

1994

## Supporting Design Communication with Explicit Representation of Evaluation Feedback

Torbjörn Näslund

*Linköping University, Sweden, tor@ida.liu.se*

Follow this and additional works at: <http://aisel.aisnet.org/sjis>

---

### Recommended Citation

Näslund, Torbjörn (1994) "Supporting Design Communication with Explicit Representation of Evaluation Feedback," *Scandinavian Journal of Information Systems*: Vol. 6 : Iss. 2 , Article 5.

Available at: <http://aisel.aisnet.org/sjis/vol6/iss2/5>

This material is brought to you by the Journals at AIS Electronic Library (AISeL). It has been accepted for inclusion in Scandinavian Journal of Information Systems by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact [elibrary@aisnet.org](mailto:elibrary@aisnet.org).

# Supporting Design Communication with Explicit Representation of Evaluation Feedback

**Torbjörn Näslund**

*Department of Computer and Information Science at Linköping University  
S-581 83 Linköping, Sweden  
tor@ida.liu.se*

---

## Abstract

*Usability is an important characteristic for information systems intended to support their users. In order to achieve usability, an iterative change of focus between design and formative evaluation during development is an often recommended approach. Our experience from industrial projects is, however, that communication problems and misinterpretations often occur during this change of focus. It is not sufficient that the evaluation reports give feedback—they must also support the re-design work. One promising attempt to achieve such evaluation reports would be to*

*use Design Space Analysis, which is an approach to design rationale representation. In this paper, we take an excerpt from a real-life evaluation report from one of our industrial projects, and show the implications of using a more structured approach to make the arguments in the evaluation report more explicit.*

*Key words: Usability evaluation, formative evaluation, design rationale, iterative development, argumentative design.*

## 1. Introduction

A computerized information system is usable if its users can perform their tasks effectively, efficiently and comfortably (cf. Booth 1989, Hix & Hartson 1993, Lewis & Rieman 1993). Usability is thus a necessary characteristic for any information system. It can be more or less difficult to achieve during systems development. It is particularly difficult to achieve for information systems which are intended to function as tools in advanced human activities. An investigation into the concept of usability and its characteristics for decision support systems has earlier been published in this journal (Ovaska 1991).

It is well recognised that iterative development is important for achieving usability (Gould & Lewis 1985). Iterative development encompasses an iterative shift between design and evaluation, where design is the creative task of exploring an idea of the future system, and evaluation is a systematic reflection on the qualities of explored ideas. Multiple persons' knowledge is regularly needed both for design and for evaluation in information systems development. Some of the evaluation activities should be performed by other people than the designers themselves, in order to challenge the design and avoid risks for blindness in design and design fixation (Winograd & Flores 1986, Finke *et al.* 1992).

For a relatively long time, there has been HCI research on *usability evaluation*. While usability evaluation has earlier had a strong flavour of quantitative research methods taken from experimental psychology, there has been increasing interest in finding simpler evaluation methods with a primary aim of guiding

information systems development towards usability. "Discount methods" (Nielsen 1993) have emerged as a term stressing the effectiveness of the evaluation methods in information systems development in business and industry. Discussions of the relative merits of different evaluation methods are common (Jeffries *et al.* 1991, Desurvire *et al.* 1992, Jeffries & Desurvire 1992, Burger & Apperley 1991). In systems development, the intention is that the evaluations should be *formative* (i.e., inform the design process), and that they should provide the needed information despite tight limits of cost and time.

Let us use the term *design proposal* for an explicit representation of a design idea, so that it can be understood by someone else than the designer. Research and experience have shown that for ensuring usability of an information system under development, three complementary approaches to usability evaluation can be used:

- Evaluation by prospective users, where the users are given the opportunity to assess design proposals in relation to their own experiences.
- Empirical evaluation, where design proposals are tested by representative users solving representative tasks.
- Analytical evaluation, where expert knowledge is used for assessing design proposals.

The three approaches are complementary, and can only to a very limited degree substitute each other. Users are experts in their own work practices, although they often have difficulties in describing them, as well as to see alternative ways of performing their work. (For an ap-

proach to cooperatively challenging work practices, see Mogensen 1994). Empirical evaluations are important for detecting unanticipated troubles when using an information system (Desurvire *et al.* 1992). It is often very hard for designers to anticipate what difficulties users will have when using an information system (Wright & Monk 1991). Analytical evaluations are important for applying existing knowledge about design and use. (Nielsen 1993, Polson *et al.* 1992). (For discussions of the complementary character of empirical and analytical evaluations, see Jeffries & Desurvire 1992, Karat *et al.* 1992, and Nielsen & Phillips 1993)

In this paper, the focus is on analytical usability evaluation. Analytical evaluation is a necessary complement to the other two approaches since it can be used early in the development process, it can apply important existing knowledge on design proposals at low cost, and it can reduce the number of design flaws which would distract users when the two other approaches are used. The recommendation is that analytical evaluations precede empirical evaluations.

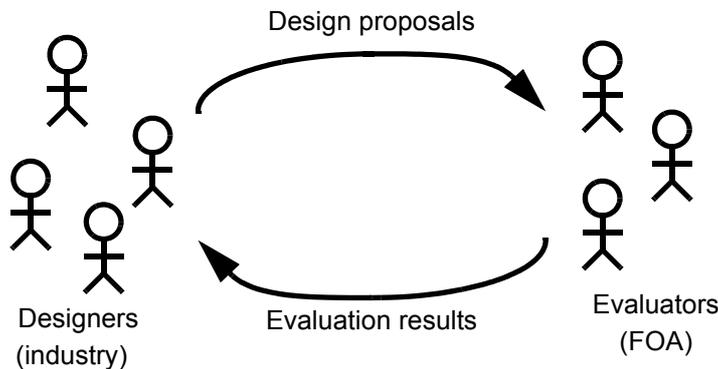
In all three approaches, the evaluation result must be fed back from the persons who perform the evaluation to the developers, so that the evaluation result can inform the further design of the system. In the first approach, the users are giving their feedback in a dialogue with the designers, while in the two other approaches the feedback is given by the usability evaluators which are responsible for the evaluations. The evaluation results are usually summarized in an evaluation report.

## **2. Experiences from analytical usability evaluation in industrial information systems development**

The work of systems design has changed considerably in industrial information systems development during the last years. Many developers are given new opportunities for designing information systems, including options such as multi-window, multi-colour and multi-media graphical user interfaces with multiple mechanisms for user input. For many developers, this implies a large increase in the number of design options to select among, compared to those offered by earlier technology. The demands for design knowledge and skills which many of them have not yet acquired are high. A large number of design mistakes are done. Analytical usability evaluations can help in finding these mistakes early, and draw attention to important design issues early in the development process. In usability-oriented design, there is an early focus on how the information system should be designed in order to fulfill the needs in the actual use situation. Usability is not seen as a technical issue or an issue of designing a user interface layer to a program product. The objective for usability-oriented design is instead to develop artifacts with a good match between technology, supporting documentation, user characteristics, and the supported task and organization.

In cooperation with the National Defence Research Establishment (Försvarets Forskningsanstalt, FOA) we have been involved in several real-life development projects, where the aim has been to develop usable support systems, most of them for support in military command and control. In these projects,

FIGURE 1. Basic roles in the analytical evaluations



the design work was performed by professional information systems developers in industry. Since it was known that several earlier development efforts in the area had resulted in applications which had deficiencies in usability, we tried to reduce the inherent uncertainty in the development process by providing formative evaluations during the process (Näslund 1992a, 1992b).

The work was consequently split; the development activities were performed by industrial developers, while the evaluation activities were performed by us at FOA. This split forces many issues to be stated explicitly, which supports reflection and discussions about the design suggestions. Figure 1 illustrates the two basic roles in the communication process we are discussing here. The evaluation team consists of people with several competencies, such as domain expertise and persons trained in usability, information systems development and psychology.

It was a deliberate choice that design work should be performed by the designers, and that we as evaluators should give

informative comments on the design, but not take over as designers by providing the developers with our own design solutions.

In the projects, we performed both analytical and empirical evaluations, but with a clear bias towards analytical evaluations. The bias can partially be explained by project characteristics, but it could also be noted that the project leaders expressed preference for analytical evaluations. One belief which was expressed several times was that participation of user representatives in design could somehow be a substitute for empirical evaluation. Our experience is that this is not correct. User representatives are very valuable sources of domain knowledge; their participation is also valuable for gaining acceptance and smooth implementation of the application in the work setting. They are, however, not very good at anticipating user behaviour, and this is what would be needed if user participation would be a viable substitute for empirical evaluations. Our experience from the projects is that user participation in design has be-

come accepted and used in development practice. The argument that this user involvement can reduce the need for empirical evaluation can, however, be an obstacle for the development of usable systems.

The evaluation results have been fed back to designers in the form of evaluation reports. These reports have often been rather lengthy, textual descriptions (sometimes complemented by simple sketches) describing what part of the design the evaluators wanted to draw attention to, as well as a description and/or a discussion of the identified problems with the proposed design.

The formative evaluations resulted in large amounts of reported deficiencies and highlighted design issues. Despite the fact that the developers were professionals, it was very easy to find a large amount of questionable design solutions, also when using very simple evaluation methods. Large parts of the evaluation reports dealt with relatively simple deficiencies such as lack of feedback to the user, use of a technical language rather than the language of the application domain, and lack of consistency in design. Most deficiencies of this kind severely decrease the usability of an application, but have the advantage of being relatively easy to correct once they are detected. Among the deficiencies which are harder to find a late remedy for, we could note a tendency among the developers to make design decisions based on their own inferences about the task and work procedures—inferences which often were neither made explicit nor checked with domain expertise.

It was consequently easy for the evaluators to perform useful evaluations and the evaluation reports contained a large

amount of important information for the continued development of the application. The communication between the developers and the evaluators was more cumbersome than expected, however. The evaluation reports tended to be rather lengthy, in order to describe the findings. This caused both a large amount of work in writing and reading the reports, and a risk that important issues were swamped in large amounts of text. Several issues were repeatedly brought to attention in subsequent evaluation reports. In these cases, either the reported problem remained in the subsequent design, or the new design replaced one drawback with another.

The conclusion from these projects is that analytical usability evaluations have a large *potential* for increasing usability in information systems, since it was easy to detect design flaws in the project. Since the detection and communication of these flaws were not sufficient for ensuring that design flaws were avoided in the subsequent design, some issues must be investigated further:

- There is a need for investigating how the communication between the evaluators and the designers can be improved. Since the evaluation reports tended to be lengthy, *increased overview* of the reports would probably be an improvement.
- There is a need for investigating how to help the designers to find new design proposals which are better than the previous. *Support for argumentation* about the merits and consequences of design proposals would probably help.

These issues are currently not well understood. The issues appear to be on the

border of different research areas. Research in human computer interaction has investigated how to perform usability evaluations, but lack systematic knowledge on how these evaluations relate to the whole development process and the different actors involved. Research in information systems has focused on the development process, and has acquired important knowledge on working with users (Greenbaum & Kyng 1991). Issues about empirical and analytical usability evaluations do not seem to have been incorporated, however. Research in software engineering, finally, has made thorough investigations into verification and validation during construction of software, but demands that the requirements of an information system are defined early in the development process. This would be extremely difficult for usability issues. Regrettably, there is a substantial gap between techniques advocated for software engineering and techniques advocated for design with usability in focus.

Similar issues have received attention earlier regarding other parts of the design process, however. Parnas & Clements (1986) note that there are problems in understanding the rationale behind design decisions, since rejected design proposals and the reasons for rejection are not documented. They also state other problems with documentation from the design process. Approaches for supporting the explicit recording and communication of decisions and trade-offs during design have begun to emerge in the area called Design Rationale (Preece *et al.* 1994, chapter 26; Human Computer Interaction 1991).

### 3. Using a design rationale approach for improving evaluation feedback

The remaining part of this paper discusses the use of a new format for structuring evaluation reports, using a design rationale approach called Design Space Analysis (DSA). The hypothesis is that design rationale approaches can help in structuring and clarifying evaluation reports, in order to increase the designers' opportunities to understand and make use of the evaluation results.

The work reported here is a post hoc analysis of existing evaluation reports from an industrial project, in which I was a member of the usability evaluation team. The evaluation reports were textual. During the project, we could note problems of the kind described in the previous section. A question is thus what consequences the use of a design rationale technique could have had. During the analysis, I have used the QOC notation from Design Space Analysis in order to study the implications of using the notation.

Design rationale is the common term for a set of approaches for capturing design argumentation and design decisions during the development of an artifact. Some well known approaches are IBIS (Grant 1976, Conklin & Begeman 1989), DRL (Lee & Lai 1991) and Design Space Analysis, DSA (MacLean *et al.* 1989, MacLean *et al.* 1991). For a comparison of approaches, see Shum (1991) or Sjöberg (1994). Several approaches are described in the special issue of Human-Computer Interaction (1991). Preece *et al.* (1994) discuss design rationale in a separate chapter.

The primary reason for choosing Design Space Analysis (DSA) was its ex-

licit recognition of design rationale for exploring design spaces, i.e. the set of possible design options. This view fits very well with the problem many professional system developers face today, with an increasing number of design opportunities to choose from.

Of course, the impact of using techniques such as DSA for communicating the results of formative evaluations requires empirical studies in real-life projects. Such studies are currently performed, and will be subsequently reported. The post-hoc analysis of evaluation reports reported here is an early part of this long term research effort.

The next section of the paper, section 4, introduces the QOC notation used in DSA. Section 5 describes the common pattern found in the analysis of the evaluation reports. Section 6 gives a detailed example of how one small part of an evaluation report could be represented in the QOC notation, and highlights opportunities and problems encountered in this re-representation process. Section 7 summarizes the experiences.

#### 4. The QOC Notation

QOC is short for “Questions, Options, and Criteria”, which are the three basic concepts used in a design space analysis. The notation was developed by researchers at RankXerox EuroPARC (MacLean *et al.* 1989, MacLean *et al.* 1991).

We can think of a design space as all the design possibilities for an artifact to be designed, each with its own characteristics. Obviously, many of these possibilities would not be satisfactory to work with; others would be extremely cumbersome to accomplish during systems

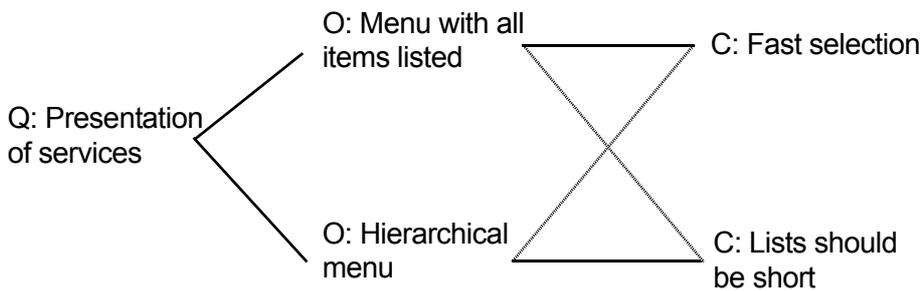
development. Among the more reasonable design possibilities, it will probably be difficult to state which is the best; they will each have their advantages and disadvantages. The intention with a design space analysis is to investigate and represent the characteristics of different design possibilities, and thus to increase the awareness of the trade-offs involved in decision-making about design.

A *question* is a particular design issue, to which there are different design *options*. Each of these options can be positively or negatively assessed by different *criteria*. These criteria thus describe the characteristics of the different options in relation to each other.

Let us study a simple example of how to use QOC for representing an exploration of a design space: Assume that we have a design situation, where we as designers need to design a dialogue where the user should select one of twenty different services. The first option we think of is to design a menu, with all services listed. Such a list would however be very long, and it would be difficult for the user to get an overview. As an alternative option, we consider grouping the services, and design the dialogue based on a menu hierarchy instead. We regard this option as better since it avoids very long menus, but due to problems we have with slow computer response, we are concerned that traversing a menu hierarchy would slow down the dialogue so much that the user would get irritated.

This line of reasoning can be illustrated using the QOC notation (figure 2). As we can see, there are two identified options, each of them positively assessed to one criterion and negatively assessed to another criterion. If one of the two options should be selected in a design deci-

FIGURE 2. Early QOC structure for the example

**Question****Options****Criteria**

sion, there will be a trade-off between different criteria. Which of the two criteria that will be most important will be specific to the situation at hand. The representation can, however, also be used as an aid in searching for new options: Are there any options which would be positively assessed by both criteria?

One such new option would be to avoid using menus, and instead using a command-based dialogue. A command based dialogue will achieve high speed also for large sets of alternatives to choose from, without the need for presenting very long menus. In a comparison between the three options we will easily detect that we need another criterion, however. A disadvantage with using a command language compared to menus is that the user needs to remember the different commands, while in a menu based interaction, the user will always be informed about the alternatives. The extended QOC structure would be as in figure 3.

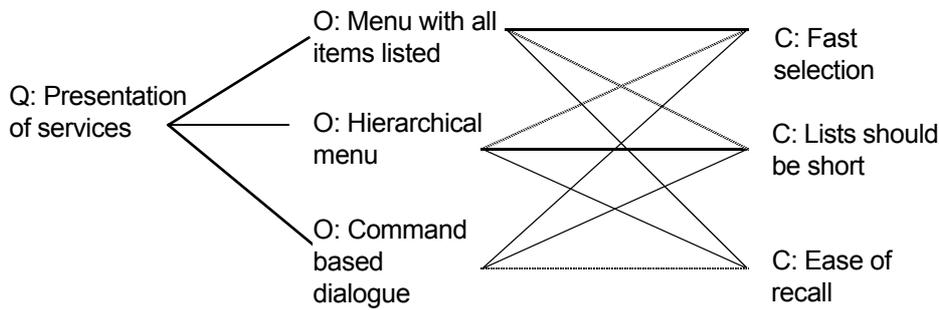
This rather simplistic example was given in order to introduce the QOC notation. Several different design options addressing the same design question are

compared. The comparison is made by assessing the options relative to a set of relevant criteria. There are positive and negative assessments; the positive assessments are illustrated with solid lines, and the negative assessments with a dotted line. When a design decision has been reached, it is common to illustrate this in the QOC structure by a rectangle around the selected option.

It should be noted that a design decision can never be *deduced* from a QOC structure. The structure illustrates the trade-offs involved, but cannot show the importance of different criteria, nor the strengths of the assessments. As an example, it would not be appropriate to select an option just because it is positively assessed by ten criteria and negatively assessed by only one criterion—the single negative assessment could rule out the option completely.

It should also be noted that the example is not meant to argue about the best solution to the simplistic design problem. A common solution used in many modern interfaces is to use cascading menus complemented with keyboard short-cuts. Another option which would

FIGURE 3. Extended QOC structure for the example



be positively assessed by all three identified criteria, is to use a command language but to provide a clearly shown way for the user to ask for help if the appropriate command is forgotten. However, what is a good design decision will always depend on the situation at hand.

### 5. Analysis of the evaluation reports from a DSA perspective

Several evaluation reports from our formative evaluation efforts in the development projects were analyzed using the DSA framework. Compared to a design space analysis, the evaluation reports contained several systematic differences:

1. Often, the evaluation reports considered only one design option, namely the design option which was suggested by the designer. In DSA, several options are contrasted to each other.
2. The comments were often based on a comparison between the suggested design option and a perceived objective of the application. In DSA, the criteria highlights differences in

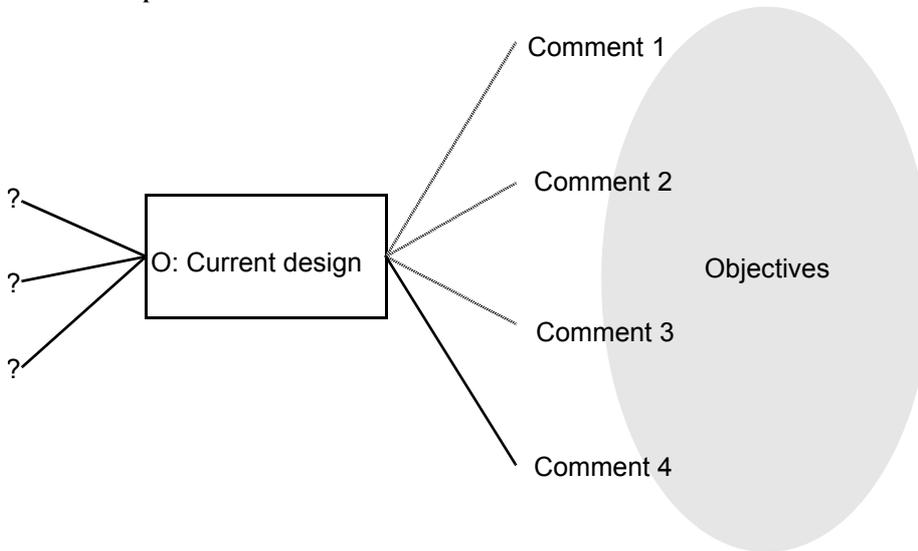
characteristics between the different options, but usually do not connect these criteria to the actual objectives of the design project. Neither the evaluation reports nor DSA explicitly deals with project objectives, although both do so implicitly.

3. The comments often highlighted a misfit between an objective and the suggested design solution. In other words, negative relationships between objectives and the design option were emphasized, while positive relationships were seldom mentioned.
4. While DSA groups several design options around a design question, the evaluation reports rather tended to comment on each design option, connecting it to one *or more* design questions. These design questions were seldom mentioned explicitly.

Using the QOC notation would yield something like the sketch in figure 4, with the additional complication of the connection between the comments and the project objectives indicated.

Consequently, seen from a DSA viewpoint, the evaluation reports did not

**FIGURE 4. Sketch of a generic QOC representation for the comments in analyzed evaluation reports**



explicitly mention the addressed question and since there was only one option to comment on, there were seldom explicit discussions about criteria.

Each of the identified differences between the content of the evaluations reports and DSA can easily be explained. Using the same numbers as above, the following rationalization for the structure of the evaluation reports can be given.

1. Since it was a deliberate choice of the evaluators not to take over the designer role, the evaluation reports could only comment on the suggested design options. The evaluators did in fact ask the designers to supply several design options in cases where the designer thought it was hard to make a design decision, but this seldom occurred in practice. Instead, it seems like the designers preferred to make design decisions,

and then to revise them in the light of negative reactions from colleagues, evaluators or user representatives.

2. The comparison between objectives and design solutions is central for all kinds of evaluation activities. Since DSA highlights differences between options, but does not offer means for making design decisions (i.e., trading the different criteria off against each other), its structure must be enhanced with objectives if it should support design decisions and evaluations. The evaluation reports, on the other hand, did not need to position the design option in a design space, since it was seldom needed to contrast different options against each other.
3. Since the evaluation reports tended to be rather lengthy, priority was

given to reporting found deficiencies over reporting objectives which were in fact fulfilled by the design. This trade-off seems to be appropriate, since there is probably an infinite set of positive comments which can be given to a suggested design. In fact, it seems that many of the cases where positive comments were in fact given in evaluation reports, it was done for the purpose of either giving a notification that a new design option solved a problem highlighted in an earlier evaluation, or simply to make the tone of the evaluation report somewhat milder after having given a set of negative comments on a design option.

4. It was considered convenient both for the evaluators and the designers to structure the evaluation reports around the services provided by the application. This led to the situation where several design questions could be commented together in cases where they were all resolved by the same designed service.

The structure of the evaluation reports clearly makes sense. The evaluation reports contain what they promised to contain, namely a structured set of evaluation results or comments on a suggested design. It would be wrong to argue that the evaluation reports were bad, just because their structure did not conform to the structure of a design space analysis.

Given the identified problems with the utilization of the evaluation report, the situation becomes somewhat different. As discussed in section 2, above, it was not uncommon that new design solutions contained the same or new design deficiencies. There is a need to strength-

en the use of evaluation reports to support *re*-design, still without moving the responsibility of design decisions from the designers to the evaluators. It was also identified that the lengthy evaluation reports created communication problems. The time for production of the evaluation reports was sometimes longer than the time for the actual evaluation sessions. The QOC structure has a potential for both increasing the overview of what is discussed in the evaluation reports and to provide support for argumentation which was asked for in section 2.

#### **6. An example: applying DSA to an existing evaluation report**

To illustrate the implications of using the QOC notation in communication of evaluation results, let us look at an example in some depth. The example is taken from a real-life, industrial development project, in which the author took part as an evaluator. It is selected partly on its merits of being easy to understand taken out of its context, and partly because the actual evaluation report in this passage was brief and concise. It is thus intended to be a good illustration, but it should be noted that most other parts of the evaluation reports contained longer discussions of highlighted issues than this example does.

The evaluated application can be described as an office information system tailored for use by military staffs. In military staffs, much of the work centres around geographical information. *The map* consequently plays a central role in the work. The application has several services for handling maps with military

**TABLE 1. Excerpt of an evaluation report**

- |   |
|---|
| <p>(1) Only RT-coordinates seem to be implemented.</p> <p>(2) RT-coordinates are not displayed in a correct format (X/Y format should be used. See "Army handbook 4").</p> <p>(3) The precision of the coordinates is generally too high.</p> <p>(4) There is no advice-giving text, explaining how to click the mouse buttons.</p> <p>(5) It would be advantageous if the selected positions were marked on the map.</p> <p>(6) When a sequence of coordinates is given, it is hardly practical to wait until the last position is given. Another possibility would be to display each coordinate after each marked position.</p> <p>(7) The two last remarks make it difficult to find a position on a map when the coordinates are known. Such situations will be common, since the system lacks a function for showing the position corresponding to given coordinates.</p> |
|---|

information overlaid. An important service for working with the map is the conversion between a position on the map and the geographical coordinates of this position. This is the service focused in the example.

In Swedish military work, three different systems are used for describing the coordinates of a position. Of them, "Rikets nät" is the most commonly used, and is the decided standard for use in the Swedish army. Coordinates in "Rikets nät" are called "RT-coordinates." (As an example, *X 647646 Y 149049* are the RT-coordinates for my apartment in Linköping).

The excerpt above is taken from the evaluation report written after an analytical evaluation of an early prototype of the application<sup>1</sup>. The service under consideration is the service for providing the coordinates to a position. The user invokes the service by first selecting "Position analysis" from a menu, then selecting the geographical system to be used for expressing the coordinates, and finally by pointing on the map and clicking the left mouse-button. The coordinates will then be displayed in a separate window. When coordinates are needed for

several positions, the user can make a corresponding selection, then click the left mouse-button for each of the positions, and finally click the right button to mark the end of the sequence and get the list of all the coordinates.

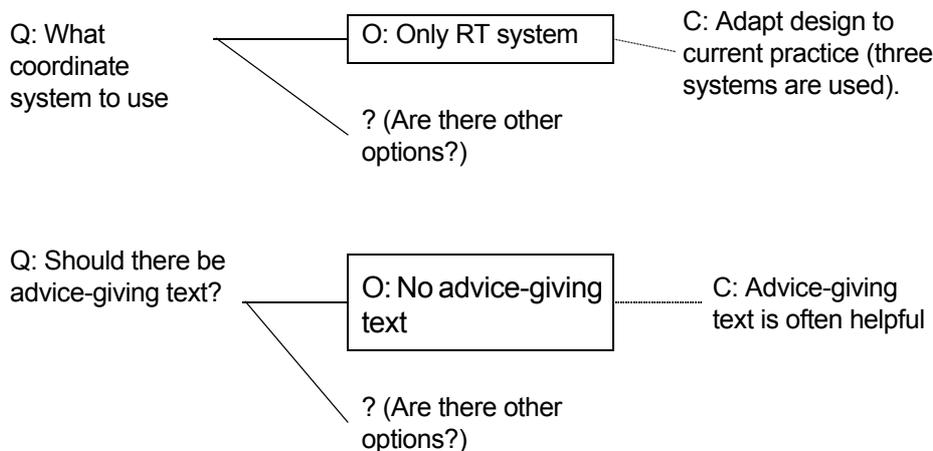
The evaluation report contained the text in Table 1, translated from Swedish<sup>2</sup>.

It can be noted that the formulation of the comments is not very elegant, but this is not an uncommon situation. An important demand on formative evaluation is that the comments are needed shortly after an evaluation session. Delay of the feedback makes it much less useful. Thus, formulations leading to misunderstanding should be avoided, while inelegant but understandable formulations can pass.

#### 6.1. Comments 1 and 4

Prototypes are incomplete models of a proposed system. A very common situation when evaluating prototypes is that there is a lack of statement of the status of the current version compared to the intended final system. This makes it difficult for the evaluators to decide if missing pieces are forgotten and hence need

FIGURE 5. QOC structure for comments 1 and 4.

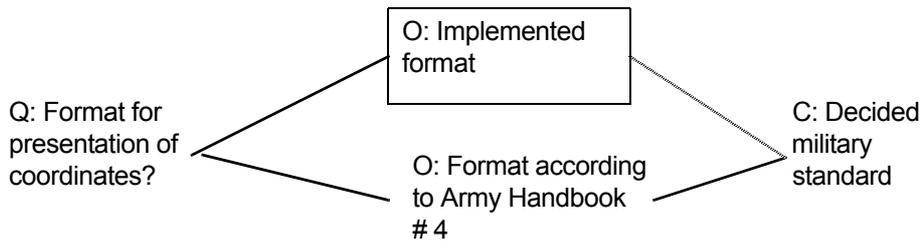


to be highlighted in the evaluation report, or if they are deliberately omitted in the current prototype and are planned to be added later. The most frequent solution to this dilemma during our projects has been to stress the lacking functionality, but not spend many words on explaining why the functionality is needed. A typical example can be found in comment (1). Here, the designers probably had a deliberate plan to investigate the service with only one of the three geographical coordinate systems implemented, and then to add the other two later in the development. It is more questionable whether the same kind of reasoning is true for the other comments, such as (4). Clearly, it could be the case that the developers had the intention to first wait for the evaluators comments on a given design, and then add instructions to the user on how to use the service. It could, however, also be the case that the designers simply forgot the advice, or thought that it was not needed since it was evident to the users how to use the service<sup>3</sup>.

Both comments can be seen as statements where the evaluators prompt for either a consideration of other design options, or an explanation for why the suggested option is good. It is not the intention of the evaluators to start an extensive discussion unless the designers show that this is needed. Figure 5 tries to illustrate both comments using the QOC notation.

This is an example of how it often can be difficult to know for the evaluators what issues should be brought into focus, and when it should be brought into focus. To achieve the maximal effect, the evaluators' comments should neither be too late, i.e., when the designers already have committed themselves to a design proposal, nor too early, i.e. when the designers have decided to explore the particular issue later. It is thus important that the designers state what qualities they want to show by a prototype or model, and what qualities they want to explore later. The notion of *question* in DSA can be used by the designers for making ex-

FIGURE 6. QOC structure for comment 2



explicit what they want to highlight and what they do not want to discuss at the current state. Questions can also be used by the evaluators to stress what issues need to be brought into focus, now or at a later stage. A list of open questions can thus be maintained as an *agenda* for design discussions.

#### 6.2. Comment 2

- (2) RT-coordinates are not displayed in a correct format (X/Y format should be used. See "Army handbook 4").

When sketched using the QOC notation, comment 2 will be rather simple, as shown in figure 6.

Although it is not very clear from the text, two different design options are in fact discussed—one which is actually implemented, and one which is suggested by the evaluators. One criterion is identified. The implemented format has a negative link to the criterion, while the suggested format has a positive link to the criterion. Note that this does not necessarily mean that the *best* design decision is to use the format given in the Army Handbook. Rather, the QOC struc-

ture should be used for asking the designers the question "Are there any other criteria which speak in favour of your implemented format?" Examples of such criteria could be that there are severe technical difficulties to implement the standard format, or that the user representatives have given a strong argument for why the implemented format would be more useful for them.

#### 6.3. Comment 3

- (3) The precision of the coordinates is generally too high.

Comment 3 is more delicate. The textual structure of the comment is something like the structure shown in figure 7. Here we can note that just in order to be able to present a QOC structure, we need to add information, namely the precision used in the implemented design. In the evaluation report, it was obviously assumed that the readers were aware of the details of the implemented solution—an assumption which was probably true, but anyway forced the reader to make an additional effort when reading the report, and, as was just shown, also decreases the comprehensibility of the evaluation

FIGURE 7. QOC structure for comment 3

Q: Precision to use for display of coordinates?



(C: Seldom need for precision of one meter)

report for later use. Let us assume here what was probably also the case, namely that the system specified the exact position in meters.

It also becomes clear that the criterion given in the evaluation report is vague. In order to make a thorough investigation of the design space around this, better formulations of the criteria are needed. The two probably most important criteria would be (a) the demands of the military tasks being performed, and (b) the practical possibilities to make a precise selection of a position on the map using the mouse. Clearly, it is of limited use to know exactly to the meter where, for instance, a company is, since a company consists of approximately 150 soldiers, and will thus be hard to fit into a meter squared. It will also be of limited use to present the coordinates with higher precision than the precision with which the user will be able to mark the position on the map.

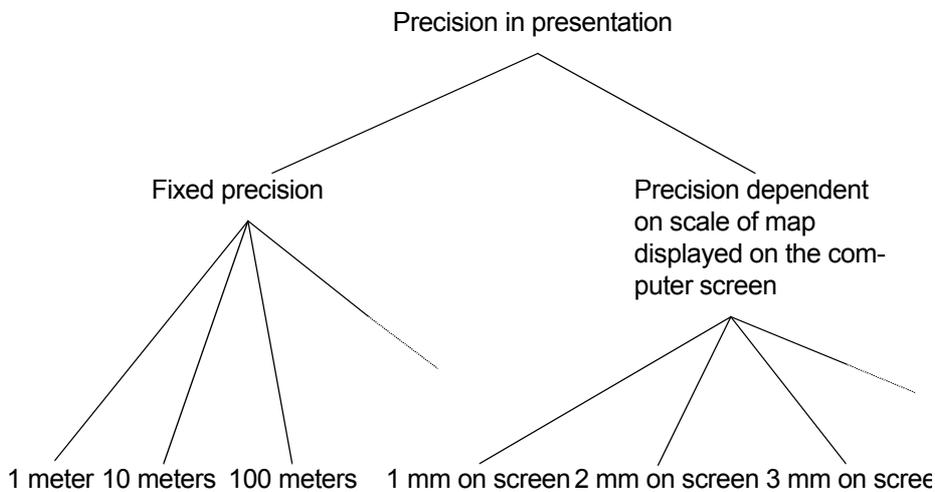
An interesting thing happens when these two criteria are used in assessing different design options. The first identified design options were a set of different precision values such as the set {1 m, 10 m, 100 m, 1 km}. Neither of these options can be directly assessed by the criteria. A designer or evaluator with domain knowledge will soon realize that the two criteria are best tied, not to the precision of coordinates in the world *per se*, but rather to a function of the *scale* of the actual map. On a detailed map, it will

of course be much easier to mark a precise position than on an overview map. Furthermore, an officer who selects an overview map most likely performs a task with less need for precision than an officer who selects a detailed map. The point of interest is that this was easily realized when the two identified criteria were used to assess the initial set of options {1 m, 10 m, 100 m, 1 km}. The increased set of identified options could instead be something like the structure shown in figure 8, with one set of options describing options with fixed precision and one set describing scale-dependent options.

A criterion such as “match to precision of pointing” would be possible to use to assess the second set of options, but would not be applicable to the first set since the match would vary with the actual scale of the presented map. In a similar way, a criterion such as “task match” would be easier to apply to the second group than to the first group. Consequently, the explicit use of criteria not only makes it easier to find new options, but rather *forces* the recognition that an option is unsuitable since the assessment of criteria cannot be made.

The example illustrates both how QOC forces explicit formulation of important characteristics and how the explicit assessment of options to criteria may clarify the issue also for the evaluator and in fact force the discovery of new and better options.

**FIGURE 8. Options for precision in presentation**



6.4. Comment 5

- (5) It would be advantageous if the selected positions were marked on the map.

Comment 5 highlights that something is missing, but does not give any argument for why it should be there. From a DSA viewpoint, the *question* and the *options* are relatively clear, but there are no *criteria*. A suitable QOC structure, such as the one in figure 9, could provide a better backing for the comment. In this case it is made explicit that the argued advantages for marking the positions on the map are to reduce the risk for erroneous interpretation of the given coordinates, and to reduce the load on the user's memory.<sup>4</sup>

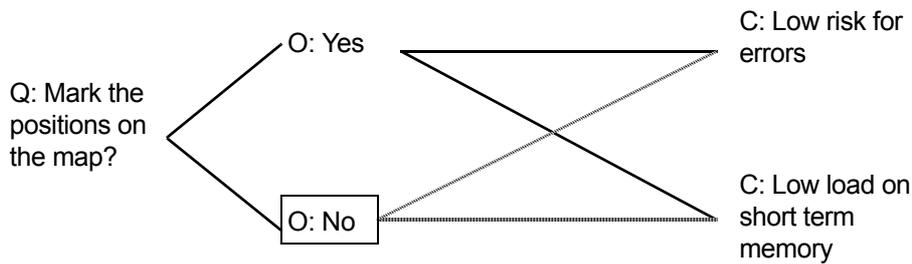
Note that the suggested design option creates a need for further design decisions, such as the rather difficult ques-

tion of when to remove the mark from the map.

6.5. Comments 6 and 7

- (6) When a sequence of coordinates is given, it is hardly practical to wait until the last position is given. Another possibility would be to display each coordinate after each marked position.
- (7) The two last remarks [i.e. 5 & 6] make it difficult to find a position on a map when the coordinates are known. Such situations will be common, since the system lacks a function for showing the position corresponding to given coordinates.

FIGURE 9. Extended QOC structure for comment 5



Comment 6 argues, in a surprisingly mild tone, that the designers should consider also another option for the presentation of the coordinates for several positions. Given the information in comment 7, the suggested design is actually not good at all. In order to find the position given by a pair of coordinates the user must employ iterative refinement. The iterative refinement will be considerably more efficient if the coordinates are displayed immediately for each marked position.

The mild tone in the evaluation report can be explained by the fact that the designers need to reconsider the different options. There are several possible decisions to be made, and what is a suitable decision is dependent on the complete set of decisions. The *questions* which need to be investigated are for example “How to find a position, given the coordinates?”, “How to get the coordinates for several positions?” and “When should the coordinates be displayed if using the ‘several-position-function’?” These questions should be addressed together, and in the given order. Comment 6 is mild, since it is a comment to a specific option, but where the evaluators did not want to stress a specific design option since there are more questions and

options to investigate before an assessment can be done.

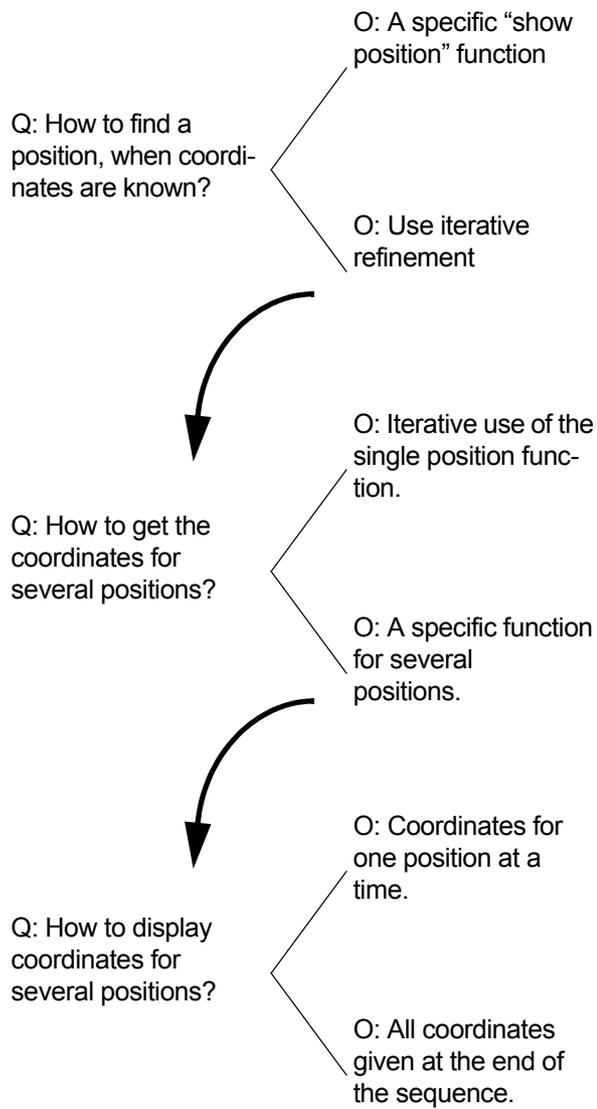
Figure 10 illustrates these three linked questions, and some possible design options for resolving these questions, but does not illustrate any criteria. The message this conveys is that these are design decisions which are not yet ready for formative evaluation. The evaluators would perform too much of the design work if they would suggest also the important criteria for these questions. The evaluators should not *take over* the design work from the responsible designers.

## 7. Discussion

In the example, we have seen some of the effects of representing the design comments from an existing evaluation report by using the QOC notation taken from Design Space Analysis (MacLean *et al.* 1989, MacLean *et al.* 1991). In summary, the following advantages of using the QOC structure in evaluation reports can be foreseen:

- The criteria can be made more explicit, which can improve the designer’s understanding of why a particular design has deficiencies. In

**FIGURE 10. Linked questions to explore for improving the design commented upon by comments 6 and 7**



many circumstances the designer will also be provided with an excellent tool for assessing design trade-offs when considering design changes.

- Since the QOC structure facilitates considerations of new options, it can be made more clear to the designers that it is not always necessary to make minor modifications to a criticized design. A completely different design option can be the proper solution to a given problem. This might not have been so easy for the designer to see when the evaluation report only contained comments on a proposed design.
- Undetected questions, options, and criteria can be discovered during the construction of the QOC structure. The construction of a QOC structure is thus not only a means for communication, but also a tool for the evaluation and design work.
- The QOC notation provides a graphic overview of the arguments in the evaluation report. Since a serious problem with the evaluation reports was the extensive amount of text, the provision of easily understandable graphical overview is likely to improve the communication of evaluation results. In many cases, the QOC can reduce the required amount of text, since the notation can explain also relatively complex relationships in the design space.

Since design work should be performed by the designers and not by the evaluators, the QOC structures will in many circumstances only be partial. The structures can in these situations be seen as a prompt for the designers to “fill in

the gaps” (as an example, see the explicit prompt for consideration of new options in figure 5). The iterative process of design and evaluation can thus be seen as a cooperative design discussion, in which the design space is gradually explored. The QOC structure can both provide a tool for the work, and serve as documentation of the activity.

Another question is what qualities QOC does not have but which would be important improvements when it is used for supporting communication between evaluators and designer. Two major limitations can be identified from the example:

- QOC is useful for illustrating the design space, but for evaluation purposes it would be an improvement if also the objectives driving the development effort could be represented. The QOC structure can show the identified trade-offs involved when choosing among design options. For evaluation purposes and design decision making, the actual objectives can help in clarifying the relative importance of each criterion (cf. Goldkuhl 1991).
- QOC does not convey information about the status of the evaluated application. In the standard QOC notation, there is a possibility to show which of the options is the design decision. For use in evaluation reports, it will instead be important to show the status of the options, such as “currently implemented option”, “planned, but not yet implemented option”, “recommended option”, etc.
- Information about what tasks a certain service is intended to support is

vital for making good formative evaluations in usability-centred design. A possible extension of a QOC structure could include also this information. Since this information primarily is important for communication from the designers to the evaluators, such an extension should have its primary value in development projects where there is a vital dialogue between designers and evaluators about design issues. (This information is important also for the designers themselves, see Herbsleb & Kuwana 1993).

An evaluation report also contains a lot of other issues which are not easily shown using DSA. Of course, the evaluator should not try to squeeze everything into a QOC representation—the QOC notation should be a help, not a hindrance.

## 8. Conclusions and Further Work

Usability-oriented design requires extensive exploration of design alternatives in order to ensure the usability of information systems which are produced. Our earlier work (Näslund 1992a, 1992b) shows that reports from formative evaluations can contain important information for the continued development process. There has, however, been problems with the utilization of this information. It is not enough to find problems with a design; in order to ensure usability, the communication of the evaluation results and the subsequent redesign process is crucial. An interpretation of the encountered problems is that they are due both to troubles with the communication be-

tween evaluators and designers, and to difficulties for the designers to find good remedies to reported shortcomings.

The use of the QOC notation appears to have a potential to decrease these problems. In this paper, we have used an example from an existing evaluation report to show how the notation can help in clarifying the discussion, and how it can provide a handle for the designers' continued work. The work shows that the explicit representation of design questions, design options and design criteria can provide an overview of what are the important issues in an evaluation report. Through the explicitness, it can also support the understanding of the relationships between design options, and support the discovery of new design possibilities. The use of partial QOC structures can provide a handle for subsequent design discussions, leading to an elaboration of the design rationale.

Whether this approach is useful in real-life projects needs to be further investigated. We are currently studying the practical implications of using this more structured approach to formulate evaluation feedback in an industrial information systems development project.

## Notes

<sup>1</sup>To use an empirical evaluation at this stage would in most cases be to misuse the time and enthusiasm of the users.

<sup>2</sup>The numbers have been added during translation, in order to make references easier. The original Swedish version can be obtained from the author.

<sup>3</sup>In the discussed project, it was not considered suitable for the evaluators to investigate the reason a designer had for making a design change. If a comment such as comment 4 was not applicable to the design solution presented in the next design iteration, the evaluators felt satisfied, whatever the

reasons for the change could be.

<sup>4</sup>A common recommendation for the use of DSA is to avoid Yes/No-questions in design exploration (Shum 1991). Here, a better question for design exploration could thus be "How could the system give the user information about the mapping between coordinates and position on the map?" or "How could the system provide feedback to the user on what position is selected?" The reason for using a Yes/No-question in the example is that this best maps the actual comment given by the evaluators.

### Acknowledgements

This work was supported in part by the National Defence Research Establishment (FOA), and in part by the Swedish National Board for Industrial Development (NUTEK).

Jonas Löwgren, Göran Goldkuhl, Tone Bratteteig, Göran Forslund, Pär Carlshamre and two anonymous reviewers provided valuable comments on earlier drafts of this paper. Trond Knudsen provided valuable suggestions for our further work.

### References

- Booth, P., (1989). *An Introduction to Human-Computer Interaction*. Lawrence Erlbaum, Hove.
- Burger, S. & M. Apperley, (1991). A Multi-Dimensional Approach to Interface Evaluation. In: M. I. Nurminen *et al.* (editors), *Proceedings of Human Jobs and Computer Interfaces Conference*. University of Tampere, Tampere.
- Conklin, J. & M. Begeman, (1989). gIBIS: A Hypertext Tool for Exploratory Policy Discussion. *ACM Transaction on Office Information Systems*, 6(4): 303-331.
- Desurvire, H., J. Kondziela & M. Atwood, (1992). What is Gained and Lost when Using Evaluation Methods Other than Empirical Testing? In: A. Monk *et al.* (editors), *People and Computers VII (Proceedings of HCI 92)*. Cambridge University press, Cambridge.
- Finke, R., T. Ward & S. Smith, (1992). *Creative Cognition: Theory, Research and Applications*. MIT Press, Cambridge.
- Goldkuhl, G., (1991). Information Systems Design as Argumentation: An investigation into design rationale as a conceptualization of design. In: K. Ivanov (editor), *Proceeding of 14th IRIS (Information systems research in Scandinavia)*. Umeå universitet, Umeå.
- Gould, J. & C. Lewis, (1985). Designing for Usability: Key principles and what designers think. *Communications of the ACM*, 28(3): 300-311.
- Grant, D. P., (1976). How to Use the IBIS as a Procedure for Deliberation and Argument in Environmental Design Planning. *Design Methods and Theories*, 11(4).
- Greenbaum, J. & M. Kyng, (editors), (1991). *Design at Work: Cooperative design of computer systems*. Lawrence Erlbaum Associates, Hillsdale.
- Herbsleb, J. & E. Kuwana, (1993). Preserving Knowledge in Design Projects: What designers need to know. In: S. Ashlund *et al.* (editors), *Human Factors in Computing Systems (InterCHI'93 Proceedings)*, ACM, New York.
- Hix, D. & H. R. Hartson, (1993). *Developing User Interfaces: Ensuring usability through product & process*. John Wiley & Sons, New York.
- Human Computer Interaction, (1991). *Human-Computer Interaction*, 6(3-4). Special issue on Design Rationale.
- Jeffries, R. & H. Desurvire, (1992). Usability Testing vs. Heuristic Evaluation: Was there a contest? *SIGCHI Bulletin*, 24(4): 39-41.
- Jeffries, R., J. Miller, C. Wharton & K. Uyeda, (1991). User Interface Evaluation in the Real World: A comparison of four techniques. In: S. P. Robertson *et al.* (editors), *Human Factors in Computing Sys-*

- tems (*CHI'91 Proceedings*). ACM, New York.
- Karat, C.-M., R. Campbell & T. Fiegel, (1992). Comparison of Empirical Testing and Walkthrough Methods in User Interface Evaluation. In: P. Bauersfeld *et al.* (editors), *Human Factors in Computing Systems (CHI'92 Proceedings)*. ACM, New York.
- Lee, J. & K. Lai, (1991). What's in Design Rationale? *Human Computer Interaction*, 6(3-4): 251-280.
- Lewis C. & J. Rieman, (1993). *Task-centered User Interface Design*. Shareware book; available via anonymous ftp from ftp.cs.colorado.edu (Internet).
- MacLean, A., R. Young & T. Moran, (1989). Design Rationale: The argument behind the artifact. In: K. Bice & C. Lewis (editors), *Human Factors in Computing Systems (CHI'89 Proceedings)*. ACM, New York.
- MacLean, A., R. Young, V. Bellotti & T. Moran, (1991). Questions, Options, and Criteria: Elements of design space analysis. *Human Computer Interaction*, 6(3-4): 201-205.
- Mogensen, P., (1994). Challenging Practice: An approach to cooperative analysis. Thesis DAIMI PB-456. Computer Science Department, Aarhus University, Århus.
- Näslund, T., (1992a). On the Role of Evaluations in Iterative Development of Managerial Support Systems. Licentiate thesis. Linköping Studies in Science and Technology No 335. Linköping University, Linköping.
- Näslund, T., (1992b). Injective Formative Evaluation for Enhancing Computer Support for Cooperative Staff Work. In: *Proceedings of 8th symposium on Human Interface*. Kawasaki.
- Nielsen, J., (1993). *Usability engineering*. Academic Press, San Diego.
- Nielsen J., & V. Phillips, (1993). Estimating the Relative Usability of Two Interfaces: Heuristic, formal, and empirical methods compared. In: S. Ashlund *et al.* (editors), *Human Factors in Computing Systems (InterCHI'93 Proceedings)*. ACM, New York.
- Ovaska, S., (1991). Usability as a Goal for the Design of Computer Systems. *Scandinavian Journal of Information Systems*, 3: 47-62.
- Parnas D. & P. Clements, (1986). A Rational Design Process: How and why to fake it. *IEEE Transactions on Software Engineering*, SE-12(2): 251-257.
- Polson, P., C. Lewis, J. Rieman & C. Wharton, (1992). Cognitive Walkthroughs: A method for theory-based evaluation of user interfaces. *International Journal of Man-Machine Studies*, 36(5): 741-773.
- Preece J., Y. Rogers, H. Sharp, D. Benyon, S. Holland & T. Carey, (1994). *Human-Computer Interaction*. Addison Wesley, Wokingham.
- Shum, S., (1991). A Cognitive Analysis of Design Rationale Representation. Doctoral dissertation. Department of Psychology, University of York, York. (Available from Librarian, RankXerox EuroPARC, 61 Regent Street, Cambridge CB2 1AB, UK).
- Sjöberg, C., (1994). Voices in Design: Argumentation in participatory development. Licentiate Thesis. Linköping Studies in Science and Technology No 436. Linköping University, Linköping.
- Winograd, T. & F. Flores, (1986). *Understanding Computers and Cognition: A new foundation for design*. Ablex, Norwood.
- Wright P. & A. Monk, (1991). A Cost-effective Evaluation Method for Use by Designers. *International Journal of Man-Machine Studies*, 35(6): 891-912.