A METHODOLOGY FOR MAPPING INFORMATION FROM TASK DOMAINS TO INTERACTIVE MODALITIES

Niels Ole Bernsen, CCI, Roskilde University

Annick Bertels, ESADG, Katholiek Universiteit Leuven

Summary: Today's human-computer interface designers are facing the opportunity of being able to combine an increasing number of input/output modalities for a given application. This opportunity is accompanied by the problem of how to select the right combination of modalities for the application in a way which is principled rather than *ad hoc*. This exploratory paper proposes a methodology for this purpose illustrated by two case studies. The methodology is preliminary as it makes use of a number of HCI design support tools some of which are in the process of development in parallel with the methodology itself, including tools based on concepts and taxonomies of interactive modalities.

Keywords: Multimodal systems, interface design, usability engineering, HCI design methodology, modality theory.

1. Introduction

Literally thousands of different combinations of input and/or output information representation modalities are currently becoming available to human-computer interaction designers, from unimodal spoken language input to complete multimodal virtual reality interactive systems. Each single modality or multimodal combination has its own specific capabilities of representing or conveying information and it is obviously important to be able to select the right combination of modalities for a given application. The question is how this might be done in a principled manner so as to optimise the usability of the interface, given the specific purpose of the artifact to be designed. Answering this question from the point of view of usability engineering involves addressing issues such as those listed in the following research agenda which forms part of the ESPRIT Basic Research project GRACE (Bernsen 1993a). The agenda items are listed in order of increasing complexity:

(1) to establish sound foundations, both conceptually and in terms of an operational taxonomy, for describing and analysing any particular type of unimodal or multimodal output representation relevant to human-computer interaction (HCI);

(2) to create a conceptual framework for describing interactive computer interfaces so as to cover both input and output of information;

(3) to apply the results of steps (1) and (2) above to the analysis of the problems of information-mapping and information-transformation between work/task domains and human-computer interfaces in information systems design.

This paper is of an exploratory nature. It proposes a methodology for addressing agenda item 3 above based on emerging results from GRACE work on agenda item 1, and discusses some of the research needed to complete the methodology. Briefly, the main problem raised by agenda item 1 is how to build a principled foundation for addressing the information representing capabilities of thousands of different, potentially useful combinations of modalities. The only viable approach seems to be through the definition and analysis of a limited set of elementary modalities from which any particular modality combination can be built and analysed. First results from adopting this approach have been reported in (Bernsen 1993a,b,c). We are only now beginning to address agenda item 2 and have no results available for use in this paper.

2. A Methodology for Mapping Information from Task Domains to Interactive Modalities

The methodology proceeds in five steps:

Step 1
The first problem is to identify the information to be exchanged by user and system during task performance in the application domain. So the aim of Step 1 is to obtain the information from the task domain which is needed to select a reasonable and possibly optimal mapping from task domain information to interface input/output representation. The nature and variety of the information relevant to this end should not be underestimated. Often, but not always (see Sect. 4 below), a central part of the information needed for solving an information-mapping problem is information on users' tasks. However, any reasonably versatile or powerful IT artifact can be used for performing a multitude of different tasks and it is obviously not possible during practical systems design to consider each and every such task as to its information-mapping requirements. In other words, it is necessary during practical interface design to be selective as to the tasks to be considered in any detail. The ideal way to be selective is to identify a limited set of representative tasks or scenarios to be performed by using the intended artifact and carry out the information-mapping analysis on these. The problem is that no guaranteed method for generating an appropriate set of scenarios currently exists in HCI/usability engineering. One heuristics which has been proposed (Carroll et al. 1992) is to weak for our purposes and we are currently testing an alternative method (Klausen and Bernsen 1993). Let us simply assume at this stage that the best current methods or heuristics are to be applied for the purpose of identifying representative tasks.

The results of Step 1 would normally be (a) high-level information required to solve the information-mapping problem and (b) a small set of representative tasks which users should be able to carry out on or with the intended artifact. These results constitute an operationalisation of the information-mapping problem. Step 1 is crucial to the success of the methodology. To the extent that the selected tasks lack representativity, the risk is that important information requirements on the artifact have been overlooked.

Step 2
In Step 2 the representative tasks are individually analysed in as much detail as possible in order to identify their goals and initial states, the activities and procedures involved, how they might go wrong, the task (work) environment, the intended users and their

experience, etc. The analysis should primarily aim at revealing the input/output information representation needs of the tasks. That is, while a more or less complete task analysis may be performed either formally or informally, not all of the information it produces needs to be explicitly represented in order that the information-mapping methodology will be successful.

Step 2 is closely related to Step 3:

Step 3

In Step 3 the relevant information acquired through Steps 1 and 2 is represented explicitly and succinctly, for instance using the Design Space Development (DSD) notation for representing design space structure (see below). In principle, this representation should contain everything which is relevant to the input/output modality choices to be subsequently made. The conceptual apparatus and terminology used in the representation should be close to, or ideally the same as, the one described in Step 4 below.

Step 3 serves the purpose of making explicit the requirements on interactive information to be satisfied by the interface to the intended artifact. Step 3 thus concludes the first main phase of the methodology.

Step 4

Step 4 consists in considering and applying the theoretically developed framework for representing the basic components of interactive unimodal or multimodal interfaces, i.e., the results of research on agenda items 1 and 2 in Sect. 1 above. This framework consists of a limited number of 'pure' or unimodal representational output types such as, e.g., written natural language text or sound diagrams, a number of structural output types which cut across different modalities such as icons or lists, analyses of the basic properties of these as well as a set of input modalities and analyses of their basic properties (Bernsen 1992a, Bernsen 1993a,b,c, May 1993a,b,c,d). The framework should eventually contain the elements needed for analysing any specific type of unimodal or multimodal input or output representation. The characterisation of each element is phrased in the same conceptual apparatus and terminology as that used in Step 3 above.

A mapping is performed of the results of Step 3 into the elements of Step 4. The result will be sets of possible input/output modalities and modality combinations which might be capable of representing the information needed for the representative tasks. It is likely that the mapping will often produce a number of alternative solutions which subsequently have to be judged and compared to identify an optimal set of input/output modalities for the representative tasks and hence for the artifact to be developed.

Step 5

In Step 5 a 'higher level filtering' is performed to trade off potential solutions against one another given the results of Steps 1 to 4 above. Preferably, this trade-off process should be explicitly represented in some form of Design Rationale representation (see below). The result of Step 5 is a solution to the task domain/interface mapping problem together with its Design Rationale. In some cases, several solutions can be expected to emerge from the trade-off process with identical scores.

3. What the Methodology Does and Does not Contribute

As always in usability engineering and HCI, there is no such thing as a stand-alone methodology which solves your problem just by itself. The methodology proposed here is no exception. While the methodology is intended for use early on in the design process, its effective use requires the concurrent application of low-level or relatively a-theoretical usability engineering approaches, heuristics and methodologies selected, e.g., from the following list:

- designer craft skills for usability problem prevention, detection and solution;
- written material on usability problems with previous systems of similar kinds;
- empirical results from field studies in the intended users' work environment;
- results from talking to intended users of the artifact;
- empirical results from observing users' task performance and from having them think aloud during task performance;
- results from using rapid prototyping methods of various kinds, from paper mock-ups through to semi-implemented systems and advanced, specialised methods such as the Wizard of Oz technique for natural language dialogue systems design;
- designing around basic tasks during user-system interaction development;
- scenarios and step-by-step designer walkthroughs on specified tasks;
- usability guidelines for heuristic artifact development and evaluation;
- prototype testing and iterative design.

The information-mapping methodology proposed here is compatible with the application of such approaches and may only produce valid results if combined with some of these. Thus, for instance, Steps 1-3 of the methodology rely heavily on the presence of sufficient information on the task domain of the intended artifact, and Steps 4 and 5 may easily generate a need for more such information. Since the success of the methodology rests on the identification of representative tasks, some way of verifying that the selected tasks (Step 1) really are sufficiently representative may be needed. Here again, some of the low-level methods just mentioned may be needed and prove useful.

Furthermore, as presented above, the proposed information-mapping methodology is not self-sufficient either with respect to higher-level usability engineering methodologies. Thus it was argued that Step 1 of the methodology would benefit from a method for generating representative scenarios. Moreover, the entire methodology would benefit from a notation for the explicit representation of evolving design space structure (see below). Step 5 of the methodology would benefit from some form of Design Rationale representation. And, finally, the information-mapping to be done in Step 4 and the trade-off design reasoning to be done in Step 5 might both benefit from many different kinds of science-based usability engineering support, such as user-modelling.

What the methodology itself does contribute is a step-wise procedure for applying results from the emerging discipline of modality theory (Bernsen 1993b) to the design of interactive human-computer interfaces. Whereas HCI-relevant studies in modality theory go back some decades already (e.g. Twyman 1979) and the GRACE agenda on information-mapping research is preceeded by interesting work (e.g., Hovy and Arens

1990), no methodology for applying modality theory to realistic design processes seems yet to have been proposed. We will now provide two case studies illustrating how the methodology might work in practice. We emphasise, once again, that much research on the GRACE agenda (Sect. 1 above), including much testing on real cases of interface design, still needs to be done to obtain a properly working methodology.

4. Case Study I: Spoken Language Dialogue

Let us begin by presenting a simple example from a national R&D design project in spoken language dialogue systems (Bernsen 1992b, Dybkjaer et al. 1993a, Dybkjaer et al. 1993b, Dybkjær and Dybkjær 1993). At an early stage in the design process, the designers faced the problem of the *realism* of the intended artifact. This problem turned out to entail an information-mapping sub-problem. The problem context is represented in DSD frame (x) below. DSD (Design Space Development) is a frame notation for succinctly representing, for usability engineering purposes, the design space structure as it evolves around an artifact during the design process (Bernsen 1993d, Bernsen 1993e).

DSD No. (x)

A. General constraints and criteria

Overall design goal:
- spoken language dialogue system prototype operating via the telephone and capable of replacing a human operator;

Design process type:
- experimental, application-oriented;

General feasibility constraints: N/A
Scientific and technological feasibility constraints: N/A
Designer preferences: N/A

Realism criteria:
- the artifact should meet real and/or known user needs;
- the artifact should be preferable to current technological alternatives;
- the artifact should be tolerably inferior to the human it replaces, i.e., it should be at least functional (to be expanded in the functionality criteria);

Functionality criteria: N/A
Usability criteria: N/A

B. Application of constraints and criteria to the artifact within the design space:

C =
O =
S =
I =

T = obtain information on and perform booking of flights between two specific cities;

U = Danish users;

E = novices (walk-up-and-use users); intermediates; experts;

C. Hypothetical issues:
- is a spoken telephone dialogue interface optimal?

D. Conventions:
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. ( ) indicates the number of the current DSD specification.

We need not go into the details of the DSD frame notation here. DSD frame (x) represents the overall design goal of the project; states that the design process is experimental and application-oriented; makes explicit the criteria of realism which apply to the intended artifact; defines the task domain (T) of the artifact as being that of obtaining information on and performing booking of flights between two specific cities; and specifies properties of the intended users of the artifact. The 'Hypothetical issues' section of the frame states the information-mapping sub-problem.

If the realism criteria cannot be met by the intended artifact there is no reason to build it in the first place, given the fact that we are dealing with an application-oriented design process. DSD (x) raises the design problem whether the overall design goal really meets that part of the realism criteria which states that the artifact should be preferable to current technological alternatives. A Design Rationale representation of this design problem, its solution by the designers and their reasoning towards the solution is provided below. The Design Rationale notation used here is a simplified version of the Design Rationale approach to design space analysis through Questions, Options and Criteria (MacLean et al. 1991, MacLean et al. 1993).

Design problem: is a spoken telephone dialogue interface preferable to current technological alternatives?
Option 1: telephone keystrokes for input, spoken language for output.
Option 2: screen, keyboard and mouse for input/output.
Option 3: spoken telephone dialogue.
Resolution: Option 3.
Justification: in all or most countries today, the telephone is already being used extensively for the selected tasks. In principle, these tasks might be mechanised instead by having users perform long series of telephone keystrokes or through the use of screen, keyboard and mouse. However, the first option implies rapidly decreasing usability as the task-relevant dialogues grow in complexity, and the second option is not (yet) available to most potential Danish users (cf. Bernsen 1992b).

In terms of our proposed methodology what has happened in this example is the following. Very early in the design process and after having identified a few critical criteria and constraints on the design space, the designers encountered the information-mapping problem represented in DSD frame (x) above. The identification of design criteria and constraints corresponds to Step 1 of the methodology. The DSD (x) representation of the problem corresponds to Step 3 of the methodology. The listing of three options for solving the problem corresponds to Step 4 of the methodology, and the justification for selecting a particular option among these corresponds to Step 5 of the methodology. Part of Step 1, i.e. scenario generation, and the whole of Step 2 of the methodology are not relevant in this case. The general point illustrated here is that *information-mapping problems arise at very different levels of generality during design.* For instance, they may be so general that there is no need to involve the generation and analysis of representative tasks to solve them.

A second important point illustrated here is that *the information which is relevant to the definition and resolution of an information-mapping problem derives from widely different aspects of the design space.* The general information-mapping problem between task/work domain and human-computer interface is emphatically not reducible to a problem about matching (a) the narrow information contents of the interactive messages between user and system during task performance into (b) the capabilities of information presentation and communication of different interface modalities. This is only one, albeit both interesting and difficult, part of the general information-mapping problem between task/work domain and human-computer interface. Rather, it would seem that the types of information relevant to the definition and resolution of a particular information-mapping problem may derive from any part of the design space. The DSD notation has in fact been designed to allow the explicit representation of all the types of constraint which need to be taken into account during design reasoning and decision making, including the making of design commitments on the nature of interfaces of designed artifacts (Bernsen 1993e).

What, if anything, might our information-mapping methodology have contributed to the resolution of the design problem in Case Study I? The answer seems to be that the methodology might primarily have assisted in identifying the three design options which were actually considered by the designers. These options would seem to exhaust current technological possibilities at the level of generality at which the information-mapping problem arose and was resolved. It would have been easy to obtain a more complex information-mapping design problem had we had a slightly more specific problem context. If, for instance, Option 2 had been chosen (i.e., screen, keyboard and mouse for input/output), then a number of new options would have had to be considered such as: is the mouse required? Is the keyboard? Should the input be through form-filling or through written natural language or both? Might we add spoken language as an option for input and/or output (cf. Nigay and Coutaz 1993, Coutaz et al. 1993)? It is an obvious advantage to be able to resolve information-mapping problems as early on in the design process as possible where options and constraints are few and general. However, as illustrated in this paragraph, even if one high-level information-mapping problem has been resolved, new information-mapping problems are likely to appear later on in the design specification process. A third, important observation, therefore, is that *information-mapping problems can be expected to arise continuously during the design process and all the way through to the implementation and testing stages.*

5. Case Study II: The Water Bath System

We now turn to the second case study which deals with a system considered in the process of developing an information-mapping methodology for control room applications prior to real system development in ESPRIT project PROMISE. The system is a toy system rather than a real application. This means that we have a certain freedom in specifying its properties and that it hardly makes sense to be fully specific in all respects.

The system consists of a container with an invalve (Vi) and an outvalve (Vo) which can be manipulated by the operator in order to control the flow rate (FR) of water through the system. The container is filled to a certain level (L) and a thermometer allows the operator to read off the temperature (T) of the water, which can be changed by controlling the heater (H) placed underneath the container. The goal of the operator is to drive and maintain the state variables (water temperature, flowrate, level) within predefined target ranges by manipulating the control variables (Vi, Vo, H) while taking into account that the state variables are not independent of one another. The system may appear quite simple at first sight and the control task is reminiscent of the filling of one's own bath tub. However, the state and control variables are interrelated in fairly complex ways. For example, by opening the invalve not only the level will rise, but also the flow rate will rise due to increased pressure in the system, and the temperature will decrease due to the added (cold) water (see Fig. 1).

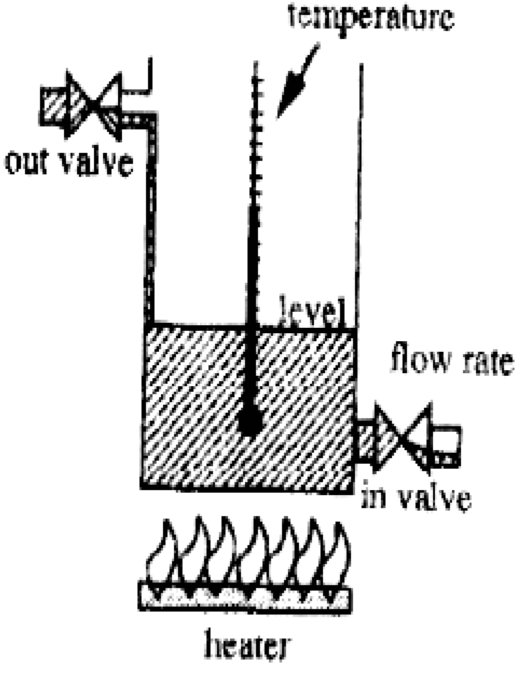

Figure 1: The Water Bath System

Figure 1. The Water Bath system.

The design problem is to identify the optimal input/output modalities for the system. Let us apply the information-mapping methodology proposed above.

Step 1

In this step we want (1.1) to identify some of the high-level information required to solve the information-mapping problem and, if necessary, (1.2) select one or several representative tasks for deeper analysis.

1.1 High-level information requirements

Since we are dealing with a toy system, we only list here a small number of important system characteristics which are representative of information that might be found by designers to be immediately and obviously relevant during early system specification. More constraints will be identified during the task analysis to follow. Clearly, designer teams will differ as to what they manage to identify during early system specification and what they identify later during detailed task analysis. Our list of important system characteristics is the following:

- the system is a toy multimodal system for methodology testing prior to real system development;
- the system is safety-critical, target values must be reached as quickly as possible;
- the control room environment is noisy;
- operators are professionals;
- the system must be monitored and controlled around the clock;
- operators will not always be within viewing distance of the screen.

1.2 Representative tasks

The water bath system can be used to perform a number of different tasks depending on whether one, two or three state variables have to be brought within a predefined target range. Let us assume that scenario generation results in the identification of the following single representative task: the operator has to bring the three state variables (T, L, FR) within range (Trange, Lrange, FRrange).

Step 2

In this step, (2.1) the selected representative task is analysed in as much detail as possible in order to complete the list of high-level task domain constraints as well as identify (2.2) the additional set of information-mapping requirements which derive from the task itself. The task analysis is informal but aims at being exhaustive as far as the information-mapping problem is concerned. There are several frameworks for task analysis in the literature which may be invoked to support Step 2 task analysis (e.g. Norman 1986).

2.1 Task analysis

Task goal: bring the state variables (T, L, FR) within range (Trange, Lrange, FRrange) as quickly as possible.
Initial state: In order to decide which action to take, the operator needs clearly perceivable information on the initial state of the system, i.e. on the state variables (SV: level, temperature and flow rate) and control variables (CV: invalve, outvalve and

setting of the heater), as well as on the target values. Let us assume that the initial state in the representative task is:

SV $T > Trange$
$L < Lrange$
$FR < FRrange$
CV H: On
Vi: Open
Vo: Open

In addition, in view of the safety-criticality of the system, there is a need for efficient (and hence salient) warning of the operator if the system's state values get outside of the target ranges.

Planning: The operator examines what needs to be done to reach the target ranges and scans his options based on training and experience. To reach the temperature range, he may either decrease H, increase Vi, decrease Vo or apply a combination of these measures. To reach the level range as well as the flow rate range (both being too low in the initial state), he may either increase Vi, decrease Vo or both.

Execution: Since the operator is controlling a safety-critical process, the goal state of the system must be achieved as quickly as possible without incurring unnecessary risk. The correct action is to increase the setting of the invalve and decrease the setting of the outvalve at once and, depending on the amount of temperature deviation from the target range, the heater setting may have to be decreased as well.

Feedback: Given the safety-criticality of the system, the operator needs immediate feedback on the results of actions performed.

2.2 Information characteristics

Let us now examine in greater detail the information characteristics of the state variables and control variables of the system. We shall use as heuristics an incomplete conceptual tool which has been developed for this purpose in Esprit project PROMISE.

The state variables (level, temperature and flow rate) all share the same characteristics:

- *quantitative, qualitative, ordinal, or nominal:* quantitative and not just ordinal because it does not always suffice to know that the value of a variable is larger than the goal range. It is often necessary to know exactly how much larger is the variable compared to the goal range. Thus, in our representative task above, increasing Vi will obviously make the level rise. If the temperature is only slightly too high, increasing Vi will cause the temperature to fall rapidly within range, but if the temperature is much too high the operator needs to change the heater setting as well.
- *discrete or continuous:* both representations can be used;
- *default detectability:criticality and relative importance:* the three state variables are highly critical and equally important;
- *coordinates or amounts:* amounts;
- *update rate of the variable:* as it is of central importance to the operator to know the exact state of the process at any given time, the information on temperature, level and flow rate needs a very high update rate;
- *variable range and stability:* the range covered is different for each state variable. The fluctuation range of the level is dependent on the size of the container. In

theory, the level may range from zero (empty container) to the value attached to a full container. The range of the flow rate partly depends on the size of the valves. The temperature range is large but, in practice, the fluctuation ranges of temperature, flow rate and level are highly dependent on their target ranges as the operator will try to maintain the process within the target ranges. As the target ranges of each variable are relatively small, it is unlikely that great accuracy is needed over a wide range. In other words, the parameter values may be assumed to be relatively stable;

- *current value, direction and rate of change:* the operator can obtain all these properties of each state variable from their current values;
- *size of the data set:* only three state variables need to be monitored at any given time;
- *temporal endurance: permanent or transient:* as the operator must monitor the evolution of the process, the data need to be permanently available.

The control variables (invalve, outvalve and setting of the heater) share the characteristics and requirements of the state variables except that the control variables do not need updating since they can only be changed manually. The control variables are not equally important. The invalve and outvalve settings are somewhat more important than that of the heater, since they are the only means of control of the flow rate. The water temperature can be influenced both by the setting of the heater and by the invalve and outvalve settings.

The only relevant characteristic of the warning messages is that they are highly critical.

Step 3

In this step we want to explicitly represent the relevant information acquired through Steps 1 and 2. This has been done in DSD frame (y) below.

DSD No. (y)

A. General constraints and criteria

Overall design goal:
- safety-critical water bath system;

Design process type:
- toy multimodal system for methodology testing prior to real system development;

General feasibility constraints: Nil
Scientific and technological feasibility constraints: Nil
Designer preferences: Nil
Realism criteria: N/A

Functionality criteria:
- make sure that the artifact makes it possible for users to perform their tasks;

Usability criteria:
- optimise the usability of, and hence user performance on, the artifact;

B. Application of constraints and criteria to the artifact within the design space:

C = Monitoring and control around the clock by different operators;

O = Operators not always within viewing distance of the screen;
S =

I = *output information:*
Clearly perceivable display of values = SV (T, L, FR) and CV (H, Vi, Vo);
Clearly perceivable display of target value ranges (Trange, Lrange, FRrange);
Provide immediate feedback on results of control actions;
Provide salient warnings when state values get outside the target ranges;

information characteristics:
Quantitative;
Discrete or continuous;
Default detectability high for SV;
Amounts;
Update rate high;
Great accuracy around target ranges, otherwise moderate accuracy; Current values constantly available;
Permanent display of target range values;

input information:
Allow change of control variables (H, Vi, Vo);

T = Bring state variables (T, L, FR) within range (Trange, Lrange, FRrange) as quickly as possible by manipulating the control variables based on perception of initial state;
Maintain state variables within range;
Continuous monitoring and control;
Noisy control room;

U =

E = Experts;

C. Hypothetical issues:
- which input-output modality combination is optimal for the representative task?

D. Conventions:
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. ( ) indicates the number of the current CO-SITUE specification.

Step 4

In this step we consider (4.1) the taxonomy of basic components of interactive unimodal or multimodal interfaces and apply (4.2) the taxonomy through information-mapping with respect to the intended artifact. This process generates sets of input/output modalities which obey the constraints stated in DSD (y) of Step 3 above. As indicated earlier, we shall ignore the problem of correct input modality selection.

4.1 The taxonomy

The taxonomy currently includes a large number of generic unimodal information (output) representations drawn from the media of graphics, sound and touch and is claimed to be both exclusive and exhaustive at the level of generality at which it operates (Bernsen 1993a). In practice, it would seem natural that designers start by focusing on a subset of the taxonomy at the exclusion of modalities which appear obviously irrelevant to the application at hand, either because such modalities clearly cannot represent the information needed or for technological or other feasibility reasons. Similarly, the designers are of course free to combine unimodal (UM) output representations into multimodal (MM) representations which appear to be likely information-mapping candidates for the application. Assuming this screening process we obtain the following list of possible unimodal or multimodal output modalities:
- *spoken language (UM);*
- static written language (UM);
- static diagrammatic pictures annotated in static written *or spoken* language (MM);
- animated diagrammatic pictures annotated in static written *or spoken* language (MM);
- static graphs combining a graph space and static written *or spoken* language annotation (MM);
- dynamic graphs combining a graph space and static written *or spoken* language annotation (MM);
- *arbitrary, diagrammatic or realistic sound (UM).*

However, the noise level in the control room (cf. DSD (y)) excludes use of the sound medium for output information representation, thus eliminating spoken language and other uses of sound from the above list (italicised in the list). This creates a problem about how to represent warning messages (see below). An important goal of ongoing taxonomy development is to be able to characterise the modalities of the taxonomy using the same concepts and terminology as that used in characterising the information needs of intended artifacts. Since we are not yet ready for doing that at either 'end' of the information-mapping process, we shall assume that the taxonomy provides the following characterisation of the modalities we need to consider:

1. Static written language
can express data that are:
- nominal, qualitative, **quantitative** or ordinal;
- **discrete**;
- coordinates or **amounts**;

and has the following characteristics:
- **permanent**, changing in discrete steps;
- low default detectability;
- **high update rate** possible;
- **high accuracy** possible;
- **clear perceivability** possible;

2. Static diagrammatic pictures annotated in static written language
can express data that are:
- nominal, qualitative, **quantitative** or ordinal;
- **discrete**, continuous;
- **amounts**;

and have the following characteristics:
- **permanent**, changing in discrete steps;
- low default detectability;
- medium high update rate possible;
- **high accuracy** possible;
- **clear perceivability** possible;

3. Animated diagrammatic pictures annotated in static written language
can express data that are:
- nominal, qualitative, **quantitative** or ordinal;
- **discrete**, continuous;
- **amounts**;

and have the following characteristics:
- **permanent** or continuously changing;
- **high default detectability**;
- **high update rate** possible;
- **high accuracy** possible;
- **clear perceivability** possible;

4. Static graphs combining a graph space and static written language annotation
can express data that are:
- nominal, qualitative, **quantitative** or ordinal;
- **discrete**, continuous;
- coordinates or **amounts**;

and have the following characteristics:
- **permanent**, changing in discrete steps;
- low default detectability;
- medium high update rate possible;
- **high accuracy** possible;
- **clear perceivability** possible;

5. Dynamic graphs combining a graph space and static written language annotation
can express data that are:
- nominal, qualitative, **quantitative** or ordinal;
- **discrete**, continuous;

- coordinates or **amounts**;

and have the following characteristics:

- **permanent** or continuously changing;
- **high default detectability**;
- **high update rate** possible,
- **high accuracy** possible;
- **clear perceivability** possible;

4.2 Information mapping

In this step the characteristics of the selected modalities (Sect. 4.1) and variables (Sect. 3) are mapped against one another. We shall start from the modalities available and examine whether they are capable of representing the variables. However, the mapping process may also be done in reverse order, starting from the variables and checking their properties against those of the modalities. It may not matter much which order is preferred in a particular case. To simplify the mapping process it seems natural to focus on points where a specific modality is *not* able to represent the desired information since that may be sufficient to exclude the modality from further consideration. This is immediately evident from the output modality listing in Sect. 4.1 above. Boldface indicates that a modality conforms to an information representation requirement.

It appears that only the dynamic modalities are able to conform to all the representational requirements of the state variables. The control variables do not need a high update rate and might in principle be represented by the non-dynamic modalities. This leaves us with the following picture:

Static written language is suitable for representing the control variables. It cannot represent the state variables because it lacks high default detectability.

Static diagrammatic pictures annotated in static written language and *static graphs combining a graph space and static written language annotation* are suitable for representing the control variables. They cannot represent the state variables because they lack high default detectability. In addition, their update rate is too slow.

Animated diagrammatic pictures annotated in static written language and *dynamic graphs combining a graph space and static written language annotation* are capable of representing the state variables because of their high default detectability and high update rates.

This leaves us with the following options for further consideration in Step 5 below:

For representing the control variables:
- static written language;
- static diagrams;
- static graphs;
- dynamic diagrams;
- dynamic graphs.

For representing the state variables:
- animated diagrams;
- dynamic graphs.

If we return to the relevant part of DSD (y) we see that most output information requirements have been taken care of (marked by asterisks):

B. Application of constraints and criteria to the artifact within the design space:

C = Monitoring and control around the clock by different operators;

O = Operators not always within viewing distance of the screen;
S =

I = *output information:*
* Clearly perceivable display of values = SV (T, L, FR) and CV (H, Vi, Vo);
* Clearly perceivable display of target value ranges (Trange, Lrange, FRrange);
* Provide immediate feedback on results of control actions;
Provide salient warnings when state values get outside the target ranges;

information characteristics:
* Quantitative;
* Discrete or continuous;
* Default detectability high for SV;
* Amounts;
* Update rate high;
* Great accuracy around target ranges, otherwise moderate accuracy;
* Current values constantly available;
Permanent display of target range values;

input information:
Allow change of control variables (H, Vi, Vo);

T = Bring state variables (T, L, FR) within range (Trange, Lrange, FRrange) as quickly as possible by manipulating the control variables based on perception of initial state;
Maintain state variables within range;
Continuous monitoring and control;
* Noisy control room;

U =

E = Experts;

What remains to be dealt with are primarily (a) how to provide salient warnings when the state values get outside their target ranges and (b) how to permanently display the target range values of the state variables. The use of sound having been excluded, warnings seem to have to be provided by graphical means. Since both issues ((a) and (b)) are to do with the state variables, it would seem preferable to make their representation intrinsic to the chosen representation of the state variables. In addition, one might consider introducing a separate, highly salient, graphical global warning function such as, e.g., a dynamically changing warning icon which may tell the operator to urgently inspect the state variables representations in order to identify the one(s) which are out of range. However, given the fact that operators cannot be expected to be always within viewing distance of the screen, this second, global warning function might preferably be realised in the form of a sound loud enough to penetrate the control room noise level. In that case a single, loud arbitrary sound which cannot be mistaken for anything else would seem preferable. Other sound representations, such as spoken language, diagrammatic or realistic sound (cf. 4.1 above) impose increased demands on auditory discrimination and therefore work badly in noisy environments.

Step 5
In this step, a trade-off among the potential solutions from Step 4 is performed resulting in either a design commitment to implement one of those solutions or in the realisation that additional information is required to allow a design commitment to be made.

Let us start by considering the possible state variables representations:

Design problem: How should the state variables be represented?
Option 1: Animated diagrams.
Option 2: Dynamic graphs.
Resolution: Option 2.
Justification: Both animated diagrams (such as, e.g., thermometer diagrams) and dynamic graphs are able to incorporate representations of target range values and salient warning features. However, only dynamic graphs are able to incorporate 'history-at-a-glance'. When operators change (cf. the C-component of DSD (y)), the new operator shift might want to inspect how the state variables have evolved in the previous period. Dynamic graphs are a very efficient means of compactly representing such developmental information whereas animated diagrams are not.

As to the control variables which are to be changed manually by the operators, it would seem difficult to make a principled choice without also considering a set of input modality options. We refrain from attempting to do that in this paper.

6. Discussion

During the last couple of years, the field of HCI/usability engineering has entered a fascinating new stage of development and we are beginning to be able to concretely specify the requirements for the field as a mature applied science (cf. Carroll 1993; for a pioneering collection of papers see Carroll 1991). On top of the existing, widely used 'toolkit' of low-level usability engineering methods (cf. Sect. 3 above), a new layer of design support methods are emerging which serve to make the design process explicit in terms of design space structure and development as well as in terms of the designer reasoning which operates in the design space and drives its development. Examples are the DSD framework and the simplified Design Rationale approach illustrated in this paper. Improved methods for scenario generation would appear to belong to this same methodological layer (cf. Sect. 2 above). However, while adding much needed perspicuity and explicitness to the design process, this new methodological layer in itself contributes little in terms of basic science. Rather, it acts as a series of bridging representations (Barnard 1991) between basic science and practical design. Arguably, the provision of more explicit structure and contents to design processes is a precondition for the systematic application of basic science to the solution of usability problems in computer artifact design.

Our proposed information-mapping methodology constitutes yet another bridging representation between basic science and practical design. The basic science contribution is the implementation of the GRACE research agenda (Sect. 1 above). This exploratory paper has shown, first of all, that the GRACE research agenda for the comparatively new field of modality theory is in one piece. The information-mapping methodology needs a comprehensive analysis of both output and input modalities (i.e., agenda items 1 and 2). Furthermore, the results of these analyses need to be extensively tested on real interface design problems in order to shape the contribution of basic science to the needs of practical design (i.e., agenda item 3). Secondly, it is clear that the methodology needs substantial further research including supporting research on scenario generation and task analysis. Also, the PROMISE approach to interface information requirements for process control applications (Sect. 5 above) clearly cannot be generalised to all other types of human-computer interfaces.

References

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

Bernsen 1992a: Bernsen, N.O.: Matching Information and Interface Modalities. An Example Study.*Working Papers in Cognitive Science* WPCS-92-1, Centre of Cognitive Science, Roskilde University 1992. *Esprit Basic Research project GRACE Deliverable* 2.1.1, 1993.

Bernsen 1992b: Bernsen, N.O.: Design of a Spoken Language Dialogue System. A Study of the Initial Specification Phase. *Working Papers in Cognitive Science* WPCS-92-5. Centre for Cognitive Science, Roskilde University, 1992.

Bernsen 1993a: Bernsen, N.O.: Foundations of multimodal representations: A taxonomy of pure modalities. *Esprit Basic Research project GRACE Deliverable* 2.1.2, 1993. To appear in *Interacting with Computers*.

Bernsen 1993b: Bernsen, N.O.: Specificity and Focus. Two Complementary Aspects of Analogue Graphics and Natural Language. *Esprit Basic Research project GRACE Deliverable* 2.1.3, 1993.

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

Bernsen 1993d: Bernsen, N.O.: Structuring design spaces. *InterCHI '93 Adjunct Proceedings*, 211-12, 1993.

Bernsen 1993e: Bernsen, N.O.: The structure of the design space. In Byerley, P.F., Barnard, P.J. and May, J. (Eds.): *Computers, Communication and Usability: Design issues, research and methods for integrated services.* Amsterdam, North-Holland, 1993, 221-244.

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

Carroll, J. M., Kellogg, W. A. 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, 181-212.

Carroll, J.M.: Creating a design science of human-computer interaction. *Interacting with Computers* Vol. 5, No. 1, March 1993, 3-12.

Coutaz, J., Nigay, L. and Salber, D.: The MSM framework design space for multi-sensory-motor systems. To appear in Moran, T.P. and Carroll, J.M. (Eds.): *Design Rationale.* Hillsdale, NJ: Erlbaum 1993 (in press).

Dybkjaer et al. 1993a: Dybkjaer, L., Bernsen, N.O. and Dybkjaer, H.: Knowledge acquisition for a constrained speech system using WoZ. *Proceedings of the Sixth Conference of the EACL,* 21-23 April 1993, 467.

Dybkjaer et al. 1993b: Dybkjaer, H., Bernsen, N.O. and Dybkjaer, L.: Wizard-of-Oz and the trade-off between naturalness and recogniser constraints. *Proceedings of EUROSPEECH '93,* Berlin, September 1993, 947-50.

Dybkjær, L. and Dybkjær, H.: Wizard of Oz Experiments in the Development of the Dialogue Model for P1. Report 3, Spoken Language Dialogue Systems, STC Aalborg University, CCI Roskilde University, CST University of Copenhagen, 1993.

Hovy, E. and Arens, Y.: When is a picture worth a thousand words? Allocation of modalities in multimedia communication. Paper presented at the *AAAI Symposium on Human-Computer Interfaces,* Stanford 1990.

Klausen, T. and Bernsen, N.O.: CO-SITUE: Towards a methodology for constructing scenarios. In Hollnagel, E. and Lind, M. (Eds.): *Proceedings of the Fourth European Meeting on Cognitive Science Approaches to Process Control (CSAPC '93): Designing for Simplicity.* Copenhagen, August 1993, 1-16.

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

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

May 1993a: May, M.: Representations and homomorphisms. A taxonomy of representations for HCI, part 1. *Esprit Basic Research project GRACE Deliverable* 2.1.5, 1993.

May 1993b: May, M.: Levels of representations and mappings. A taxonomy of representations for HCI, part 2. *Esprit Basic Research project GRACE Deliverable* 2.1.6, 1993.

May 1993c: May, M.: From semantic types to multimodal presentations: Case-studies. A taxonomy of representations for HCI, part 3. *Esprit Basic Research project GRACE Deliverable* 2.1.7, 1993.

May 1993d: May, M.: Taxonomy and levels of mappings in the construction of multimodal interfaces. 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 1993.

Nigay, L. and Coutaz, J.: A design space for multimodal systems: Concurrent processing and data fusion. *Proceedings of InterCHI '93,* 172-78.

Norman, D.A.: Cognitive engineering. In Norman, D.A. and Draper, S.W. (Eds.): *User Centered System Design,* Hillsdale, N.J.: Erlbaum 1986, 31-62.

Twyman, M.: A schema for the study of graphic language. In Kolers, P., Wrolstad, M. and Bouna, H. (Eds.): *Processing of Visual Language* Vol. 1. New York: Plenum Press 1979.

Acknowledgements: The work described in this paper was carried out under Esprit Basic Research project 6296 GRACE whose support is gratefully acknowledged.

List of publications in the series
"Working Papers in Cognitive Science"

1991

91-1 *Kognitionsforskning ved AUC*. Per Hasle.
91-2 *Johnson-Laird's Theory of Mental Models*. Niels Ole Bernsen.
91-3 *Potential Flashbulbs: Memories of Ordinary News as the Baseline*. Steen F. Larsen.
91-4 *"Icons Make me Click" - The Problem of Semantics*. Ib Ulbæk.
91-5 *Mental Models in Human-Computer Interaction*. Niels Ole Bernsen.
91-6 *Two Games in Town. Systematicity in Distributed Connectionist Systems*. Niels Ole Bernsen and Ib Ulbæk.
91-7 *Bevidsthedsfilosofi og kognitionsforskning*. Ib Ulbæk.

1992

92-1 *Matching Information and Interface Modalities. An Example Study*. Niels Ole Bernsen.
92-2 *Systematicity, Thought and Attention in a Distributed Connectionist System*. Niels Ole Bernsen and Ib Ulbæk.
92-3 *Communication Failure and Mental Models*. Brynja Svane and Niels Ole Bernsen.
92-4 *Kognitiv semantik og diagrammatisk tænken: fra intuition til formalisering*. Ph.D. Thesis. Michael May.
92-5 *Design of a Spoken Language Dialogue System. A Study of the Initial Specification Phase*. Niels Ole Bernsen.
92-6 *Learning the time constants of a Dynamic Task*. Bernd Brehmer, Leif Løvborg, and Anders Winman.
92-7 *Cost Information in a Dynamic Decision Task*. Bernd Brehmer, Leif Løvborg, and Anders Winman.
92-8 *New Facilities in NEWFIRE*. Leif Løvborg and Bernd Brehmer.
92-9 *C.S. Peirce on the Possibility of a Temporal Logic*. Peter Øhrstrøm.
92-10 *Hypertid*. Peter Bøgh Andersen og Peter Øhrstrøm.

1993

93-1 *A logical approach to handling uncertainty*. Morten Elvang-Gøransson, Paul Krause, John Fox.
93-3 *Den kognitive revolution*. Niels Ole Bernsen.
93-4 *Systematicity in the vision to language chain*. Niels Ole Bernsen.
93-5 *The structure of the design space*. Niels Ole Bernsen.
93-6 *Errors in a team context.* James Reason and Niels Ole Bernsen.
93-7 *Talking to a Wizard: report from the design of a natural speech understanding system*. Tove Klausen.
93-8 *Eksempler på samspillet mellem simultane og sekventielle kognitive processer.* John Paulin Hansen.
93-9 *Foundations of Multimodal Representations. A Taxonomy of Pure Modalities*. Niels Ole Bernsen.

93-10 *A methodology for mapping information from task domains to interactive modalities*. Niels Ole Bernsen.
93-11 *A research agenda for modality theory.* Niels Ole Bernsen.
93-12 *Types of user probems in design.* Niels Ole Bernsen.