K. and Rueter, H.: Classifying Graphical Information. *Behaviour and Information Technology* 10, 5, 419-36.

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

[Martin and Béroule 1995] Martin, J.C. and Béroule, D. (1995) Multimodal Interfaces Based on Types and Goals of Cooperation Between Modalities. *Proceedings of the First International Workshop on Intelligence and Multimodality in Multimedia interfaces (IMMI-1),* July, Edinburgh, Scotland.

[Maybury 1993] Maybury, M.T., Ed.: *Intelligent Multimedia Interfaces.* Cambridge, MA, MIT Press.

[Neuwirth et al. 1994] Neuwirth, C.M., Chandhok, R. Charney, D., Wojahn, P. and Kim, L.: Distributed collaborative writing: A comparison of spoken and written modalities for reviewing and revising documents. *Proceedings of CHI'94,* 51-57.

[Nigay and Coutaz 1995] Nigay, L. and Coutaz, J.: Multifeature systems: From HCI properties to software design. *Proceedings of the First International Workshop on Intelligence and Multimodality in Multimedia interfaces (IMMI-1),* July, Edinburgh, Scotland.

[Ogozalek 1994] Ogozalek, V.Z. (1994): A comparison of the use of text and multimedia interfaces to provide information to the elderly. *Proceedings of CHI'94,* 65-71.

[Paterno 1994] Paterno, F. (Ed.): *Design, Specification and Verification of Interactive Systems. Proceedings of the Eurographics Workshop,* Carrara, Italy, June. Also as Paterno, F. (Ed.): *Interactive Systems: Design, Specification and Verification.* Springer-Verlag 1995.

[Praxis-document 1990] Central control function: System requirements specification, Vol. 6, Approach Sequencing. *Praxis Document,* 6th November.

[Ramsay and Bernsen 1994] Ramsay, J. and Bernsen, N.O.: DSD report on the CERD exemplar. In Buckingham Shum, S. (Ed.): Preliminary modelling of the CERD flight sequencing tool. *Esprit Basic Research project AMODEUS-2 Working Paper* RP4-TA-WP23.

[Siroux et al. 1995] Siroux, J., Guyomard, M., Multon, F. and Remondeau, C.: Oral and gestural activities of the users in the GEORAL system. *Proceedings of the First International Workshop on Intelligence and Multimodality in Multimedia interfaces (IMMI-1),* July, Edinburgh, Scotland.

[Tufte 1983] Tufte, E.R.: *The Visual Display of Quantitative Information.* Cheshire, CT, Graphics Press.

[Tufte 1990] Tufte, E.: *Envisioning Information.* Cheshire, CT, Graphics Press.

[Twyman 1979] Twyman, M.: A Schema for the Study of Graphic Language. In Kolers, P., Wrolstad, M. and Bouna, H. (Eds.): *Processing of Visual Language* Vol. 1. New York, Plenum Press.

[Vanderdonckt 1995] Vanderdonckt, J.: Knowledge-based systems for automated user interface generation: the TRIDENT experience. In *Proceedings of the CHI '95 Workshop on Knowledge-Based Support for the User Interface Design Process,* Denver, CO, May 1995.

[Verjans 94] Verjans, S.: Computer semiotics as a basis for integrating different views in human-computer interaction. *Esprit Basic Research project AMODEUS-2 Working Paper* RP5-TM-WP10.

[Verjans 1995] Verjans, S.: State-of-the-art for input modalities. *Esprit Basic Research project AMODEUS-2 Working Paper* RP5-TM-WP20.

[Verjans and Bernsen 1994] Verjans, S. and Bernsen, N.O.: PaTerm: A case study in information mapping. *Esprit Basic Research project AMODEUS-2 Working Paper* RP5-TM-WP6. *CCI Working Papers in Cognitive Science and HCI*, WPCS-94-4. Centre for Cognitive Science, Roskilde University 1994.

Acknowledgements. Thanks are due to Laila Dybkjær, Flemming Jensager, Shijian Lu and Judith Ramsay for many helpful comments on an earlier draft. We also wish to thank the PaTerm designers at the Centre for Language Technology, Copenhagen, for their kind assistance and valuable time. Thanks are also due to the anonymous reviewers of the extended abstract preceding this paper. The work reported in this paper was done on grants from the Esprit Basic Research projects GRACE and AMODEUS-2 whose support is gratefully acknowledged.