The AMODEUS Project
ESPRIT Basic Research Action 7040

Design Structure, Process and Reasoning. The Advancement of a Tool for the Development of Design Spaces

Bernsen, N.O and Ramsay, J.

20.6.94

ID/WP28

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

Design Structure, Process and Reasoning
The Advancement of a Tool for the Development of Design Spaces

Figures

DESIGN STRUCTURE, PROCESS AND REASONING

The Advancement of a Tool for the Development of Design Spaces

Abstract. This paper introduces the Design Space Development (DSD) framework and methodology for developing design spaces. DSD is about *structure, process and reasoning* during design. Using DSD lends *structure* to the design space by way of a semi-formal notation which encourages the designer to consider the space in terms of a set of invariant elements. The technique's methodology supports the design *process* as an incremental gathering of design commitments as the design space develops, and provides a second notation for the recording of design *reasoning* or problem solving, and its accompanying justification. The development of DSD was realised through its application to a series of design processes, two of which are presented in this paper, one in multimedia audio-visual communication, and one in air-traffic control. We discuss the stabilising of a framework for structuring the design space, which promotes transparency and explicitness, and the extension of the technique through theoretical integration with a representation of design rationale (DR), lending closure to design space representation and development.

PART ONE

INTRODUCTION TO DESIGN SPACE DEVELOPMENT

Introductory Remark

This AMODEUS II Working Paper is a draft presentation of the DSD approach to be completed in due course. The paper is based on two case studies presented in AMODEUS II Internal Reports (Bernsen and Ramsay 1994a, Ramsay and Bernsen 1994) and a draft DSD Manual (Bernsen and Ramsay 1994b, in preparation). The studies reported in the internal reports paved the way for integrating Design Rationale into DSD and the DSD Manual will be completed taking into account the lessons learned from the case studies.

Design Space Development: Why we need it

Current software design practice is lacking in principled support for the proper usability engineering of artifacts and most software in actual use fails badly when measured by reasonable usability engineering standards. We are often only able to use it for our own purposes because we somehow manage to assimilate a small subset of its functionality through trial and error and ample help from colleagues who have already struggled through the same ill-guided learning process. Most users quickly realise that manuals and on-line help functions provide much less support than initially expected (Ramsay and Oatley, 1992). There is good reason to believe that the major productivity gains to be reaped from properly usability engineered artifacts lie still ahead (Nielsen 1993).

The Design Space Development (DSD) approach represents one piece of the much larger puzzle which must be solved to establish a complete, general and practically useful framework for the support of IT artifact usability engineering. To see where development of the design space might fit in we may distinguish from the bottom up between three levels which need to be addressed to establish such a framework.

At *Level 1* we find the actual design process in which a variety of usability engineering approaches already fit naturally, such as:

- designer craft skills for usability problem prevention, detection and solution;
- written material on usability problems with previous systems of similar kinds, software reviews, etc.;

- empirical results from field studies in the intended users' work environments including results from talking to intended users, semi-structured interviews, etc. and empirical results from observing users' task performance and from having them think aloud;
- results from the use of rapid prototyping methods of various kinds, from paper mock-ups and story-boarding through to semi-implemented systems and advanced, specialised methods such as the Wizard of Oz for natural language dialogue systems design;
- designing around tasks during user-system interaction development;
- results from using scenarios and step-by-step designer walkthroughs on specified tasks;
- usability guidelines for heuristic artifact evaluation;
- prototype testing including task performance observation and thinking-aloud.

The DSD approach to be presented below assumes and encourages the use of the results from an appropriate selection of such methods during artifact development and therefore is in no way intended to replace them. The same holds true of the relationship between DSD and Level 3 approaches (see below).

At *Level 2* we find the rapidly developing area of Design Rationale (DR) approaches to design space analysis (e.g., MacLean et al. 1991a,b, 1993) as well as approaches to representing overall design space structure in conjunction with DR such as DSD (Bernsen 1993a,b,c). These approaches aim at facilitating and making explicit design commitments and design reasoning in a way which is systematic and natural to use and re-use by designers.

At *Level 3* we find the science-based approaches to usability engineering such as:

- task analysis: GOMS (Card, Moran and Newell 1983), TAGs (Payne and Green 1986), CCT (Kieras and Polson, 1985), etc.
- cognitive walkthroughs (Lewis et al. 1990);
- claims generation support (Carroll and Rosson 1992);
- user modelling (e.g., Barnard and May 1993); or
- modality theory (Bernsen 1993d,e, 1994a).

Common to the approaches at Level 3 are that they seek to provide analytic and theoretically based leverage on the discovery of and solution to design problems to do with usability. The approaches at Level 2 share this aim but in a particular way, as will appear shortly. As we move downwards in the hierarchy of levels, from actual design processes towards the science base of HCI, the more unsolved problems we find. Level 1 includes the approaches which are currently being used in design practice. If an appropriate subset of these were to be generally applied during systems design processes, there is no doubt that artifacts would be much more usable than is currently the case.

Roughly, the Level 2 approaches can be seen as providing added *systematicity, structure and explicitness* to design processes while the Level 3 approaches are intended to supply the added *problem-solving power* which may be expected from applying relevant scientific theories of tasks, users, representational modalities, etc. The design representations at Level 2 can to a large extent be developed and used in practical design independently of developments at Level 3, just as the Level 1 approaches are already in actual use independently of current development efforts at Level 2. Once we have mature approaches at Level 2, these may be expected to seamlessly integrate with Level 1 approaches in actual design practice. Moreover, it seems reasonable to assume that the Level 2 approaches to explicit representation of design space structure and development constitute useful bridges for the realistic application of the science-based approaches to usability engineering at Level 3. An example is the use of DSD for representing task and task domain information as a basis for applying Modality Theory in the Information Mapping Methodology (Bernsen and Bertels 1993, Verjans and Bernsen 1994, Verjans 1994).

DSD: Structure, Process and Reasoning

The usability engineering of artifacts is a process of creative problem-solving. Once a more or less precise and tentative, overall design goal has been established, this problem-solving process unfolds as a path through the design (problem) space defined by the design goal. Arguments and problem solutions represent design commitments. DSD is intended to add some measure of systematicity, structure and explicitness to this process by providing

(i) a general initial structure to design spaces,
(ii) a method of incrementally capturing the design commitments made during the design process by mapping these into the general design space structure, and
(iii) a simple representation of the design reasoning underlying particular design commitments.

DSD and the *Structure* of Design Processes

A general initial *structure* to design spaces is important as designers work within a design space that contains invariant aspects which should be taken into account in the design process, since user performance is a function of design commitments made with regard to these invariants, and the optimisation of user performance with the intended artifact is the aim of cognitive engineering. By working within the confines of a DSD frame, none of these invariants should be overseen. In this way, the application of DSD helps to contain and guide the route that is taken through any one design (problem) space. The assumption that DSD is making is that artifacts are complex products of an optimisation process which, i.a., take into account such factors as COSITUE (Cooperation, Organisation, System, Interface, Task, User and user Experience). DSD has been developed as a top-down framework for articulating the design space in systems design from the point of view of usability engineering.

The individual COSITUE elements referred to above constitute one set of invariant aspects of design spaces. COSITUE is just one of three parts of the DSD design space structure representation. The two other sets of aspects are the general design goals and the general constraints and criteria on the design process (see Figure 1 below). Figure 1 presents the DSD frame, which is a structured template for representing design spaces. The categories in the frame cover those invariant aspects of all design spaces, to which design teams need pay heed, in terms of usability engineering. What happens during an actual design process is that design problem-solving and decision-making gradually specifies the intended artifact as a particular solution to the design goals within the invariant design space structure. DSD aims to make explicit this incremental process or important parts thereof.

Generating Commitments

Since DSD is being developed as a usability engineering support tool, what should be entered into a DSD representation or frame is information which is relevant to the usability of the artifact to be designed. That the information is thus relevant means that it constitutes actual or potential arguments in design problem-solving to do with artifact usability. Such information - design commitments - seems to be of two broad types, factual and evaluative. *Factual information* comprises constraints on the design process, such as the general feasibility constraints (see further below), which have to be accepted as matters of fact by the design team as well as design commitments that the artifact is to have specific properties. *Evaluative information* comprises the usability-related constraints or criteria used by the designers to evaluate specific design options concerning the properties of the designed artifact. During design, information belonging to these two broad classes accumulate, act as mutual constraints in design problem-solving and thereby determine the properties of the designed artifact.

Figure 1. The DSD Frame.

Design Project:		
DSD No.	**Date:**	**Sign:**
A. General constraints and criteria		
Overall design goal(s)		**DSD No.**
General feasibility constraints		
Scientific and technological feasibility constraints		
Design process type		
Designer preferences		
Realism criteria		
Functionality criteria		

Usability criteria	
B. Constraints and criteria applied to the artifact within the design space	
Collaborative aspects	
Organisational aspects	
System aspects	
Interface aspects	
Task aspects	
User and user Experience aspects	
C. Hypothetical issues	
D. Documentation	
E. Key:	DSD No. (n) indicates the number of the current DSD frame. ‘Null’ means that the artifact does not embody a certain aspect of DSD. Italics indicate new elements in DSD (n) as compared to DSD (n-1).

To date, DSD has been applied to rather different design processes. Firstly, DSD was applied in a large-scale generic technology design process, namely a spoken language dialogue system, intended for use in the air traffic information system (ATIS) domain (Bernsen, 1993a). The design team in this case comprises three geographically distributed groups in Denmark. The first prototype of the system is currently being tested. DSD will be used during the development of prototype two.

A second application of DSD has been in the development of the RAVE and Portholes applications (Bellotti and MacLean 1993, Dourish and Bly, 1992, Bernsen 1993f), two experimental systems developed at Rank Xerox EuroPARC, which connect researchers by providing them with ‘general awareness’ of each others’ availability for communication.

DSD is being applied to the re-design of a Taxonomy Workbench at the Centre for Cognitive Science, Roskilde, Denmark (Bernsen, Lu and May 1994). This is a design process with four design goals. Firstly, there is the presentation and illustration of Modality Theory, secondly, the facilitation of the conceptual analysis of unimodal and multimodal objects by enabling the user to build large databases of examples. The third goal is the assisting of experimental classification of the objects in the data-base. And fourthly, the advancement of the Taxonomy Workbench system towards its being used as a practical design tool.

DSD is also being used to succinctly represent task domain information as part of the Information Mapping Mehodology. This methodology serves to (i) make explicit the information to be represented and exchanged between a system-to-be-designed and its intended users and (ii) map this information onto suitable input/output modalities during interface design, using Modality Theory to support the mapping process (Bernsen and Bertels 1993, Verjans and Bernsen 1994, Verjans 1994).

In the course of the next few pages, which provide a detailed introduction to DSD, examples will be taken from the application of DSD in the ATIS domain, after which we will discuss two large-scale design process case studies, the purpose of which was the further development of DSD.

Structuring the Space: *What* to Represent and *Where*

Once it has been decided to represent a design commitment in DSD, the question arises about where to enter it. The categories of the DSD frame provide a way of structuring the accumulating factual and evaluative constraints upon a design process. Part A of a DSD frame, the 'general constraints and criteria', comprises general up-front information on the design process whereas Part B of a DSD frame, 'constraints and criteria applied to the artifact within the design space', comprises more specific information on the designed artifact as it evolves. Given the iterative nature of design processes, this distinction between Parts A and B is logical rather than temporal. One may assume that Part A information is being entered into DSD early on in the design process whereupon the designers' focus would shift to Part B. The reason for this assumption is that Part A information is quite fundamental to the design as a whole and often also is available at the start of a design process. However, Part B information may also be entered quite early and later additions of Part B information may require additions or revisions to the information contained in Part A.

The individual entries of Parts A and B of a DSD frame should be incrementally filled with information according to the general guidelines above. The sections below provide specific guidelines on how to do this for each individual entry. The guidelines consist of (a) a general characterisation of the information belonging to a specific entry, (b) illustrations from our own work and (c) comments on the entry's relevance to usability engineering and the revisability of the information it contains.

Overall design goal(s)

The overall design goals specify the nature of the artifact to be designed. The overall goals of a design process guide and constrain design problem-solving. It is not possible to make a commitment to a set of overall design goals without having been through a process of evaluating their feasibility and possibly also their mutual consistency. For instance, it took an entire design meeting to verify that the four overall goals of the Taxonomy Workbench re-design project were in fact both feasible and mutually consistent. Even in stable environments, overall design goals are not immune to revision during the design process, especially not in high-risk design processes such as explorative design and generic technology design.

General feasibility constraints

General feasibility constraints impose broad limitations on what can be achieved during the design process. Few design processes have no general feasibility constraints worth mentioning. In all other cases, general feasibility influences design decision-making and the setting of overall design goals in important ways, including the effects of budget, time limits, available manpower, hardware and software resources, social and legal constraints, etc. For instance, in the ATIS spoken language dialogue project mentioned above, which is in excess of 20 man-years of effort, budget constraints forced the focussing of the development effort on proving points of generic technology while reducing effort on comparatively trivial extensions of the system's capabilities, such as larger vocabulary or more flight destinations. No argument is needed to show that general feasibility constraints may function as high-level usability arguments, such as that, e.g., the absence of resources prevent the designers from pursuing a promising alternative discovered in the course of design discussions. It is often possible during initial design specification to explicitly capture all or most relevant general feasibility constraints on the design process. Revision immunity, however, is another matter which often cannot be ensured.

Scientific and technological constraints

Scientific and technological constraints are limitations on design processes due to lack of scientific knowledge and technology, respectively. Many design processes have no general scientific or technological constraints as the technology necessary to achieve the overall design goal(s) is clearly in place and available within the feasibility constraints on the design process. However, information technology continues to hold a vast potential which cannot be exploited because the scientific basis is not there yet or because appropriate supporting technologies and tools have not been developed. The ATIS dialogue project provided many illustrations of this point. For instance, because of speech recognition technology limitations, the user input had to be limited to an average of 4 words per utterance in the first prototype. This limitation continued to influence usability reasoning throughout the design process, forcing a large number of trade-offs on the design, and thus showed how scientific and technological constraints may have strong effects on usability. Scientific and technological constraints interact heavily with the overall design goals and general feasibility. It is often possible during initial design specification to explicitly capture all or most relevant scientific and technological constraints on the design process.

Type of Design Process

By design process type we refer to the differences between, e.g., exploratory, demonstrator and commercial design, or between different types of commercial design process. Also, any such process may be either design from scratch or re-design. The nature of the design process strongly influences design problem-solving and decision-making. For instance, in generic technology design, such as the ATIS dialogue project, while still aiming at maximally usable systems, one can afford to ignore trivial extensions of the system's capabilities even though these would be crucial to the usability of any commercial system developed on the basis of the generic technology. In other words, the design process type may strongly influence the types of usability issues it makes sense to address during design. The design process type is closely related to the overall design goals and normally remains stable from the outset of the design process.

Designer preferences

Designer preferences are the designers' choices of hardware or software platforms, development tools, programming languages, particular combinations of modalities for the expression and exchange of information between system and user, etc., as long as these are not being dictated by other constraints. Thus, designer preferences are less mandatory than, and different from both general feasibility and scientific and technological constraints. Such preferences may sometimes strongly constrain reasoning on usability aspects of the designed artifact. In the ATIS dialogue project, the design team decided early on to use the DDL (Dialogue Description Language) tool for the implementation of the dialogue model although DDL had never been used for this particular purpose. The implementation process demonstrated a number of important limitations of DDL such as, e.g., the impossibility of representing the entire history of the dialogues conducted between user and system. This again means that, when more advanced dialogue models are to be implemented, DDL will either have to be significantly extended, or replaced. Designer preferences can normally be stated early on in the design process but may have effects that enforce their revision.

Realism criteria

Realism criteria are concerned with the complex and highly contextualised question of whether the artifact to be designed will *effectively* meet real user needs. The point is that an artifact may fail to do so for many different reasons and, if it fails, the development effort may have been wasted. It may be too expensive for the intended customers, an entirely different combination of input/output technologies may have superior usability, the artifact may require a hardware platform which the intended users cannot be expected to have or want, the intended artifact may not be superior to traditional work practices or may be inferior to already existing systems on the market. In other words, the realism criteria should ensure an evaluation of the design process to be undertaken which makes it worthwhile. In the ATIS dialogue project, the following realism criteria were apparent:

- The artifact should meet real and/or known user needs;
- The artifact should be preferable to current technological alternatives;
- The system should run on machines which could be purchased by a travel agency;
- The artifact should be tolerably inferior to the human it replaces, i.e., it should be acceptable by users (to be expanded in the usability criteria) while offering travel agencies financial advantage;

The realism problem affects all design process types albeit in different ways. For instance, exploratory design often is design in search for one or several realistic applications, which means that the realism criteria have to be more abstract than in commercial design which normally has to provide a strong justification for the realism of the intended artifact before getting started. Realism criteria are clearly linked to usability. However, they are of a wider scope than usability criteria because even the most usable artifact, i.e. an artifact which satisfies all thinkable usability criteria, may fail as an eventual product if its realism is flawed. Conversely, products which are far from being optimally usable may be realistic as long as the end-users do not assign high priority to them. Many commercially available spell-checkers illustrate this point. Like the overall design goals, general feasibility, scientific and technological feasibility, design process type and designer preferences, with which the realism criteria interact, the realism of the intended artifact is something to be meticulously decided on at the start of the design process. Still, this does not prevent later additions and revisions occurring to the realism criteria.

Functionality criteria

Functionality criteria essentially point to the need to make sure that it should be *possible* for the intended users to (somehow) carry out all the tasks which the artifact is intended to support. Like many of the other criteria and constraints in Part (A) of a DSD frame, the functionality criteria will often support the generation of more specific design constraints in Part (B) of DSD. As any designer would testify, actually meeting the functionality criterion and its descendants in Part (B) of DSD is one of the major tasks during the design process. The following example of possible entries in a functionality field in a DSD frame is extracted from the air traffic control case study described later in this paper. The artifact should support the main user task, namely: managing the approach of aircraft into major airport complexes. This involves the supporting of the following top-level sub-tasks:

1- to assign a two-letter code to a flight (Special Category Indicator);
2- to reposition flights in the SAS (Stable Approach Sequence);
3- to resequence the order of flights within the SAS;
4- to swap the places of flights in the SAS.
5-to tidy the screen.

Usability criteria

Arguably, usability engineering and HCI in general exist because artifact functionality is not *sufficient* to usability. From a usability point of view, it is not sufficient that the artifact makes it possible for users to carry out their intended tasks. DSD is intended to support optimisation of artifact usability and hence the creation of artifacts which not only, somehow, allow task completion but do so without creating major usability problems. For instance, continuing the example above, sufficient task domain coverage is not sufficient to usability. The system may indeed possess the information needed by users but may not, e.g., make this information available when they need it during their task. When the designers discovered that this was in fact not the case, this usability problem was solved and the following usability constraint added to part (B) of the DSD frame:

- It should be possible for users to fully exploit the system's task domain knowledge when they need it;

In the design of intelligent, e.g., plan-based software, meeting this constraint can be a major problem. Standard help systems for users normally meet the constraint but fail to meet the more stringent constraint:

- It should be easy for users to fully exploit the system's task domain knowledge when they need it;

- for which reason standard help systems are often virtually useless.

To sum up this presentation of the entries belonging to Part (A) of a DSD frame, this part of the frame sets the stage for later, more detailed usability design reasoning by explicitly representing relevant information on the overall design context and aims. Although DSD information can often be re-used from one design process to another, this should be done with circumspection. In principle, DSD does not represent general design guidelines but rather contextualised information on the design process at hand. So the transfer of information from design process X to design process Y should rather serve as an occasion to identify important differences between the two design processes, which may then be reflected in differences between their accompanying DSD representations. It is our hypothesis that most of the entries of Part (A) of a DSD frame will turn out to be reasonably unambiguous in practical use. One possible exception are the general usability criteria which anticipate the more detailed concern with usability in Part (B) of DSD. One general guideline which may help prevent a proliferation of usability criteria in Part (A) of DSD is to only use this part of DSD for representing quite general criteria whose implications and subsumptions may then be stated in Part (B).

C:, O: Collaborative and Organisational aspects

It is imperative that attention needs to be paid to collaborative/organisational aspects, as artifact development and use normally does not occur in a vacuum but, at least, within some organisational setting or other. Moreover, today's work practices have moved away from the one user-one system paradigm, to that of multiple user-multiple systems. Computer systems more often than not support distributed collaborative work, thus one needs to bear in mind user status, accessibility rights and protection of privacy, within organisational hierarchies. An example of a collaborative aspect would be a limitation on the number of users to whom it is possible to make AV connections simultaneously. By contrast, an organisational aspect would be the issue of protection of privacy via control of accessibility. This genre of information should be entered against the C and O sections of DSD frames.

S: System aspects

System aspects further develop the functionality aspects of Part A of the DSD frame and hence requirements as to what the system under development should be able to do. A natural extension of the system aspects part of the DSD frame would be to include specifications on *how* the system should do what it is intended to do, thus bringing the DSD representation closer to software specification. This latter part of the system specification would normally happen in tandem with, or subsequent to, the task specification discussed below. System aspects can be exemplified by way of the following, taken from the spoken language dialogue system. In the S section, the following entries were made.

- 500 words vocabulary;
- Large enough task-related vocabulary;
- Natural grammar;
- Appropriate semantics;
- Natural discourse handling;
- Limited speaker-independent recognition of continuous speech;
- Close-to-real-time response;
- Take users' relevant background knowledge into account;
- Take into account possible (and possibly erroneous) user inferences by analogy from related task domains.

Many of these system constraints were actually derived from user considerations. Thus, for instance, the natural grammar requirement derives from the fact that compromises with the grammar which is natural to users in the task domain are likely to render the system virtually useless.

I: Interface aspects

Interface aspects should be distinguished from system aspects, in so much as interface aspects are concerned with the "front line", namely everything that interactively engages the user. An example of an interface factual commitment would be that a touch screen is being used, perhaps as the users are too young, or unable to use keyboards, or for other reasons as in the case of CERD to be discussed below. Some interface constraints, such as the decision to include a touch screen, would normally be decided on early in the design process, whereas others, such as the detailed interface layout, would normally await an analysis of users' tasks.

T: Task aspects

Under this heading come the tasks to be done by the user, and by the system, as task accomplishment is interactively shared between system and user. These tasks would optimally come from a task analysis of the work domain. In addition, it is necessary to make explicit the *parameters* of these tasks i.e. *how* the task should be carried out, for example, in the ATIS project, the user should use short sentences (max. average 4 words).

U: User and user Experience aspects

Entries should encompass the user profile, including such aspects as physical and mental handicaps, novices, intermediates or experts, and special user groups, such as Danish speakers.

Hypothetical Issues

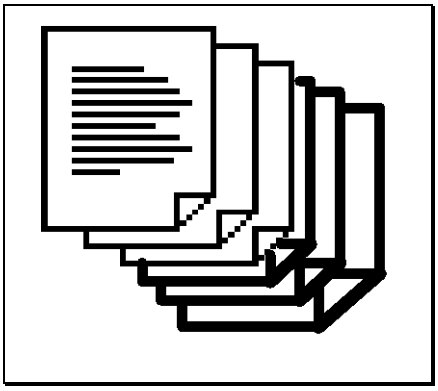
This section allows the noting of questions that have been left open or which came to be discovered during a certain stage of the design process, 'notepad' exchanges between the designers, etc.

Documentation

What to enter in a DSD frame is orthogonal to the level of detail of the information entered. It should be remembered that the most important design decisions need to be represented in the frame. These are often based on additional, supporting documentation as well as the criteria and constraints at play. Documentation may include personal and group notes, sketches, email messages and faxes between colleagues, detailed specification structures too comprehensive to be included in DSD itself, etc. DSD comes with supplementary documentation and does not have to capture what is in there in full detail. But the problem of detailed representation remains a difficult one. For instance, in the Dialogue project the reasoning behind all the steps taken in developing the dialogue model was not represented. It was decided to represent in DSD the criteria applied in problem-solving when difficulties were encountered. We need to consider the level of detail that is of use to designers, and also that there will exist individual differences across designers in this. Experience in using Questions, Options and Criteria (QOC), MacLean et al. (1991a) has revealed that for archival purposes, QOC is normally complemented by more extensive notes and detailed documents, and as a background document on its own, QOC is insufficient. The same may hold true for DSD.

DSD and the Design *Process*

Figure 2. Overall Design Process Framework.



Use of DSD results in a numbered series of DSD frames (Figure 2 above), accompanied by the relevant supporting documentation. In a completed design process, the final frame in such a numbered series will represent the artifact to be implemented (at least) as far as its usability aspects are concerned. Thus, DSD lends not only logical, but temporal structure to the design space. From its inception, design problem-solving takes place *in context.* In DSD terms, this means that problem-solving at a certain point during design is constrained, not only by the overall design space structure but also by the specific design commitments which have already been made. At any stage during design, the most recent DSD frame serves to explicitly capture the context within which new problems are to be addressed. However, design processes are notoriously *non-linear* in nature, a feature which is both recognised and supported by DSD, as is outlined in the following section.

DSD and *Reasoning*: Integration with Design Rationale Representation

We have already seen that DSD is about structure and process during design. Using DSD lends *structure* to the design space by way of the semi-formal notation which encourages the designer to consider the space in terms of a set of invariant elements. The technique's methodology supports the design *process* as an incremental gathering of design commitments as the design space develops. The third part of the story is that DSD is about *reasoning*. DSD provides a notation for the recording of design *reasoning* or problem solving, and its accompanying justification. System design often takes place in a changing environment, which may lead to sudden changes even in the most basic design commitments made so far, such as the overall design goal(s). And even if such changes are not imposed, it is rarely possible to guarantee consistency across early design commitments and those made later on. Design reasoning further down the line may enforce revisions of commitments made earlier in the design process. Given the fact that design commitments, both factual and evaluative, behave as interacting constraints, this process of revision may in principle affect any earlier design commitment. It is in this way that DSD supports iteration.

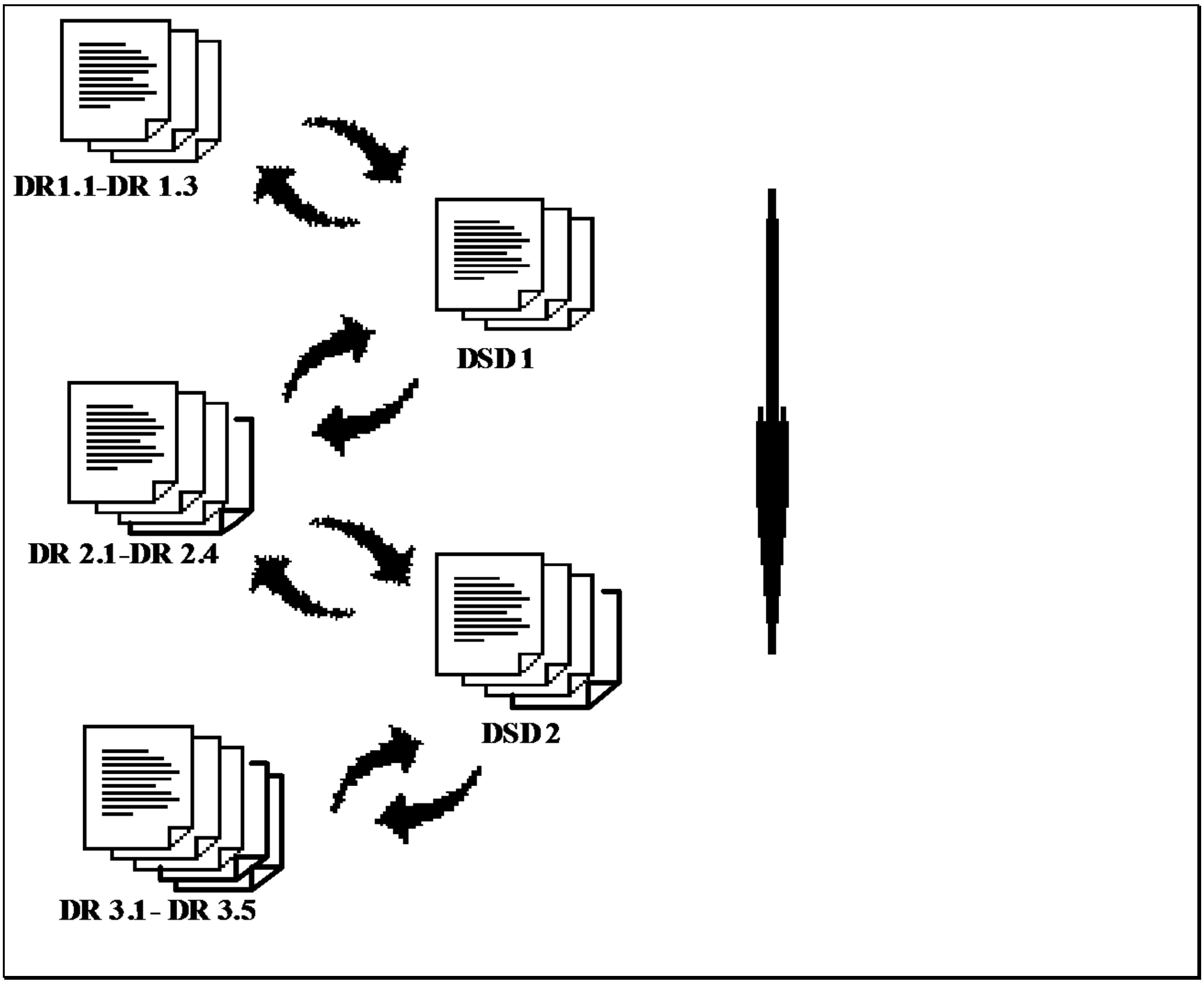
DSD and DR are complementary, in so far as DSD provides a top-down framework for making explicit the most general aspects, criteria and constraints characterising the design space, and enforces a consideration of these from the point of view of usability during artifact design. We have so far seen that it provides a means of explicitly and concisely capturing the most important design decisions, and hence the context for the design reasoning represented in the DR frames.

Design space analysis using Questions, Options and Criteria (MacLean, at el. 1991a) has little to say about what domain information is relevant, or should be represented. A DSD representation partially explains where the criteria used in a DR at a certain stage during design, come from. Combining DSD and DR allows the description of a design process as a path through the design problem space taking the following form:

(DR1 - DSD(1)) - (DR2 - DSD(2)) - etc.

This is represented graphically in Figure 3 below. The bi-directional arrows reflect the non-linearity of the design process.

Figure 3. Integrating Structure with Reasoning.



Design commitments represented in DSD can be incomprehensible without their DR representation. DSD essentially involves the *tracking* of problem-solving, and the binding of the user of the technique, to the way in which problems were solved. Thus, a series of DSD frames make explicit the gradual shaping of the design space around the artifact, and makes explicit relevant designer commitments with respect to overall design goal and general constraints and criteria on the design process as illustrated below. However, DR should only be used when designers "stumble" and need explicit analyses of complex trade-offs or when cost-benefit terms indicate that a more thorough DR analysis is feasible, such as in safety-critical systems development.

Whilst DSD frames provide a representation of the usability-related commitments made during the design process, our DR notation (Figure 4 below) explicitly represents the reasoning which generates new design commitments. It represents the *design problem*, the *option(s)* considered for its solution, the design *commitments* applied in attempting to solve the problem through rational selection between options, the *resolution* of the problem and the *justification* for the solution. Through making explicit the justification for each solution as part of the notation, we are giving closure to the design rationale process, in a way previously neglected by other DR notations such as Questions, Options, and Criteria (MacLean et al., 1991a). Design reasoning can range from acceptance of the obvious solution to a design problem, given its context as captured in the DSD representation, through to a laborious process of explicit design reasoning in which several different options are generated, evaluated against the commitments already made as well as newly discovered commitments, and traded off against each other again referring to commitments already made as well as ones newly discovered. The ‘winning’ option becomes a new design commitment entry in an update of the most recent DSD frame. The same holds true of the evaluative commitments used in resolving the design problem, in so far as these commitments did not already appear in the most recent DSD frame.

The *comments* section allows the statement of unclarities and open questions by the design team. The final line in the DR frame summarises the outcome (i.e. the chosen design option and new evaluative design commitments, if any), which will be carried forward into the subsequent DSD frame. In addition to the above, more context is added by way of slots for number of the DR frame, if there are several, the date, and into which DSD frame the results of the DR will go. Links to other DRs reflect the non-linear nature of the design process and thus support the process by linking the current frame backwards and forwards to other design decisions.

Figure 4. The DR Frame.

Design Project:		
Prepares DSD No.	**DR No.**	**Date:**
Design problem		
Options		
1		
2		
3		
4		
5		
Commitments involved		
1		
2		
3		
4		
Resolution		
Justification		
Comments		
Links to other DRs		
Documentation		
Insert in next DSD frame		

From Constraints Comes Direction: Reasoning within Structure

The use of DSD frames encourages the designer to consider requirements *in terms of* the design space's relevant high-level constraints, collaborative and organisational aspects, system and interface aspects, task and task domain aspects, and user and user experience aspects. An attempt to satisfy requirements stipulated by these factors should, we anticipate, be conducive to the development of an interface whose usability is grounded in reality. It should be emphasised, however, that although any design team operates within a set of constraints, it is the individual commitments within the DSD frames that are directive in the normal sense of this term, rather than the whole context formed by these commitments. The context as a whole is what increasingly constrains the design space. Ideally, the content of DSD frames, when supported by display sketches, presents the design team with precise and succinct descriptions of displays and functionality, and a complete task breakdown. As such, it forms a basis and a focal point for design sessions and the subsequent decision making. Important and difficult design problems, in particular those which address difficult trade-off issues, may additionally be represented in the DR frames which form an integral part of the DSD approach. In safety-critical applications, such as one of the design process case studies to be described, it might pay off to represent a larger-than-usual part of the design reasoning in these DR frames.

PART TWO

THE CASE STUDIES

Developing DSD: An Introduction to the Case Studies

In the introduction to this paper, we mentioned that DSD has already been applied to a diversity of design processes. In this section, we introduce two large case studies. These case studies are the EuroCODE design project, (ECOM) (Bellotti, 1994) and the Computer Entry Readout Device project (CERD) Buckingham Shum, Duke, Hammond, and Jørgensen, (1994) . Both of these case studies represent an important landmark in the development of DSD, namely the first attempts at integrating DSD and DR within design processes.

OVERVIEW OF THE EUROCODE DESIGN PROJECT

The EuroCODE design process entailed the development of an example CSCW system, integrating the use of text, audio and video in a high bandwidth multimedia digital network. The design team are developing suitable, sophisticated collaboration technologies to support the builders of the Great Belt bridge in Denmark. The system is to support both planned, unplanned and formal collaboration and interaction between geographically separate colleagues on the Danish Great Belt construction scheme. This scheme involves building the world's largest spanning bridge which will connect two of the main islands of Denmark, Funen and Sealand. Designers, engineers, administrators and others distributed over a wide area have to co-ordinate their work to ensure that the bridge is built according to plan. Sometimes communications or meetings must take place within a fixed period but it is not always feasible for people to travel. This means that modern multi-media communications technologies are likely to prove useful in reducing traveling needs or increasing the richness of remote communication.

EuroCODE is a demonstrator application of a generic exploratory design process into innovative technology which had already resulted in the RAVE, Portholes and Cave experimental systems at Xerox EuroPARC, Xerox PARC and in Toronto. The EuroCODE designers were thus working in a context within which certain kinds of innovative technology have been applied to design domains to see what kinds of benefits they provide. Thus, the technology predated, drove and constrained the design project. The users of the system at the Great Belt would be diverse, ranging from administrators through local engineering supervisors to remote designers, consultants and contractors (supplying parts from remote production lines in other European countries for the bridge). Supervisors and contractors need easy access to drawings and the ability to find out quickly what changes have been made. They also need to be able to discuss design changes at a distance, e.g., when one person is on the bridge site and the other is at a design office on shore. A number of people on site in Denmark need to know the current status of work at remote sites, e.g., the girder construction site in Livorno, Italy. There are also groups of people, such as the safety group or the surveyors, who perform the same function but at different locations and need to co-ordinate their activities. A “day-in-the-life” scenario illustrating what it might be like for a supervisor to use novel technology such as AV connections systems in a standard working day was made available to us. This scenario (Appendix 1) was based upon studies of current practices and requirements carried out by Aarhus University in Denmark, and the design team.

The Advancement of DSD Through the EuroCODE Case Study

Our top-level goal was the application to DSD to the EuroCODE design process, with a view to gleaning as much insight as possible, into as many facets of DSD as possible, with an especial interest in how the integration with design rationale would be handled in practise. This section introduces these insights into the Design Space Development technique, both at the macro-level, and then in more specific detail, by relation to examples.

A Reconstructive Exercise

Perhaps the most general discovery made during the EuroCODE problem was that DSD is not optimally applied to retrospective, remote design processes whose progress must be communicated through documentation alone, rather it is the nature of DSD to best lend itself to being tested and used in ongoing, "in house" design practice where we can track its day-to-day use. The authors were based in Denmark, and our case study collaborators were located in England. We were aware that there existed a prototype interface which had been rapidly developed by them at Rank Xerox EuroPARC. However, since techniques such as design space development are not used in present-day standard design practice, there naturally existed no DSD design space representation which might form the baseline of our work on the EuroCODE exemplar design issues. Bearing in mind that DSD is about, amongst other things, reasoning within the appropriate design space structure, one had to be compiled or, rather, reconstructed in order to represent the state of the EuroCODE design process immediately before the designers began developing the interface. This reconstruction exercise began with the amassing of factual and evaluative commitments acquired from a reading of the EuroCODE background documents. Domain information found to be lacking was obtained through e-mail communications, telephone conversations and a transcription of an interview with the main EuroCODE designer about the ECOM (EuroCODE) prototype interface. Having gleaned as much information as possible from the various sources mentioned above, it was nonetheless evident that this exercise, of necessity required the divining of certain implicit assumptions under which the design team were working.

A set of commitments emerged, of a factual nature (e.g. budget, deadlines, and the use of the ATM communication protocol), and of an evaluative nature. Evaluative commitments take the nature of, e.g.: "The artifact should fully exploit the functional opportunities of high-bandwidth communication links", or: "The artifact should support the main user tasks intended, namely:

1. To make and break a variety of audio-visual connections;
2. To specify levels of availability;
3. To control signal quality and, for ISDN lines, cost."

The second step in the reconstructive founding of the baseline DSD (n) frame was essentially one of boot-strapping the outline DSD (n) with our emerging DR representation of the problem space. A set of generic tasks were pulled out of the background information available. These tasks, which are to be ultimately supported by the media space, became the focus of what are termed 'design problems' in the DR representations which follow. These tasks were decomposed into their constituent sub-issues (specific problems to be solved), listed against potential solutions (options), and judged in terms of the commitments present in DSD (n), and new commitments springing to light in the problem solving process.

Some of these newly-emergent commitments were of a level detailed enough to warrant realistically having emerged at this stage, nonetheless it was felt that others which had become apparent during problem-solving would more realistically have been inherited from DSD (n). Thus, the DR was boot-strapped back to DSD (n) by relocating these would-be inherited commitments. It should, however, be stressed at this point, that this boot-strapping exercise is a function of the artificiality and non-specificity of the modelling exercise as compared to the use of DSD in representing ongoing design processes, and is not part of the DSD methodology. The remote and retrospective nature of the design process only allowed us to *hypothesise* that these factual and evaluative commitments, as present in the baseline DSD design space representation (DSD (n)), were close approximations to the commitments taken on by the ECOM designers in addressing the ECOM design problems. Hypothesising, extrapolating and approximating are necessary in the development of design processes of this nature, but for DSD to be of optimal power, this margin for misinterpretation should be minimised.

From Tasks in DSD Frames to Problems in DR Frames: An Overview of the Integrational Process in Practise

One of the first issues elucidated by the EuroCODE application, and briefly flagged in the above paragraph, was that of "where do the problems to be solved come from (i.e. the questions to which one needs answers in order to design a working artifact) and how do they fit into the DSD process?" We looked to the top-level tasks (problems) we were initially given, such as, e.g., the following:

-the system should allow the user to specify his/her availability;
-the system should allow users to make various task-relevant connection types;
-the system should allow users to control bandwidth.

We saw these as top-level commitments to a range of functionality that we had to stand by. These commitments were broken down into sub-problems, for example, the control of bandwidth could be broken down into (a) offer minimum cost feedback on signal transmission, and (b) communications should be of an acceptable minimum quality. The problems then emerge as how to do (a) and how to do (b). Once this had been done for all of the top-level problems, the sub-problems were taken forward into DR frames. These DR frames make reference to the specific problem at hand, the options available, the commitments involved (taken from the previous DSD frame), the selected option, and the justification for having selected that option, in terms of commitment satisfaction. The option which has been chosen as best solving that particular design sub-issue, will then go forward into the next DSD frame for its parent issue.

To track this process more precisely, we illustrate by example from our EuroCODE application (Bernsen and Ramsay, 1994a). Problem (b) that was mentioned above, generates the sub-problems we find being dealt with in DR No. n+2:4.5 and DR No. n+2:4.6. The solution which resulted from DR n+2:4.6 ("sender simultaneously controls quality at both ends of the connection"), was duly entered into DSD No. n+3, as did that which resulted from DR No. n+2:4.5. When DSD n+3 is considered, this entry may spawn further detailed "off-shoot" problems. These further problems would then be inserted into DR frames, to undergo the problem-solving process.

So, in this way, DSD has provided the appropriate context for pursuing the problem solving process. The DSD frame encouraged a consideration of the tasks to be supported, from the point of view of usability. You are forced to unpack the relevant commitments to functionality (overall tasks to be supported) into sub-problems, and also the relevant evaluative commitments, such as that the control of connection quality should exploit a full set of useful opportunities offered by high bandwidth, offer minimum cost feedback on signal transmission, and that communications be of an acceptable minimum quality. Through this systematicity, control is maintained over the appropriateness of the problems addressed paying attention to the relevant commitments. In Appendix 2, we present our baseline representation (DSD n) of the design space to be developed.

Separating User and System Tasks

One issue that was clarified in the course of this application of DSD was the distinction between what is a user task and a system task. This had previously been implicitly obvious, yet the structuring of the space forced a reconsideration of how to handle these two classes of entry.

Task execution and goal attainment is essentially shared by user and system, as the system supports the user. Initially, we had referred to the list of user tasks in the “Task” section of DSD frames, and found that we were merely echoing this content under "System", by prefixing the user tasks with phrases of the type "the system should *support* the user in, for example, making and breaking connections. In so doing, we were introducing redundancy. It emerged that the requirement that the system support the user in task completion is only part of a larger picture. In addition to supporting user task completion, the system should give feedback to the user on when and what information is being captured and to whom the information is being made available, notify users of device conflicts, connection failure and inappropriate connection attempts, and convey to the user current connection status and possible connections. The system also needs to ensure that the user selects the correct connection, and that the user gets the connection expected.

Tasks, Functionality and Usability

If possible, an ordering of tasks from the “Task” section of the frame should allow the identification of a set of top-level, or main user tasks. These should then be imported into both the Functionality and Usability sections of the DSD frame, and considered in terms of such.

Where the Factual and Evaluative Constraints Come From

A question that may feasibly be posited is that of where the constraints, especially the evaluative, come from. Factual constraints, such as time limits and resources available are more evident in origin. Evaluative constraints may be drawn from heuristic lists available in the literature (e.g. Nielsen 1993), previous designer experience, common-sense, or relevant theory. They may not always be evident at the outset of a design process, but be emergent as the space is developed. The incremental amassing of new commitments will be treated in more detail in the sections to follow which discuss the proposed application of these commitments in the design reasoning process.

Using DSD as the Basis for Reasoning

As outlined in the overview of the Integrational Process in Practise section, tasks that are listed in DSD frames are "turned around" into problems to be solved in the design rationale representations. The second aspect of DSD frames that are imported into DR frames that build on them, are the relevant constraints or commitments. Figure 5 below shows exerpts from a DR frame, and illustrates our usage of commitments in such frames. This was part of a DR made in preparation of DSD (n+1), and was called DR No. n:2.3 (i.e. top-level problem number 2, sub-problem 3).

Figure 5. Options and Commitments.

Options	
1	Mouse-click on a button, permanently visible on the screen, labeled with the connection type (RAVE, EuroCODE).
2	Select from a pull-down menu.
3	
Commitments involved	
1	The making of connections of any type should be as simple as possible in steps and procedure.
2	Avoid screen clutter.
3	Permanent visibility on the screen of important functions, if possible. NEW.

From DSD n we know that the parent problem here is that of how to select audio-visual connections. Two options have been generated. Option 1 is already a design implementation in another AV system we were aware of, called RAVE, and the EuroCODE prototype. A second option was suggested, but in the context of the relevant commitments, this second option was rejected, for reasons laid out in the justificatory section of the frame (Figure 6). In this justification, we make explicit reference to the commitments we were dealing with, whilst resolving the problem.

Figure 6. Justifying A Solution.

Resolution
Option 1.
Justification
Buttons are quicker and easier to manipulate than pull-down menus. Few buttons will be needed for connection type specification as only a limited number (i.e. 4) of connection types are being offered to users. This means low screen clutter. Furthermore, the continuous and static presence of the connection-type buttons on the screen allows users a permanent overview of the types of connections offered by the interface. Such an overview supports deliberate decision-making.

In the course of solving this design problem, another design commitment was generated, namely that of "Permanent visibility on the screen of important functions, if possible" (Figure 7). This commitment, non-existent in any frame until this point, will be entered into the next DSD frame. However, until the point at which this will happen, the new commitment only appears in the DR frame from which it was generated. For this reason, we may later find links to this DR frame being referenced in the "Links to other DR frames" sections of other DR frames since later DR frames may employ this newly-generated commitment.

Figure 7. Accumulating Commitments.

Insert in next DSD frame
Specify connection type by mouse-clicking on a button, permanently visible on the screen, labeled with the connection type. Permanent visibility on the screen of important functions, if possible.

The Option Space

We saw that some of the DR frames contained only one option. It seems wrong to assume that a DR frame will always include several competing options (the word “option” misleadingly hints at an omnipresent context of alternatives). Not only do some problems appear to have a straightforward solution in context, but we also observed in some cases that, with every added DR, the available options tended to become ever fewer for each sub-problem, and ever more detailed, as a result of the increasing need to satisfy already made commitments.

What DSD Lent the ECOM Design Process

Having advanced our understanding of DSD through this applied case study, it was not entirely a one-sided endeavour. Having applied DSD to the EuroCODE design, we were in a position to offer a new, alternative interface to the one proposed by the ECOM design team. The interface we proposed was a direct result of the information contained in the final representation of the design space. To illustrate, the two interfaces are presented in Appendix .

The two interfaces displayed above are substantially different. The main differences in the DSD-developed interface cover the three main design issues investigated, i.e. the type of audio-visual connections it is possible to make, how to control access, and how to control bandwidth during service usage. The ECOM designers’ interface revealed more than one instance of redundancy in connection types available, which wastes screen space, manpower resources in implementation, and confuses users, for example Snapshot and Glance. The types of connections it is possible to make using the DSD interface are solely those connections which fulfill clear task requirements. These are the tasks specified in the DSD (n) frame in Appendix 2.

The ECOM designers’ interface used analogue graphic door state icons to indicate the degree of availability set by users of the system. These icons are highly ambiguous, and as such are practically devoid of any consistent meaning. Availability was expressed in the DSD interface by the user selecting a target, upon which a target field opens (cf. the DSD screen layout) which provides information on the accessibility of the target on two different types of connection called Glance and Connect. The target field contains the target’s name and the keywords ‘Glance’ and ‘Connect’. If Glance is permanently blocked with respect to this particular user, a ‘P-block’ keyword appears next to ‘Glance’. If Glance and Connect are temporarily blocked, a ‘T-block’ keyword appears next to ‘Glance’ and ‘Connect’. If Glance or Connect are not blocked, a ‘possible’ appears. This includes the case where the particular user has been made an exception to a T-block. If the target is not logged in, a ‘logout’ appears. If the target is occupied on Connect, an ‘occupied’ appears. There may also be information on why the target has temporarily blocked Glance and Connect and when the connections may become unblocked. It is up to the target to provide this information by writing in a field which opens each time the connections are temporarily set to blocked by target (cf. the DSD screen layout). The information may be, e.g., ‘in meeting until 3 pm’ or ‘has deadline at noon’. Thus, in the DSD-generated interface, the user receives a potentially large amount of explicit information about target availability, which is more reliable than the door states interface.

The way in which cost and quality of service was controlled in the ECOM designers’ interface was left largely unspecified. Using DSD suggested the following implementation. Cost and quality control information should be given through the provision of a cost per minute target figure linked to a recommended bandwidth and audio encoding rate before opening the connection. If the user expands on the bandwidth and/or audio encoding rate used during connection, s/he will know that the cost per minute increases even without being told the details at that time. Users of other connection types do not need cost information. However, users of Connect need quality control during the connection. On-line cost information for ISDN line Connects can be easily added. Sliders were chosen since, as frame rate and resolution are inversely proportional, they can be combined into one slider. Quick changes in task requirements during a Connect connection may require quick manipulations of bandwidth, frame rate/resolution and/or audio encoding rate. A set of three slides which are permanently on the screen during Connect connections make such continuous adjustments more easy to do than pull-down menus. This argument takes precedence over concerns about screen clutter. In this solution, cost information related to bandwidth was not included as has been proposed for ECOM, however this information could be easily added to the solution proposed here, so that the bandwidth slider during Connect connections using an ISDN line continuously expresses the cost per hour of the bandwidth used. An in-depth analysis and explanation of the DSD interface can be found in Bernsen and Ramsay (1994a).

OVERVIEW OF THE CERD CASE STUDY

The CAA (Civil Aviation Authority) has been engaged in a long-term project to upgrade the computer systems that are used to manage British air space. Part of this CCF (Central Control Function) project has recently been completed by Praxis, a software engineering company located in Bath. A system called CDIS (CCF Display Information System) is primarily concerned with managing the approach of aircraft into major airport complexes (MACs), and one component of CDIS is called CERD (Computer Entry Readout Display) which is used to display information about the arrival sequence of aircraft and which allows controllers to change that information subject to certain constraints. The CERD, in particular, is a device used in air traffic control for the display of information about the arrival sequence of aircraft into Major Airport Complexes, and which allows air traffic control officers (ATCOs) to modify that sequence.

It was the design of the CERD that was of concern to us. As in the previous case study, an important factor to bear in mind was that this was a remote, retrospective design process from our point of view. An additional factor worth pointing out is that a CERD prototype interface had been passed by the Civil Aviation Authority (CAA) four years ago, which satisfied certain CAA guidelines, and as such represented a set of design commitments which have only been altered minimally since being passed. From a reading of the documentation made available to us about this design process, it was clear that there was a degree of commitment to this interface. The development of the given design space was explained with reference to the structure of the space as it stood (Buckingham Shum, Duke, Hammond and Jørgensen (1994)), along with its *assumed* underpinning design rationale. This space and its development was compared and contrasted with an alternative design space, that which might have resulted, *had* DSD been applied by the design team from the initial generation of their design space. This process of representing and developing "the space according to DSD", is of a more prospective and prescriptive nature than the space as it stood. The comparison of the two end-state design spaces was an attempt to reveal where using the design space development technique makes a difference.

Our primary goal was thus to investigate where, and how using DSD would make a difference to design. The design space as it stands was represented, as it was of interest to witness the parallels (or lack of them) between CERD's design space which produced an interface which has been "given the green light", and that space which resulted from use of the DSD technique. This comparison allowed us to address the issue of whether use of DSD would have pointed to a design space containing the same, or different (sets of) operations as those of the current CERD interface. We hypothesised that using DSD would lend perspicuity to the design process, by providing an explicit description of the path taken through the space.

Figure 8. Tracking the Routes Taken Through the CERD Design Space.

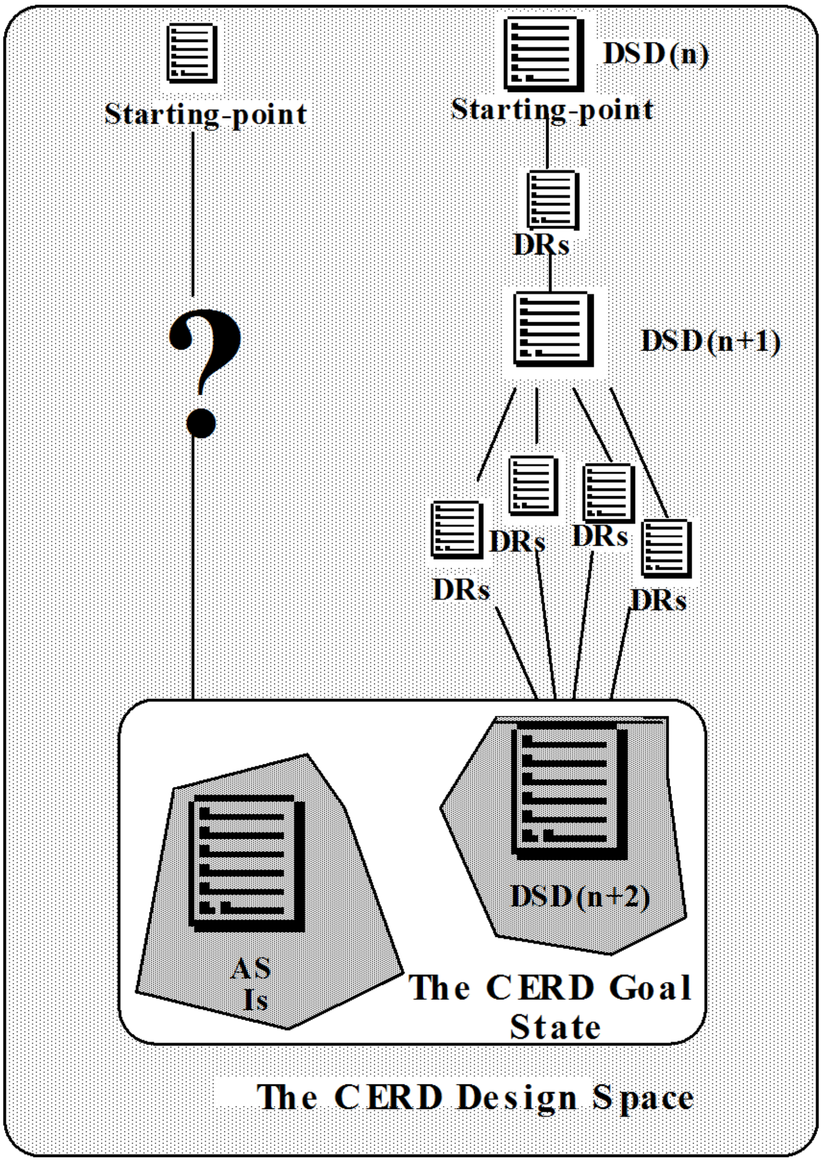


Fig. 10 displays two potential routes through the design space under discussion. One of the ‘routes’ (which is mostly unknown) is that which the authors assumed, on the basis of the evidence available, is that which had been taken by the CERD designers. It is assumed that their starting state was the system specification document. Both the authors and the CERD design team arrive at *generically* the same goal state (because the system specifications have been observed by both parties) but at *specifically* different ‘locations’ within that goal state: we have worked to the same general constraints but we have ended up in different ‘places’ within the same general ‘location’. DSD can document why we ended up where we did but we don’t have a corresponding documentation of why the Praxis design team ended up where they did. On the right-hand side of the diagram is represented the route taken by a design team (the authors) employing the DSD technique. As seen, the route chosen through the space is documented and supported by Design Rationale representations (DRs) which justify the course through the space which was taken. The development of the structure of the space was incrementally recorded in DSD frames.

Four specific design issues were addressed in the case study, namely:

Issue 1: How to assign special category indicators to flights?
Issue 2: How to reposition flights?
Issue 3: How to resequence flights?
Issue 4: How to swap flights?

A "super-level" set of more general design issues were also addressed, which were relevant to all four of the specific issues to be analysed, namely**:**

1. How should messages from NAS be displayed?
2. How should NAS messages and ATCO feedback be displayed?
3. How should the CERD signal CAP changes ?
4. How should NAS events be notified to the ATCO?
5. How should it be indicated that the CERD is about to lose some functionality?
6. How should screens be tidied?

A DSD (n) frame was derived in the same manner in which the generic frame for the EuroCODE case study was derived. This DSD frame was termed "n" as it was purely a starting point for the analysis. In 'real' design processes, DSD frames would form an ordered series numbered in cardinal numbers. The DSD (n) frame was established by gathering factual and evaluative commitments from the CERD background documents. Domain information was obtained by e-mail communication with Praxis. Implicit assumptions under which the design team were working had to be devined. This involved boot-strapping the outline DSD (n) with an emerging DR representation of the problem space. A set of generic tasks were pulled out of the background information available. These tasks, which are to be ultimately supported by the CERD, became the focus of what are termed 'design problems' in the DR representations which follow (cf. Design Issues 1-4 above). The tasks were decomposed into their constituent sub-tasks (specific design problems to be solved), listed against potential solutions (options), and judged in terms of the commitments present in DSD (n), and new commitments springing to light in the problem solving process. Thus, the DRs were boot-strapped back to DSD (n) by relocating these commitments.

The general "super-level" issues were dealt with first, and the proposed solutions to these six general design problems, including other new, relevant commitments, were inserted in DSD (n+1). DSD (n+1) provided the basis for the subsequent problem solving done upon Issues 1 to 4.

The Advancement of DSD Through this Practical Application

Apart from confirming the difficult nature of remote, retrospective design process analyses, the principal finding extracted from the case study was the usefulness of DSD in a design process that is highly constrained. With reference to Appendix 3 which displays CERD DSD (n), one gathers the extensive range of constraints at play in this design process. We were working with ten scientific and technological constraints alone, without taking into account any others from the rest of the frame.

What DSD Lent the CERD Design Process

The design space development technique was well equipped to answer a selection of issues raised by the Praxis HF designer in the course of the case study, one of which was that of how to obtain HCI requirements to support a task which has never been computer-supported.

DSD represents, amongst other things, a form of usability engineering "requirements capture" and itself also draws upon other usability engineering requirements capture methods such as interviews, questionnaires, task scenario walkthroughs, interface mockups, etc. With this is mind, it would have been easy for the output from requirements questionnaires completed by ATCOs (cf. Buckingham Shum, Duke, Hammond and Jørgensen (1994)) to have formed input to an initial DSD frame. The use of DSD frames encourages the designer to consider requirements *in terms of* the appropriate high-level constraints, collaborative and organisational aspects, system and interface aspects, task and task domain aspects, and user and user experience aspects.

A second issue was that of how to make the definition of the user interface more succinct and intelligible—how to efficiently describe displays, task flow, and functionality (i.e. using something shorter than 100 pages of user interface VDM, plus informal UI definition documents). Ideally, the content of DSD frames, when supported by display sketches, presents the design team with precise and succinct descriptions of displays and functionality, and a complete task breakdown. As such, it forms a basis and a focal point for design sessions and the subsequent decision making. Important and difficult design problems, in particular those which address difficult trade-off issues, may additionally be represented in DR frames. In safety-critical applications such as CERD, it might pay off to represent a larger-than-usual part of the design reasoning in these DR frames.

Where DSD Made a Difference: Summary of Proposed Types of Changes to CERD

There were four areas of departure from CERD, witnessed from the DSD analysis of CERD. These are laid out below with reference to the relevant DR frames which are presented in Appendices 4 to 7.

1. Symmetry of messages and screen changes when NAS gives feedback on changes proposed by the ATCO (Appendix 4, DRn:4).
2. Salient (repetitive) graphics added in the case of CERD being about to lose some functionality (Appendix 5, DRn:5).
3. Reversal of choice of flight and function (Appendix 6, DRn+1:1.1).
4. Reduction (from 3 to 1 or from 2 to 1) in number of Confirm presses (Appendix 7, DRn+1:1.2).

None of the changes proposed could be characterised as being of a revolutionary character.

1. Symmetry of messages and screen changes was generated through using the DR representation of competing options: the asymmetry of the current CERD interface was spotted and two alternative solutions were systematically generated, both of which were symmetrical. A new commitment was made which demanded symmetrical feedback by default, thus excluding the CERD option. One of the symmetrical feedback options were chosen through trading off other commitments against one another.

2**.** The question of ensuring sufficient saliency of signals to the ATCO arose from our having been impressed by the modest use of representational modalities (i.e. different forms of expressing information at the human-computer interface) of a ‘sensationalist’ or highly salient nature for representing urgent or important information in the CERD design. However, given the fact that the ATCO is not always only concentrating on monitoring CERD (cf. DSD n), in particular, during non-busy hours, we were concerned that urgent information, such as that the CERD is about to lose some functionality, is currently being brought to the ATCO’s attention only through a NAS message. We therefore proposed to add to the interface a piece of repetitive graphics which may serve to capture the ATCO’s attention more efficiently than NAS messages are. We realised that further information on the work environment of the ATCO might imply revisions to this proposal. Similarly, we did not have the time nor, perhaps, the information required, to do an in-depth study of the urgency or importance of different message categories in order to establish whether consistency principles might suggest more extended use of repetitive graphics at the interface.

3. The suggested reversal of ATCOs’ choice of flight and function was based on a hypothesis on task-related mental goal representation which might need empirical testing. We encountered a similar case of order reversal in the ECOM case study. However, in the ECOM case, reversing the order of user commands had important consequences because, on making the first command, the user might obtain feedback information which might easily lead the user to change the intended, second command into a different command. This does not seem to be the case with CERD. The proposed CERD revision, if justified, would seem to be of less import than was the case with ECOM.

4. The proposed reductions in ATCOs’ use of the Confirm button seems to be a straightforward case of removing redundancy from interactive command sequences. If we could have been sure of not having missed important information which might argue against the proposed reductions, they could yield a slightly ‘lighter’ and faster interface in terms of the user commands needed. It seemed to us that the obvious occasions upon which the ATCO was required to confirm his actions were when the command would be sent to NAS for consideration. Other, seemingly redundant, occasions were simply to effect the transfer from one screen to another.

PART THREE

LESSONS LEARNED AND GOALS IN SIGHT

A Synthesis of Lessons Learned about the DSD Technique

We now have a number of established results, based upon these two large case studies.

Facilitation of Perception of Direction

A series of DSD frames strongly facilitates perception of the overall direction of the design process, i.e., whether and to what extent the overall design goal has been completely and consistently interpreted in the design space. Thus information is yielded on the overall style of design, and on the coherence and consistency of design development with respect to the overall design goal.

Traceability

DSD is useful for keeping track of designer consensus, which is especially good for large and/or changing and/or geographically distributed teams. DSD preserves an explicit representation of the design process for use in redesign, and design of systems of a similar nature. These qualities in DSD were very well received by Sue Appelby, the Praxis Human Factors designer who was involved in the CERD case study.

Comprehensiveness and Compactness

DSD has compactness, comprehensiveness, conciseness, a wide coverage of problems, provides an explicit context for design reasoning, scenario generation and walkthroughs. When it is considered that any one DSD frame describes an important part of the overall shape of the design space around an artifact during design, contains the general design goal and a number of general constraints and criteria on the design process formed by consensus in the designer team, it can safely be claimed that use of the frame notation is helpful in keeping track of designer consensus during prototype development. Designers currently have no way of making sure that this happens. Despite describing a relatively substantial artifact development effort with a focus on usability issues, most of the important design decisions (excluding only the overall system architecture) and the generic constraints and criteria on which they are based can be represented rather compactly. Thus, there seems to be relatively little overhead on the existing design process, when it is considered that use of the notation provides structure to the activity of design and, if anything, will ultimately save designers' time, and reduce resource wastage by avoiding some of the sources of bad design.

When to use DSD

Evidence from the case studies would suggest that a DSD frame and related DR frames can best be completed after a design meeting. It is difficult to switch concentration from the DSD framework to the debate at hand. This is similar to the experience in using DSA/QOC. Design space analysis (DSA) emphasises the construction of a design space as an act of reflection by the designer, as unimpeded, fluid thinking is a prerequisite for design.

Types of Design Process

DSD frames should optimally be used from the start of a design process through to its completion including prototyping, testing and evaluation. The use of DSD goes on in parallel with all other design activities, including the use of other usability engineering methods. Essentially, the information which is presented in any DSD frame, has its origins from a variety of low-level information-gathering methods, such as interviewing users, or using questionnaires, checklists, think-aloud protocols and perhaps focus groups. Indeed, such design activities are crucial for a stable initial DSD frame. Other design activities include requirements analysis, programming, evaluating, reviewing, and prototyping. It is envisaged that the design process would begin in the usual manner, with initial discussions about what is to be designed (overall design goal), the personal goals of the users (e.g. reserving a flight from Copenhagen to Aalborg), and the constraints within which the design goal is to be reached (time-scale, finances, manpower, technical issues, user group profile, etc.). This would yield a lot of the information that is included in the first DSD frame. Task analysis would enable the further advancement through the design space, namely through the extension of DSD frame 1 into DSD frame 2, through the completion of further aspects of the DSD frame. When a consensus has been reached on the system tasks and the user tasks and their breakdown into appropriate elements, then programming could begin.

The Amount of Effort Required in Mastering DSD

An obvious question is the extent to which the concepts used in DSD parallel those concepts used "naturally" by designers. It may be assumed that system, interface, user, task, and overall design goal are everyday terms for the average designer. However, although system and interface are widely used, their definitions are less pure, and differ across designers. There are evidently empirical issues involved here which should be addressed through making DSD available to designers.

Styles of Usage

It is now clear that although DSD and DR frames structure and make explicit designer activity, there is room to exert individual freedom in how one chooses to work with DSD. Ideally, all entries would display as little subjectivity and ambiguity as possible, yet we have witnessed that this is often not the case in the absence of a proper DSD Manual. Interestingly, the Dialogue system designers proposed a partial restructuring of the DR frame, with the entries taking the order of Problems, Commitments, Justification, Options, Resolution. They preferred to have a justificatory section closer to the start of DR frames, and they would add a line containing a time estimate for developing and implementing the proposed solution, which in this case plays a large role in choosing one option against another. The problem can also be viewed as having a high or low priority in a list of problems to be considered, and this could also be worth annotating in a DR frame. As indicated earlier in this paper, we should be looking more closely into the possibility of using DSD as a general requirements capture scheme including usability requirements capture. Closer analysis showed that their proposed revision of the placement of the 'Justification' section was due to the need of justifying the *problem* to be addressed in the DR frame. This point demands our attention, as we had so far been assuming that design problems are always being transparently generated from earlier commitments.

Who should use DSD

From experience drawn from the two case studies presented, we would now recommend that one individual be responsible for completion of the frames. This might represent the easiest way to make DSD fit into existing design practise. However, it is also important to get designer consensus represented. The fact that only one designer is in charge of the DSD representation should not have an adverse effect upon designer consensus, as the frame may be filled out retrospectively, and vetted by the design team for fidelity.

How many DSD frames can be expected to be used during one design process

This clearly will depend upon the nature of the design process, and individual styles of usage, and the degree of need for design reasoning representation. If there was a great need for DR to augment the decision making process, then there would possibly exist more resultant DSD frames.

Iteration

The issue of how iteration is supported was covered in Part 1. Due to the artificiality of the design analyses we undertook, any real instances of iteration within the design processes were masked by the "reconstructive" iteration we were primarily involved in. Nonetheless, since completing these case studies, we have greatly benefited from the experiences reported by system developers working on the spoken language dialogue system mentioned in part one, who have been using DSD and DR frames to help structure their activities. Possibly the most salient observation resulting from their using DSD has been that one may need to justify the problems posed in DR frames. As we already know, DSD indeed justifies the existence of problems in DR frames through their being traceable back to the father DSD frame, yet in iterative design circumstances this is simply not an adequate methodology. In the Dialogue project, the designers arrived at a pre-testing redesign scenario, where they had uncovered areas in the system software which need to be polished. The justification for these problems comes straight from having carried out user-centred evaluations rather than from top-down specification.

Goals Within Sight: Addressing Qualitative Questions of Degree

These two large applications of DSD not only rewarded us a wealth of information about its nature, both theoretical and practical, but illuminated some further important open issues. These include the degree of usability design improvement that can be expected from the use of DSD, and where DSD would maybe give too little pay-off. It would be simplistic to assume that DSD would be amenable to evaluation in a three hour session, for example, as it is a slow, dynamical process. Having drawn upon the two case studies as test-beds for DSD, we now are in a strong enough position to move towards assaying DSD. A way forward is the development of two types of manual for DSD, one conceptual, and one practical. Both of these projects are currently underway, and will also draw upon experience from future case studies.

References

Barnard, P. and May, J. (1993): Cognitive modelling for user requirements. In Byerley, P.F., Barnard, P.J. and May, J. (Eds.): *Computers, Communication and Usability: Design issues, research and methods for integrated services.* North Holland Series in Telecommunication. Amsterdam: Elsevier.

Bellotti, V. (1994): EuroCODE AV Exemplar Round 2. *Esprit Basic Research Action AMODEUS II Internal Report* RP3-ID-IR6.

Bellotti, V. and MacLean, A. (1993): Designing for Communication and Interaction in a Ubiquitous Computing Environment. Shared Material for AMODEUS II Modelling. *Esprit Basic Research project AMODEUS II Working Paper* RP3-ID-WP10, 1993.

Bernsen, N.O. (1993a): Design of a Spoken Language Dialogue System. A Study of the Initial Specification Phase. *Esprit Basic Research Action AMODEUS II Working Paper* RP3-ID-WP3.

Bernsen, N.O. (1993b): Structuring design spaces. *INTERCHI '93 Adjunct Proceedings*. 211-12.

Bernsen, N.O. (1993c): Structuring 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.* North Holland Series in Telecommunication. Amsterdam: Elsevier.

Bernsen, N.O. (1993d): 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, N.O. (1993e): Modality Theory: Supporting multimodal interface design. To appear in *Proceedings from the ERCIM Workshop on Multimodal Human-Computer Interaction,* Nancy, November (in press).

Bernsen, N.O. (1993f): CO-SITUE Representation and Analysis of Shared Material for AMODEUS II Modelling: RAVE and Portholes. *Esprit Basic Research project AMODEUS II Working Paper* RP3-ID-WP15.

Bernsen, N.O. (1994a): Foundations of multimodal representations. A taxonomy of representational modalities. To appear in *Interacting with Computers*.

Bernsen, N.O. and Bertels, A. (1993): A methodology for mapping information from task domains to interactive modalities. *Esprit Basic Research project GRACE Deliverable* 10.1.3.

Bernsen, N.O., Lu, S. and May, M. (1994): Towards a design support tool for multimodal interface design. The taxonomy workbench and theory demonstrator. *Esprit Basic Research project AMODEUS II Working Paper* RP5-TM-WP5.

Bernsen, N.O. and Ramsay , J. (1994a): AMODEUS II Shared Modelling Exercise: DSD Modelling Technique Report on the EuroCODE Exemplar. *Esprit Basic Research Action AMODEUS II Internal Report* RP3-ID-IR8.

Bernsen, N.O. and Ramsay , J. (1994b): DSD Manual. *Esprit Basic Research project AMODEUS II Working Paper* (in preparation).

Buckingham Shum, S., Duke, D., Hammond, N. and Jørgensen, A. (1994): CERD: Functionality, Design Process, and Modelling Issues. *Esprit Basic Research project AMODEUS II Internal Report* RP4-TA-IR3.

Card, S., Moran, T. and Newell, A. (1983): *The Psychology of Human-Computer Interaction.* Amsterdam: North-Holland Elsevier.

Carroll, J.M. and Rosson, M.B. (1992): Getting around the task-artifact cycle: How to make claims and design by scenario. *ACM Transactions on Information Systems,* Vol. 10, No. 2, April, 181-212.

Dourish, P. and Bly, S. (1992): Portholes: Supporting Awareness in Distributed Work Groups. In *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI '92,* Monterey, California, 541-547.

Kieras, D. and Polson, P. G. (1985): An Approach to the Formal Analysis of User Complexity. *International Journal of Man-Machine Studies,* 30, 47-62.

Lewis, C., Polson, P., Rieman, J. and Wharton, C. (1990): Testing a walkthrough methodology for theory-based design of walk-up-and-use interfaces. *Proceedings of the CHI '90 Conference on Human Factors in Computing Systems*, 235-42. New York: ACM.

MacLean, A., Young, R., Bellotti, V. and Moran, T.P. (1991a): Questions, Options, and Criteria: Elements of Design Space Analysis. *Human-Computer Interaction* Vol. 6, 201-50.

MacLean, A., Young, R., Bellotti, V. and Moran, T.P. (1991b): Design Space Analysis: Bridging from Theory to Practice via Design Rationale. In *Proceedings of Esprit '91,* Brussels, November, Office for Official Publications of the European Communities, Luxembourg, 720-730.

MacLean, A., Bellotti, V. and Shum, S. (1993): Developing 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, 197-220.

Nielsen, J. (1993): *Usability Engineering*. New York: Academic Press.

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

Ramsay, J. and Bernsen, N.O. (1994): AMODEUS II Shared Modelling Exercise: DSD Modelling Technique Report on the CERD Exemplar. *Esprit Basic Research Action AMODEUS II Internal Report* RP3-ID-IR9.

Ramsay, J. E. and Oatley, K. (1992): Designing minimal computer manuals from scratch. In Sharples, M. (Ed.): *Computers and Writing: Issues and Implementations,* 85-98, Kluwer.

Verjans, S. (1994): EuroCODE: A Case Study in Information Mapping. *Esprit Basic Research Action AMODEUS II Working Paper* RP5-TM-WP9.

Verjans, S. and Bernsen, N. O. (1994): PaTerm: A Case Study in Information Mapping. *Esprit Basic Research Action AMODEUS II Working Paper* RP5-TM-WP6.

Appendices

Appendix 1. Scenario Used in the ECOM Case Study: A day in the life of a supervisor.

Planning

It's Monday morning. The supervisor drives to Halsskov and walks past reception to his office. He checks the Site Browser video monitor to see the status of the latest load of girders in Sines, Portugal. Once it ships, he knows he only has three weeks before they arrive. He reads his email and sees the note from his area manager, asking him to inspect the reinforcements. He prepares for the inspection trip at the pylon by reviewing his copy of a blueprint. He sends a quick email message stating that he'll be at the first pylon today, gathers his camera, phone, blueprint, etc. and then goes to catch the boat.

Inspection

He takes the boat out to the first pylon; the sea is a bit rough but he's able to land. He inspects the reinforcements and decides that they do look a bit peculiar. He shoots video and annotates his paper drawing by marking the areas that don't conform. He also indicates where he has taken video shots on his dictaphone. He is not sure whether or not this non-conformance requires immediate action, so he decides to contact the one of the designers to make sure. At the shelter on the pylon, he places his blueprint on a special desk and compares it with the electronic copy on his workstation. He thinks the electronic version may be out-of-date. He makes some glance connections to check to see which of the three designers listed on the blueprint are in. Preben turns and waves, so he sets up a videophone (vphone) connection with him to establish a voice and video connection. They also set up a global window (using a special CSCW environment) which allows them to share written comments and a shared drawing tool. After a half-hour discussion, they reach a solution and end the conference. He then goes out to explain to the workers how to proceed.

Reporting

Although in principle the supervisor has access to every electronic document and a variety of multimedia information from the pylon, he usually prefers to go back to his real office on shore to write reports.

In his office, he logs on to his workstation. He sees that he has a message from Lisette, who wants to discuss an urgent problem. She's left a description of the problem in an electronic note linked to his blueprint file. Before he deals with it, he decides to capture his own video tape so he can make his own report. He connects his camcorder to his workstation and selects the "capture videotape" application from a special menu. He finds the right section of the videotape, presses "digitise" and then "play".

While the video is being processed, the supervisor opens the on-line blueprint file and sees three icons displayed in the blueprint window, one of which is flashing. This tells him that three people have left electronic annotations on the plan already. He's already seen two and one is new. It's from Lisette. He clicks on the flashing icon and a close-up of a hand-written sketch is displayed next to the on screen blueprint. He also sees a written note describing the proposed changes. This is the problem he received the message about. He glances at her office, but she's out. He leaves an electronic message saying he'll be available in his office for the next few hours.

Because the non-conformance the supervisor identified on the bridge appears to be quite serious, he decides to inform his area manager. He rapidly composes a multimedia message consisting of parts of the recorded digitised video material he has captured, video still images of his blueprint with pencil annotations and some electronic notes he has created and sends it to the supervisor. Lisette then glances at him through the media space and they set up a vphone connection to discuss a problem with a shipment of girders from Aalborg. When they've decided on a strategy, she hangs up. It's been a long day. He takes a final look at the Sines site and sees that the girders are complete. He then heads for home.

Appendix 2. ECOM DSD (n).

<table>
<tr><td colspan="3">Design Project: EuroCODE High Road Demonstrator ECOM-NEW</td></tr>
<tr><td>DSD No. n</td><td>Date: 28.3.94</td><td>Sign: NOB, JR</td></tr>
<tr><td colspan="3">A. General constraints and criteria</td></tr>
<tr><td colspan="3">Overall design goal(s)</td></tr>
<tr><td colspan="3">Example CSCW system.
Integrate the use of text, audio and video in a high bandwidth multimedia digital network.
Support both planned, formal and unplanned, informal collaboration and interaction between geographically separate colleagues on the Danish Great Belt construction scheme.</td></tr>
<tr><td colspan="3">General feasibility constraints</td></tr>
<tr><td colspan="3">EuroCODE project costs: 12.8 million ECUs.
Timing: the project deadline is September 1995.
Implementation costs should be as low as possible.</td></tr>
<tr><td colspan="3">Scientific and technological feasibility constraints</td></tr>
<tr><td colspan="3">Only one-way and two-way connections are possible.
Use of compression and decompression techniques to keep video signal bandwidth down on all line types.</td></tr>
<tr><td colspan="3">Design process type</td></tr>
<tr><td colspan="3">A demonstrator application of a generic exploratory design process into innovative technology which already resulted in the RAVE, Portholes and Cave experimental systems. The EuroCODE designers are working in a context within which certain kinds of innovative technology have been applied to design domains to see what kinds of benefits they provide. Thus, the technology will predate, drive and constrain the design project.</td></tr>
<tr><td colspan="3">Designer preferences</td></tr>
<tr><td colspan="3">MacIntosh-based system.
ATM communication protocol will be used for sending network data.</td></tr>
<tr><td colspan="3">Realism criteria</td></tr>
<tr><td colspan="3">The artifact should be preferable to current technological alternatives, and has to be implemented on-site within the next 18 months.
Time constraints already meant that remote ISDN lines have been cut out. Their functionality will nevertheless be looked into to a limited extent (the cost problem) in this representation.</td></tr>
</table>

(DSD No. n.) Functionality criteria

The artifact should fully exploit the functional opportunities of high-bandwith communication links.

The artifact should support the main user tasks intended, namely:
- to make and break a variety of audio-visual connections;
- to specify levels of availability;
- to control signal quality.

The making and breaking of connections of any type should:
- support negotiation;
- support awareness;
- give sufficient feedback on opened connections;
- support focused collaborations;
- minimise intrusiveness.

The control of connection quality should:
- exploit a full set of useful opportunities offered by high bandwidth;
- offer minimum cost feedback on signal transmission;
- communications should be of an acceptable minimum quality;
- include financial control for ISDN lines, preferably before opening a connection;
- provide financial feedback on ISDN lines;
- be continuous;
- cover image quality;
- cover synchronisation.

The timing of bandwidth control should:
- avoid connection disruption.

Usability criteria

The set of connections should be non-redundant.

Each connection should satisfy clear user needs.

Use current formal and informal, planned and unplanned interactive work practices as benchmarks of artifact usability.

Avoid screen clutter.

The making and breaking of connections of any type should:
- be as simple as possible in steps and procedure;
- correspond with user expectations;
- give appropriate and appropriately timed feedback.

The order in which any of these connections is made should be:
- conducive to a low error-rate;
- flexible;
- intuitive.

The means of specifying availability should:
- maximise flexibility of social interactions;
- support flexible relationships with individuals;
- support flexible relationships with groups;
- be transparent;
- be meaningful;
- provide appropriate feedback;
- be as simple as possible in steps and procedure.

The control of bandwidth should:
- avoid screen clutter.

The interface should:
- display a systematic grouping of functions according to functionality.

(DSD No. n.) B. Constraints and criteria applied to the artifact within the design space
Collaborative aspects
Support remote awareness of and collaborative interaction between geographically separate colleagues. Sufficient privacy protection: Control over what information to project and to whom. Automate procedures for communication and information retrieval.
Organisational aspects
Protection of privacy via control over accessibility: when needed, users control who can connect to them and what kind of connections can be made. The HRD should connect about eleven sites, namely Copenhagen main office, the Halsskov site, the supervision office in Halsskov, the design office, Kalundborg, Aalborg,Sines, Livorno, Sprogø, pylon A and pylon B. Not all nodes should be similarly equipped (cf HRD plan, p.11, document ID-IR6). Line types: Local connections between Halsskov, Copenhagen, Sprogø and Pylons A and B are dedicated 2 Mbit lines. 150 Mbit dedicated link between supervision and admin offices in Halsskov. Remote connections (i.e. between Halsskov and all other nodes: Aalborg, Kalundborg, Sines, Livorno) are commercial ISDN lines. Only sites on high bandwidth lines will have media spaces. This DSD assumes that sites on ISDN lines will have (quality reduced) media spaces.
System aspects
Combine different types of communication channels involving advanced I/O communication equipment. Property list database defining a set of parameters for a service: who is allowed to make a given kind of service connection; what kind of reject message is broadcast; what kind of notification the user gets on a given type of connection. Enable the described user tasks. Provide feedback on when and what information is being captured and to whom the information is being made available. Notify users of device conflicts, connection failure and inappropriate connection attempts. Convey to the user current connection status and possible connections. Ensure that the user selects the correct connection. Ensure that the user gets the connection expected. Handle connection overlays. Net work is unstable.
Interface aspects
I/O equipment: Loudspeakers, microphones, tv-monitors, cameras, camcorder, digital desk and pen, workstation; The interface should present the user with a means of making various audio-visual connections, specifying availability and controlling connection quality and cost.

(DSD No. n.) Task aspects

User tasks:

1.To establish a two-way connection for the purpose of conversation and collaboration.
2. To get a quick view of what's going on in someone else's office, in order to see whether they are in and whether they are available or not.
3. To maintain ongoing awareness of the progress of work and who's around.
4. To communicate asynchronously.
5. To notify another person of the wish to communicate with them.
6. To be aware of others' access to you and their attempts to access you.
7. Users should be able to moderate the degree to which they are accessible on different connections and/or at certain times and/or for certain people, and to advertise this such that others know whether it makes sense to try to connect.
8. To ensure privacy regarding one's working activities.
9. Users should be able to dynamically decide whether to grant access on some connections.
10. To select the type of AV connection desired.
11. To set special permissions. Special permissions grant access to certain individuals only.
12. To dynamically control bandwidth, frame rate and resolution of video image so that the recipient can make sense of it.
13. To control the cost of ISDN calls which depends on time and bandwidth used, control preferably to be exerted before opening a connection.
14. Use no more bandwidth than necessary.
15. To continue the call with minimum disruption even if bandwidth requirements change.
16. To dynamically reallocate limited bandwidth resources (e.g., prioritise frame rate for conversation and then prioritise resolution for inspecting a view of the bridge).
17. To monitor the clogging of the network if there is heavy traffic competing for limited bandwidth.

User and user Experience aspects

Administrators, bridge construction supervisors, consultants (e.g. experts on how concrete sets in deep water), designers (the people who produced the original plans), contractors (people around Europe involved in supplying parts for the bridge).
The user group is relatively large and spans occasional users and expert, daily users, both working as individuals and as groups.

C. Hypothetical issues

D. Documentation

E-mail communications with Rank Xerox EuroPARC;
Transcript of interview with Daniele Pagani Friday 26th November 1993: “Discussion of Media Space Interface and Related issues”.
Ftp documents in project/rxep/RP3-ID-IR6-ec-av-exemplar/exemplar-dsa.ps, which includes the following screen dumps:
Figure 1. EuroPARC's Raven connections control interface exemplar-fig1.eps
Figure 2. EuroPARC's Xgprops access control interface ftp.mrc-apu -fig2.eps
Figure 3. EuroPARC's Portholes awareness server windowexemplar-fig3.eps
Figure 4. Toronto's Cave interfaceexemplar-fig4.eps
Figure 5. EuroCODE's ECOM interface for making a vphone connection and setting availability exceptionsexemplar-fig5.eps
Figure 6. EuroCODE's ECOM interface for viewing a bridge building siteexemplar-fig6.eps
Figure 7. nv digital, video-only interface (developed at PARC)exemplar-fig7.eps
Figure 8. ivs digital audio-video conferencing interface (developed at INRIA)exemplar-fig8.eps
Appendix II. QOC Design Space: A graphical representation of the QOC for the issues described in the document

E. Key: DSD No. (n) indicates the number of the current DSD frame.
‘Null’ means that the artifact does not embody a certain aspect of DSD.
Italics indicate new elements in DSD (n) as compared to DSD (n-1).

Appendix 3. CERD DSD (n).

Design Project: ESPRIT: Computer Entry Readout Device (CERD)		
DSD No. n	**Date: 17.5.94**	**Sign: JR**

A. General constraints and criteria
Overall design goal(s)
A device which will be used in air traffic control for the display of information about the arrival sequence of aircraft into Major Airport Complexes, and which allows air traffic control officers (ATCOs) to modify that sequence.
General feasibility constraints
Information is unavailable on financial resources and time constraints.
Scientific and technological feasibility constraints
There are some known **hardware constraints**: 1. The size of the display will be limited by the hardware panel. Larger margins than necessary will be used , a 'leftover' from an earlier (now defunct) requirement that the CERD support two other tasks (originally, touching the margins would have switched between sets of buttons). 2. The CERD will sit at a 15 degree angle, which creates a parallax effect due to the difference in depth of the electro-luminescent display and infra-red overlay which detects where it is being touched. 3. All active areas must be rectangular. 4. Only three sizes of text are possible. 5. The colour can only be black on orange, or orange on black. 6. A touch screen is used, and screen areas can only be orange, black, or hatched (but hatched background makes text unreadable). 7. The plasma panel restricts the number of active areas in a column to seven. **Design constraints:** 8. A CERD shall only be associated with a single MAC at any one time. 9. A flight can only be on one CERD at any one time. 10. It shall only be possible to manipulate the SAS for a given MAC from a single position at any one time.
Design process type
Retrospective, (potential) redesign. Design for a task which has never been computer-supported.
Designer preferences
Data and functionality required by the system was formally specified in the VDM notation.
Realism criteria
The artifact should be preferable to current alternatives. This is the first time such a device has been needed, because of the introduction of new patterns for stable approach sequencing.

(DSD No. n.)
Functionality criteria

The artifact should support the main user task, namely: managing the approach of aircraft into major airport complexes. This involves the supporting of the following top-level sub-tasks:
1- to assign a two-letter code to a flight (Special Category Indicator);
2- to reposition flights in the SAS;
3- to resequence the order of flights within the SAS;
4- to swap the places of flights in the SAS.
5-to tidy the screen.

All of the sub-tasks should do the following:
1- allow displaying of messages from NAS;
2- allow ordering of messages by class and then by arrival within class;
3- signal CAP changes;
4- notify events to NAS;
5- indicate that the system is about to lose some functionality.

All of the sub-tasks should fulfill the following Common Control Functional Requirements:
6- An ATCO can conduct no further operations whilst NAS considers a request;
7- all flights in the SAS shall be displayable on the CERD;
8- SAS entries used to describe "gaps" in the landing sequence shall not be displayable on the CERD;
9- flights displayed on the CERD shall be ordered in landing sequence order unless otherwise specified;
10- at each CERD, one transaction shall be completed prior to the commencement of the next;
11- all approach sequencing functions on the CERD shall be selectable directly or indirectly from a top-level ASA display (the rest menu);
12- the CERD shall return to the ASA rest menu from any lower levels following an adaptable period of controller input inactivity on the CERD;
13- the ASA rest menu shall be capable of displaying simultaneously at least 12 flights in addition to any rest menu selection options and status data;
14- the ASA rest menu shall be updated such that the position of flights on the CERD is fixed, i.e. a controller needs to be sure that the flight about to be selected on the CERD does not move, thus causing an incorrect selection;
15- the controller shall be able to request that any flight in the SAS may be selected and repositioned anywhere else in the SAS, including the first and last positions, with the limitation that no more than ten flights are affected;
16- the approach sequencing function shall be available at all CERDs allocated to the ASA role at all times, unless the function is manually inhibited;
17- approach sequence data shall be displayed on the CERD in such a manner that no two flights in the SAS show the same ASNO simultaneously;
18- any response shall not be greater that 0.1 seconds;
19- the modificaton of data displayed at the entering positions shall not take longer than 0.2 seconds.

1 Assigning a Special Category Indicator:
1- allow flight selection;
2- confirm code assignment;
3- allow assigning to be cancelled;
4-backtrack over command sequences;
5-specify category of event.

2 Repositioning flights in the NAS should:
1- allow flight selection;
2- confirm repositioning;
3- allow repositioning to be cancelled.

3 Resequencing the order of flights within the SAS should:
1- allow flight selection;
2- confirm resequencing;
3- allow resequencing to be cancelled.

4 Swapping the places of flights in the SAS should:
1- allow flight selection;
2- confirm swapping;
3- allow swapping to be cancelled.

Usability criteria

General criteria:
- -messages should be displayed in the order in which they arrive;
- - messages should be in English
- -ensure sufficient saliency of urgent or important messages.
- -messages should be clearly announced, but not interrupt or obscure other information
- -warnings should take priority over special messages;
- - special messages should take priority over data changes;
- - messages relating to a flight are removed automatically if that flight leaves the SAS;
- - feedback should be appropriate;
- - feedback should be timely;
- - there should be as little redundancy as possible;
- - commands should be as simple as possible to execute;
- - the screen's functionality should be logically grouped;
- - save screen space;
- - the text should be legible;
- - interaction sequences should correspond with user expectations.

(DSD No. n.) B. Constraints and criteria applied to the artifact within the design space

Collaborative aspects

During busy periods, one ATCO is dedicated to using CERD. Other ATCOs handle aircraft as they approach the airport, handing over to the CERD operator for final sequencing through the CAP to land. During quieter periods, the CERD user may cover other ATCO roles as well, and so will not be focusing on the CERD display and tasks to such a degree. ATCOs make use of radar and direct communication with the pilots to control the stacks.

Further details about how responsibility is passed between ATCOs, the different roles played by ATCOs within air traffic control, and other domain-related questions are not currently available.

Organisational aspects

The ATCO uses CERD within CDIS, a larger information system monitoring the arrival and departure of flights into and out of MACs. This information is available to ATCOs. CDIS is part of NAS. Any ATCO's requests to CDIS are passed to NAS for approval. While awaiting a response from NAS, the ATCO cannot initiate any further commands.

System aspects

The system should enable the described user tasks. It should provide feedback. The work station will include monitors, radar display, and the CERD, which has a touch-sensitive plasma display that presents information to the users as a set of logical screens.

Interface aspects

The interface should present the user with a means of assign ing a two-letter code to a flight, of repositioning flights in the SAS, of resequence the order of flights within the SAS, of swapping the places of flights in the SAS.

The functions Assign, Reposition and Resequence should be implemented in dedicated screens accessed from the ASA rest menu.

Task aspects

User tasks:

- to assign a two-letter code to a flight;
- to reposition flights in the SAS;
- to resequence the order of flights within the SAS;
- to swap the places of flights in the SAS;
- an ATCO may be responsible for more than one SAS into a MAC;
- the intense and stressful environment prohibits auditory alarms.

User aspects

English speaking air traffic control officers (ATCOs).

ATCOs have four year's general training prior to obtaining their licence.

C. Hypothetical issues

D. Documentation

E-mail communications with HF designer at PRAXIS;

Ftp documents in mrc-apu/pub/amodeus/project/yorkpsy:

"CERD: Computer Entry Readout Device"

D.J. Duke, S.J. Buckingham Shum & A. Jorgensen

Amodeus Project Document: ID / IR3;

"CERDxtra.rtf" screen scans of four pages from the CDIS ATC User Manual by the PRAXIS HF designer;

"Central Control Function: System Requirements Specification, Volume 6 , Approach Sequencing" PRAXIS document, 6th November 1990.

E. Key:

C = Collaborative aspects.
O = Organisational aspects.
S = System aspects.
I = Interface (or more generally: system Image) aspects.
T = Task aspects including task domain aspects.
U = User aspects.
E = User experience aspects.
DSD No. (n) indicates the number of the current DSD frame.
‘Null’ means that the artifact does not embody a certain aspect of DSD.
Italics indicate new elements in DSD (n) as compared to DSD (n-1).

Key to Abbreviations

ASA	Approach Sequence Allocator
ASNO	Approach Sequence Number
ATCO	Air Traffic Control Officer
CAA	Civil Aviation Authority
CAP	Common Approach Point
CERD	Computer Entry Readout Device
CCF	Central Control Function
CDIS	CCF Display Information System
CTA	Calculated Time of Arrival
MAC	Major Airport Complex
NAS	National Airspace System
SAS	Stable Approach Sequence

Appendix 4. CERD Design Rationale (DR) Frame No. n.4.

How should NAS events be notified to the ATCO?

Design Project: ESPRIT: Computer Entry Readout Device (CERD)		
Prepares DSD No. n+1	**DR No. n.4**	**16.5.94-JR/NOB**
Design problem		
How should NAS events be notified to the ATCO?		
Options		
1	During Assign, Reposition, and Resequence, the affected flight(s) is shown with a "*" in place of its ASNO. If NAS subsequently rejects the assign request, a message is displayed on the message line and the ASNO of the flight is restored. If the request is accepted, a message is displayed on the message line and the two letter code replaces the "*" on the flight data display.	
2	During Assign, Reposition, and Resequence, the affected flight(s) is shown with a "*" in place of its ASNO. If NAS subsequently rejects the assign request, a message is displayed on the message line and the ASNO of the flight is restored. If the request is accepted then the two letter code replaces the "*" on the flight data display (CERD).	
3	Nothing is announced on the message line, otherwise as in Options 1 and 2.	
Relevant commitments involved		
1	Feedback should be appropriate.	
2	Feedback should be symmetrical.	
3	An ATCO can conduct no further operations whilst NAS considers a request.	
4	There should be as little redundancy as possible.	
Resolution		
Option 1.		
Justification		
Options 1 and 3 have symmetrical feedback. Option 2 has asymmetrical feedback. Asymmetrical feedback needs a special justification, but it is not clear what that might be. Option 1, although somewhat redundant, as the same information is announced more than once, (commitment 4), gives symmetrical feedback (commitment 2). Option 3 does not provide appropriate feedback as many things may change at once on the flight display both in case of confirmation and rejection from NAS.		
Questions		
Links to other DRs		
Documentation		
See DSDn.		
Insert in next DSD frame		
Both acception and rejection of requests to NAS are announced on the message line with the appropritate changes implemented on-screen. New commitment: Feedback should be symmetrical.		

Appendix 5. CERD Design Rationale (DR) Frame No. n.5.

How should it be indicated that the CERD is about to lose some functionality?

Design Project: ESPRIT: Computer Entry Readout Device (CERD)		
Prepares DSD No. n+1	**DR No. n.5**	**16.5.94-JR/NOB**
Design problem		
How should it be indicated that the CERD is about to lose some functionality?		
Options		
1	A warning message explaining the type and seriousness of the functionality about to be lost appears on the message line from NAS (CERD).	
2	A warning message as in Option 1 appears, accompanied by repetitive graphics of some form, clearly visible on the screen from a distance.	
Relevant commitments involved		
1	The intense and stressful environment prohibits auditory alarms.	
2	Ensure sufficient saliency of urgent or important messages.	
Resolution		
Option 2.		
Justification		
Given the assumption that the message window is being constantly monitored, Option 1 might be sufficient. Since that is not always the case (during less busy periods), a warning message alone may be insufficient, thus presenting the case for Option 2. Auditory alarm might be used instead of repetitive graphics, but this Option is ruled out by commitment 1. In fact, in the CERD environment repetitive graphics represents the weakest modality to be added for extra saliency.		
Questions		
How are two or more simultaneous warning messages represented?		
Links to other DRs		
Documentation		
See DSDn.		
Insert in next DSD frame		
To indicate that the CERD is about to lose some functionality: A warning message explaining the type and seriousness of the functionality about to be lost appears on the message line from NAS accompanied by repetitive graphics of some form, clearly visible on the screen from a distance.		

Appendix 6. CERD Design Rationale (DR) Frame No.n+1:1.1.

How should flight selection for Assigning be allowed?

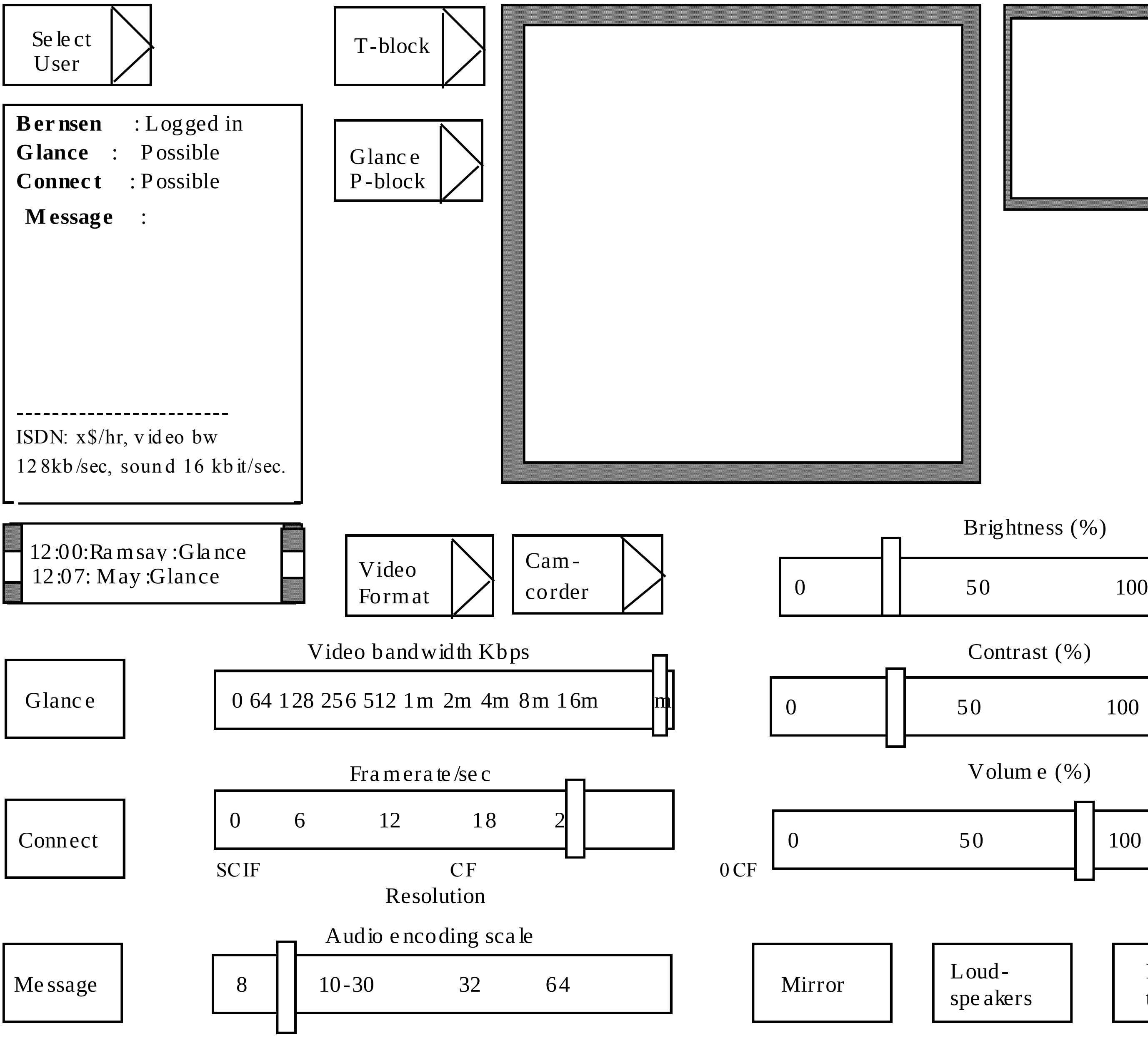
<table>
<tr><td colspan="3">Design Project: ESPRIT: Computer Entry Readout Device (CERD)</td></tr>
<tr><td>Prepares DSD No. n+2</td><td>DR No. n+1:1.1</td><td>Date: 16.5.94-JR/NOB</td></tr>
<tr><td colspan="3">Design problem</td></tr>
<tr><td colspan="3">How to should flight selection for Assigning be allowed?</td></tr>
<tr><td colspan="3">Options</td></tr>
<tr><td>1</td><td colspan="2">Flight is selected and then the operation (Flight, Assign: a dedicated screen opens for selection of special category).</td></tr>
<tr><td>2</td><td colspan="2">Operation is selected and then the flight (Assign, Flight, Confirm: a dedicated screen opens for selection of special category) (CERD).</td></tr>
<tr><td colspan="3">Relevant commitments involved</td></tr>
<tr><td>1</td><td colspan="2">Interaction sequences should correspond with user expectations.</td></tr>
<tr><td>2</td><td colspan="2">Commands should be as simple as possible to execute.</td></tr>
<tr><td>3</td><td colspan="2">The interactive flight selection command sequence should be identical for Assign, Reposition, Resequence and Swap.</td></tr>
<tr><td>4</td><td colspan="2">Feedback should be sufficient.</td></tr>
<tr><td colspan="3">Resolution</td></tr>
<tr><td colspan="3">Option 1.</td></tr>
<tr><td colspan="3">Justification</td></tr>
<tr><td colspan="3">It is hypothesised that the ATCO’s cognitive focus is primarily on the flight to be assigned a aspecial category and secondarily on the special category to assign to that flight. If true, Option 1 corresponds with user expectations whereas Option 2 does not.
On pressing Assign, the dedicated screen opens. This provides feedback on the ATCO’s choice without the need for pressing Confirm and thus simplifies command execution by comparison with Option 2.
The proposed solution assumes that the CERD use of Confirm does not serve any other purpose than opening the dedicated screen. If, e.g., pressing Assign elicits a change in the flight identification on the main screen, which should be verified by the ATCO before proceeding to the dedicated screen, Option 1 should include the pressing of Confirm as in Option 2.</td></tr>
<tr><td colspan="3">Questions</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td colspan="3">Links to other DRs</td></tr>
<tr><td colspan="3">DR No. n+1:1.2</td></tr>
<tr><td colspan="3">Documentation</td></tr>
<tr><td colspan="3">See DSDn.</td></tr>
<tr><td colspan="3">Insert in next DSD frame</td></tr>
<tr><td colspan="3">Allowing flight selection for Assigning: Flight is selected and then the operation (Flight, Assign: a dedicated screen opens for selection of special category).
New commitments: Feedback should be sufficient;
The interactive flight selection command sequence should be identical for Assign, Reposition, Resequence and Swap.</td></tr>
</table>

Appendix 7. CERD Design Rationale (DR) Frame No. n+1:1.2.

How should the assigning of a two digit code to a flight be confirmed?

Design Project: ESPRIT: Computer Entry Readout Device (CERD)		
Prepares DSD No. n+2	**DR No. n+1:1.2**	**Date: 16.5.94-JR/NOB**
Design problem		
How should the assigning of a two digit code to a flight be confirmed?		
Options		
1	After selection of special category on the dedicated screen (cf. CERD and DRn+1.1.1), Confirm is pressed. This initial confirm is answered by a “Warning: press Confirm to proceed” message”. Once the command has been confirmed a second time it is passed as a request to NAS to be effected. The flight’s ASNO is replaced by an asterisk whilst awaiting request permission or refusal from NAS (CERD).	
2	Selection of special category on the dedicated screen elicits the feedback: “Are you sure you want to assign this flight as [special category]? Once the command has been confirmed it is passed as a request to NAS to be effected. The flight’s ASNO is replaced by an asterisk whilst awaiting request permission or refusal from NAS.	
Relevant commitments involved		
1	Commands should be as simple as possible to execute.	
2	Feedback should be sufficient.	
Resolution		
Option 2.		
DR No. n.: Justification		
Option 2 provides sufficient feedback and is done with one key press less than Option 1.		
Questions		
Links to other DRs		
Documentation		
See DSDn.		
Selection of special category on the dedicated Assign screen elicits the feedback: “Are you sure you want to assign this flight as [special category]? Once the command has been confirmed it is passed as a request to NAS to be effected. The flight’s ASNO is replaced by an asterisk whilst awaiting request permission or refusal from NAS.		

Appendix 8: The DSD Interface



Bernsen : Logged in
Glance : Possible
Connect : Possible
Message :

ISDN: x$/hr, video bw 128kb/sec, sound 16 kbit/sec.

12:00:Ramsay :Glance
12:07: May :Glance

Appendix 9: The Original ECOM Interface

< please attach file Appendix9.eps >