Erschienen in: Behaviour & Information Technology ; 16 (1997), 4-5. - S. 232-245

Software evaluation using the 9241 evaluator

REINHARD OPPERMANN† and HARALD REITERER‡

†German National Research Centre for Computer Science (GMD), Institute for Applied Information Technology FIT, D-53754 Sankt Augustin, Germany; email oppermann@gmd.de

‡University of Konstanz, Department of Information Science, D-78434 Konstanz, Germany; email Harald.Reiterer@inf-wiss.uni-konstanz.de

Abstract. There is an increasing need for practical and comprehensive evaluation methods and tools for conformance testing with ISO standards. In this study, we focus on ISO 9241 which is an important ergonomic standard. A brief description shows its content and structure. Practical evaluations include the amount of time and resources which must be managed in software projects, while comprehensive evaluations require that the context of use be considered during the evaluation of user interfaces. In order to complete a comprehensive evaluation of usability, it is necessary to use more than one evaluation method. Therefore, an overview of different evaluation approaches is given, describing their advantages and disadvantages. Finally, the ISO 9241 evaluator is presented in detail as an example of a practical expert-based evaluation method for conformance testing with the ISO 9241 standard, that can be integrated in a comprehensive evaluation approach.

1. Introduction

Software-ergonomic evaluation is aimed at assessing a system's degree of usability. The criteria of the evaluation can be established by a theory or by pragmatic considerations (e.g., by a political or industrial community responsible for defining general acceptable standards). Standards are based on empirical evidence and practical and economical feasibility. Not all desiderata from research are accepted as standards. For example, there may be objections due to technical or financial constraints.

In some cases, the application of standards is enforced by law. The European Union (EU), for example, published the directive 90/270/EWG concerning the minimum safety and health requirements for VDT workers (EEC 1990) to establish common working conditions for users of visual display terminals. The national governments participating in the EU have transformed this directive into national law. The international standardization activities of ISO 9241 concerning ergonomic requirements for visual display terminals form the basis which define the relevant technological requirements necessary to fulfill the directive.

An important consequence of these standardization activities is that software developers and buyers need ergonomic requirements and principles. To assure the conformance of products with the established standards, practical software evaluation methods are needed. In this paper, an expert support method for evaluating user interfaces according to the ISO 9241 standard is presented and compared with other evaluation methods.

2. Standards

ISO 9241, the 'Ergonomic requirements for office work with visual display terminals (VDTs)', is far from being a pure technical standard that can be assured by quantitative measures. Rather, it requires interpretation and tailoring to be useful in user interface evaluation and reflection about the state-of-the-art technology in research and development. It is subject to an ongoing process of discussion and bargaining, and it has to successfully pass through several stages of negotiation and acceptance. Different expertise and interest influence the results, and they establish a 'minimum level of user oriented quality' (Dzida 1995).

ISO 9241 consists of eight parts to be considered during a software-ergonomic evaluation (the other parts are dedicated to task and hardware issues). The relevant parts of the software-ergonomic's standard are shown in Figure 1 (see Dzida 1995: 94 for the general idea of the structure).

Although common terminology has been established for the multi-party standard, no general level of concreteness concerning the definition of the standard can be found. Diverse styles have been adopted in several parts.

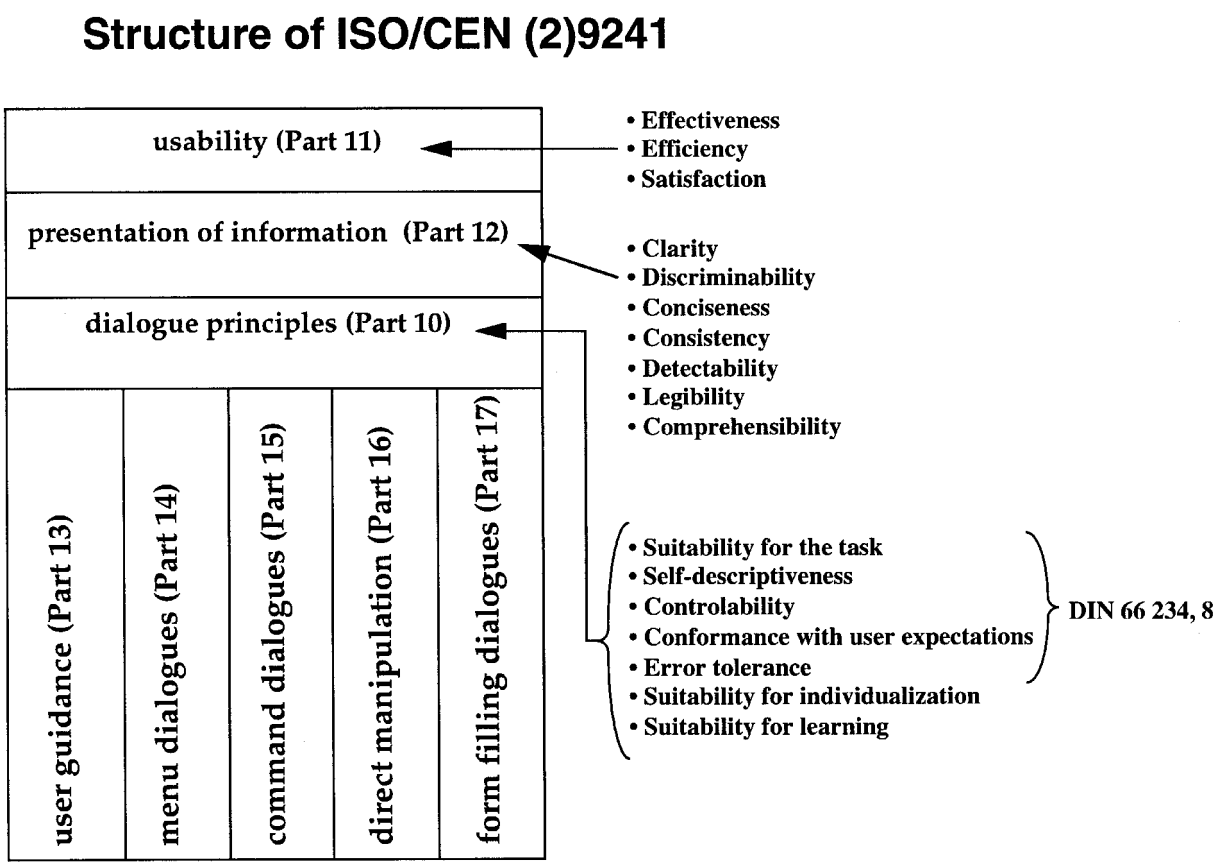

Figure 1. Structure of the multi-party standard ISO 9241.

The *concept of usability* is defined in part 11 by effectiveness, efficiency, and satisfaction of the user. Part 11 gives the following definition of usability: 'Usability is measured by the extent to which the intended goals of use of the overall system are achieved (effectiveness); the resources that have to be expended to achieve the intended goals (efficiency); and the extent to which the user finds the overall system acceptable (satisfaction).' For a more detailed discussion of the term usability see (Bevan 1995). Effectiveness, efficiency and satisfaction can be seen as quality factors of usability. To evaluate these factors, they need to be decomposed into sub-factors, and finally, into usability measures. Dzida (1995) presents a usability quality model that refines the term usability. This model introduces a stepwise operation of the factors effectiveness, efficiency and satisfaction, which ends up with specific measures called criteria[1] . Another model to refine usability factors is the linguistic decomposition approach (Bodart and Vanderdonckt 1995). For a discussion of how effective usability principles are see also Bastien and Scapin (1995).

The *dialogue requirements* are described in part 10 by the 'principles' suitability for the task, self-descriptiveness, controllability, conformance with user expectations, error tolerance, suitability for individualizatio n, and suitability for learning. Part 10 establishes a framework of ergonomic 'principles' for the dialogue techniques with high-level definitions but with only illustrative applications and examples of the principles. The principles of the dialogue represent the dynamic aspects of the interface and can be mostly regarded as the 'feel' of the interface.

The *information presentation* is described in part 12 by the 'attributes' clarity, discriminability, conciseness, consistency, detectability, legibility, and comprehensibility. The 'attributes of presented information' represent the static aspects of the interface and can be generally regarded as the 'look' of the interface. The attributes[2] are detailed in the recommendations given in the Standard (Section 4). Each of the recommendations supports one or more of the attributes stated above. The rules for presentation of information also contribute to the application of the dialogue principles, mainly to the conformity with user expectations.

Requirements for *user guidance* (prompts, feedback, status, error support and help facilities) are described in

part 13. The application of the user guidance rules also contribute to the application of the dialogue principles.

The requirements for user guidance and several dialogue techniques are described in parts 13 to 17. Each of the recommendations given in parts 13 to 17 contributes to at least one of the dialogue principles and to the attributes of information presentation, but this relationship is not explicitly stated. Part 13 to part 17 of the standard define more or less low-level and fairly exhaustive requirements for the user interface. In many cases task, user, and technical environment aspects are considered as conditions of the applicability or relative relevance of the specified requirements. These aspects constitute the context of use defined in part 11 to be considered when applying the standard to a given work system (see section 3).

3. Context of use

The software – ergonomic evaluation of usability has to be placed in the natural context of use consisting of the users, their jobs and tasks, their hardware and software, and the organizational, technical, and physical environment. Although usability is a property of the overall system, the focus of attention is usually on a specific element within the overall system — in our case, the software product. It is possible to address the usability of the user interface, but only if the particular context of use has been identified. The investigation of these elements in the context of their use is carried out by considering the following characteristics (ISO 9241 Part 11):

- *The user:* User types (e.g., user populations) based on aspects about users' skills and knowledge (e.g., software experience, hardware experience, task experience, organizational experience, education, training), personal attributes (e.g., age, sex, physical capabilities, intellectual abilities, motivation, disabilities).
- *The software:* Descriptions of the functionality and main application areas of the software, available instructional items (for example, handbooks).
- The *job and tasks:* Details about the job of the user, and the tasks for which the software will be used as an aid (for example, task goal, task frequency, task breakdown, task duration, task flexibility, task output, task dependencies).
- *Organizational environment:* Aspects of the structure of the organization (e.g., hours of work, group working, job function, work practices, management structure, communication structure, interruptions), the attitudes and culture (for example, policies on computer use, organizational aims, industrial relations), and the job design (for example, job flexibility, performance monitoring, performance feedback, pacing, autonomy, discretion).
- *Technical environment:* Hardware and basic software (for example, the operating system) which is necessary to use the software, reference material.
- *Physical environment:* Workplace conditions (e.g., humidity, temperature, noise), design of the workplace (e.g., space and furniture, user posture, location), workplace safety (e.g., health hazards, protective clothing and equipment).

4. ISO 9241 evaluator – an evaluation procedure

4.1. *Aim and structure of the evaluation method*

The ISO 9241 evaluator supports the conformance test of a given application with ISO 9241, parts 10 to 17. The evaluation procedure is based on EVADIS (Oppermann *et al.* 1988) and EVADIS II (Oppermann *et al.* 1992), an approach being designed to support the expost evaluation of a user interface that is now open to being used for evaluations of interfaces under development (mock-ups, prototypes etc.).

The ISO 9241 evaluator is a guideline oriented expert-based evaluation method that prepares the requirements of the multi-party standard ISO 9241 to be tested in about 300 test items. The test items are structured in a two dimensional space defined by technical components and software-ergonomic criteria. The dimension of the technical components is inspired by an IFIP model for user interfaces (Dzida 1983, 1988), extended and adapted to the structure of the multi-party ISO standard. The second dimension consists of the software – ergonomic principles — based on the dialogue principles of ISO 9241, part 10 and extended by requirements for the included I/O interface of a system. Each test item checks a particular aspect of ergonomic requirements specific for the given component and criteria. The two dimensional space to structure the test items is shown in Figure 2.

Besides structuring the requirements of the multi-party standard, the contribution of the ISO 9241 evaluator can be seen in the software support for the evaluation. The support contains an editing facility of technical components, software ergonomic criteria and test items. Due to the ongoing development of research and development in user interfaces, as well as in the standards, there is a need for maintenance of the instruments. In particular, the support contains the evaluation process of a given application and the

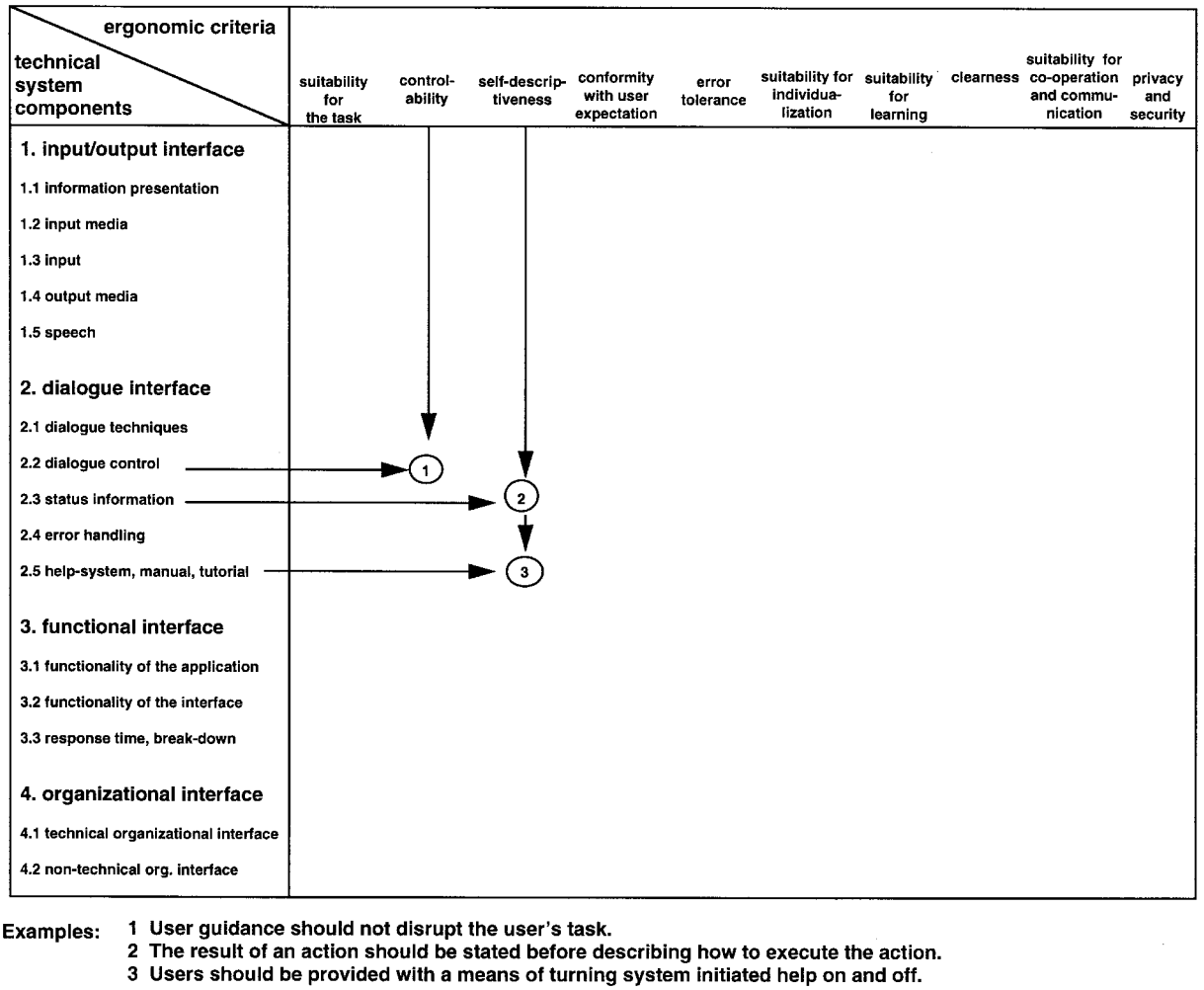

Figure 2. Two dimensional space to structure the test items.

writing of an evaluation report. To facilitate the application of the multi-party standard, explanations are provided to be used in understanding its requirements for the completion of conformance testing. These features characterize the ISO 9241 evaluator in comparison from the Annex A of parts 12 to 17. In the Annex a general procedure for assessing applicability and adherence is described. Several possible methods for applicability and adherence are defined and a checklist for the results of the testing is presented. With the ISO 9241 evaluator, the support for the testing, the documentation of the testing, the evaluation, and the report of the results is provided.

The ISO 9241 evaluator is a subjective evaluation method because the expert examines and answers questions according to his personal assessment. However, it is also objective because the ergonomic requirements are operationalized and precisely formulated, thus enabling the evaluator to answer questions based on clear test rules and traceable conditions (see 4.4.3). The evaluator is a human factors expert using the methods to evaluate the conformance with ISO 9241. The expert approach is based less on a task to be performed by the targeted system than on questions asked by software ergonomics. Advantages of an expert based evaluation method are: relatively fast, uses few resources, provides an integrated view, and can be addressed to a wide range of behaviour. A sample test item is shown in Figure 3.

Detailed test instructions in the EVADIS evaluation guide help to reduce the subjectivity of this method. The test instruction gives the evaluator useful information about how to test the specific attribute of the software product systematically. The comment contains desirable ergonomic requirements. The evaluator can use this information during the rating process. The answers to the test item in the particular test situation are made by ticking off one or more of the check boxes representing

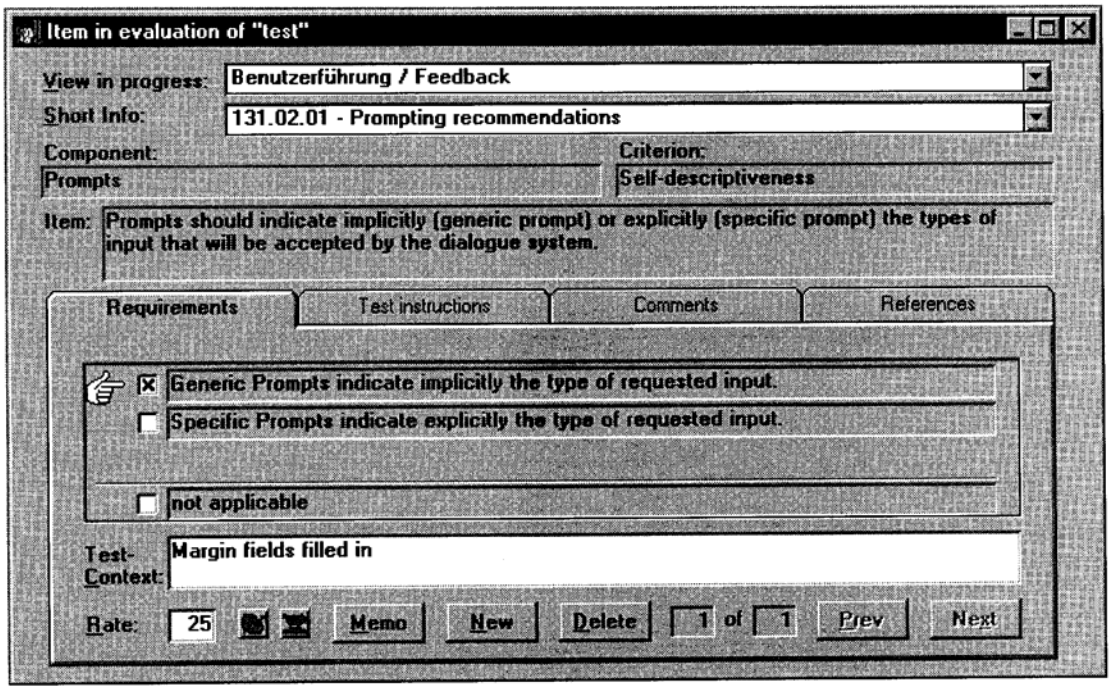

Figure 3. Sample of test item for the evaluation.

several aspects of the test item. All answer options represent the complete ergonomic requirements of this test item, as they have been defined in the corresponding part of the ISO 9241. Test items have to be answered in particular test situations, typically more than once. Each test situation will be shortly characterized by the evaluator (e.g., menu structure of the main menu; menu structure of the customer window). With the help of integrated capture tools (Lotus ScreenCam and Microsoft Paintbrush), all on-screen behaviour and window sequences, together with verbal and textual comments of the evaluator can be captured. The logged examples can be used by the evaluator to explain detected deficiencies. This is very helpful for communication with the software designers, because the evaluator can explain the deficiencies in real examples.

For each test situation, a new rating of the achievement of the ISO 9241 requirements will be done automatically by the software. Answering a test item in different test situations is necessary for a comprehensive evaluation of a software product. Different answers in different situations can be an indication of a context sensitive adaptation of the interface to different contexts of use, but they can also be an indication of an inconsistent user interface. The evaluator's description of the test situations allows him to distinguish between appropriate adaptation or inconsistency in the interpretative phase of the results, and it allows for reconstruction of the given answers in a redesign discussion with the designer at a later time. The definition and documentation of test situations allow the evaluation procedure to be used not only for summative but also for formative evaluation purposes. It can be embedded in the software engineering process at different stages (see Reiterer and Oppermann 1993). Hix (1995) discusses in greater detail the incorporation of usability evaluation techniques into the software lifecycle.

The final result of the evaluation process will be an evaluation report. The preparation of the report is supported by the software. All results of the evaluation process (for example, all comments or selected answer options of a test item) are transferred into a predefined Microsoft Word document. With the help of the text editor the final report can be written. As a second tool to support the interpretation of the results, we have integrated the spreadsheet program Microsoft Excel. With the help of this tool, the achievement of the different ISO 9241 requirements can be shown in a table or in a graphical representation.

Software evaluation is only possible in a particular context of use. Therefore, a simplified workplace analysis and a questionnaire exploring user characteristics based on the definition of the context of use in ISO 9241 part 11

are prepared for the evaluation. The gathering of the context of use is supported by software with the help of a questionnaire. The context of use allows the evaluation to be focused on the environment where the given software will be used (Figure 4).

4.2. *Defining particular evaluation views*

The evaluation can be aimed at the evaluator's or designer's particular interests by what we call 'views'. With the help of views, the evaluator can define a subset of test items that is relevant for the evaluation. Answering all 300 test items is not necessary for most of the evaluation aims. The application to be evaluated does not make use of all dialogue techniques. Only a subset of ISO parts 14 to 17 are relevant. The I/O-devices of the given hardware in the worksystem do not allow for all types of interactions, thus the evaluator can select only the actual ones. In particular developmental phases, the designer is sometimes only interested in the evaluation of one component, let's say, it's the menu design. The evaluator could select the menu design from the list of prepared views. The selection of predefined views is shown in Figure 5.

The designer can also define individual views. This is supported by a view editor where the evaluator can define views like, let's say, error management. The view editor is shown in Figure 6.

The views (in particular self defined ones) allow the specialist to create an efficient environment for repeated occasions of evaluation in composing a special repository of test items via selected views.

4.3. *Evaluation software* [3]

The software for the ISO 9241 evaluator runs on a PC under Windows. All components of the evaluation method (for example, technical components, software ergonomic criteria, and test items) are stored in a relational database. The software package supports the evaluator during the whole evaluation process and provides an assessment summary. As described above the evaluator specifies and records the test situation(s) for each test item, evaluates each test item for the (several) test situation(s), writes an explanation of his or her evaluation, and can capture detected deficiencies. Test results in the form of ticked check boxes, verbal comments and defined ratings can be exported to a text editor and a spreadsheet to produce the final evaluation report.

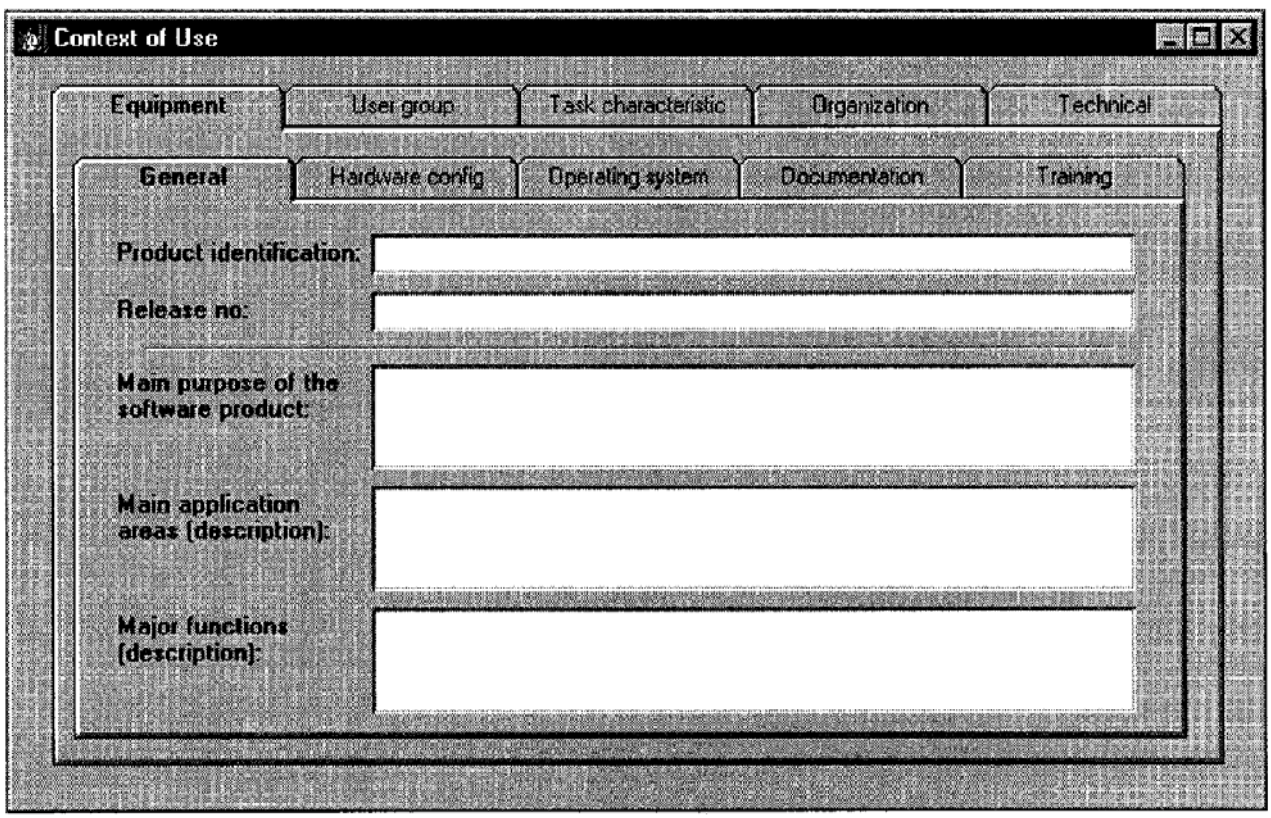

Figure 4. Specification of the context of use.

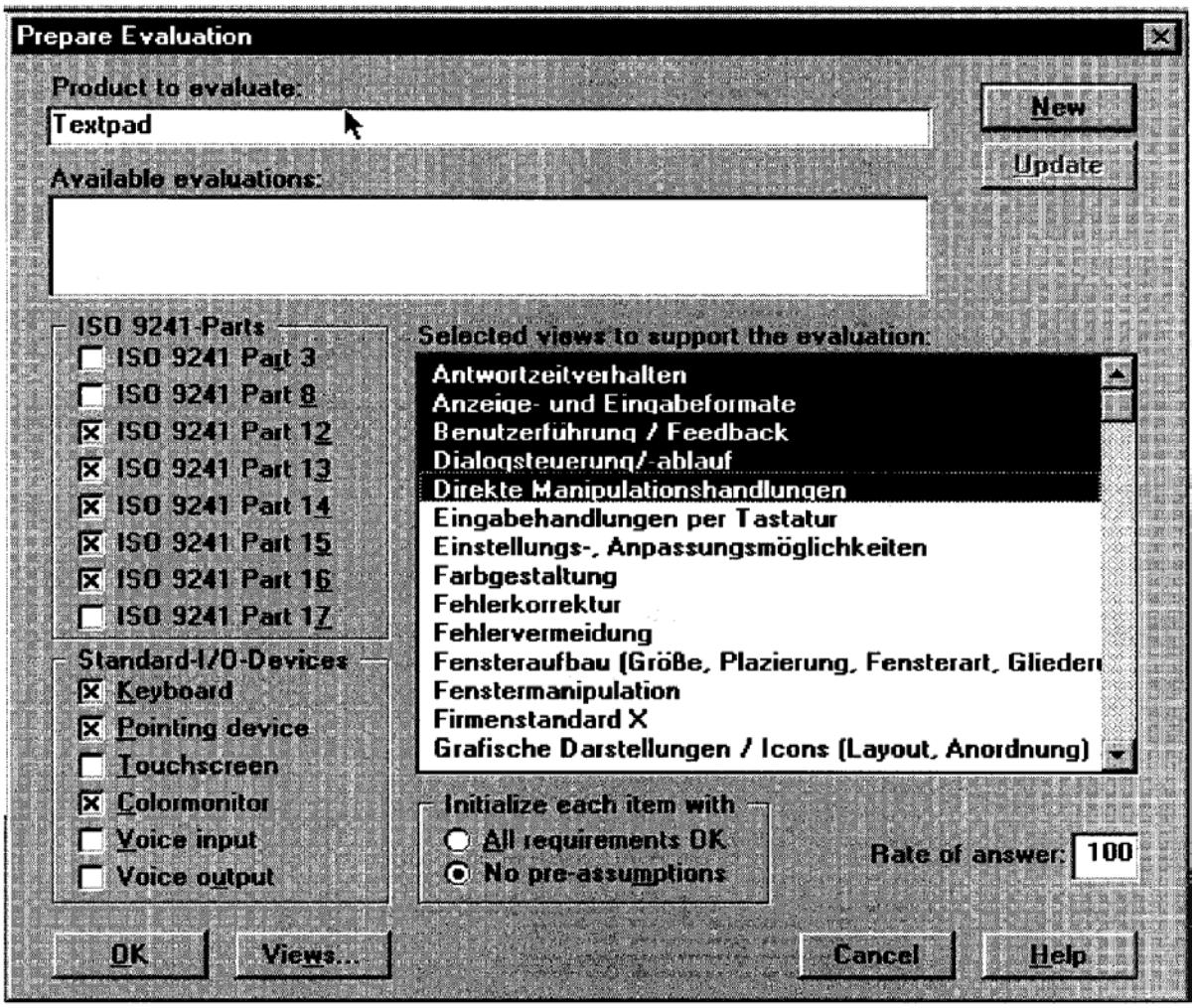

Figure 5. Selection of predefined views.

4.4. *Comparison of ISO 9241 evaluator with other evaluation methods*

The aim of this section is to show the state of the art of currently available evaluation methods, and then to compare these methods with the ISO 9241 evaluator.

4.4.1. *Subjective evaluation methods:* Subjective evaluation methods — also known as survey methods (Macleod 1992) — are directly based on the user's judgement. The methods give the evaluator access to the users' 'subjective views of the user interface of a system'. They involve real users, but information is not gathered while users interact with the system (as is the case through objective evaluation methods).

4.4.1.1. *Questionnaires:* Questionnaires offer the advantage of providing data about the end-users' views of user interface quality, as opposed to an expert or theoretician's view, and can provide data about system usability in a real work setting. It is essential to ensure that the group of people responding to the questionnaire match the actual or intended group of end-users, and that they have used the software for work purposes, performing the specific work tasks in which the evaluator is interested.

Examples of subjective evaluation methods based on questionnaires that can be practically applied are;

- the 'Questionnaire for User Interaction Satisfaction (QUIS 5.0)' developed by Norman and Shneiderman (1989),

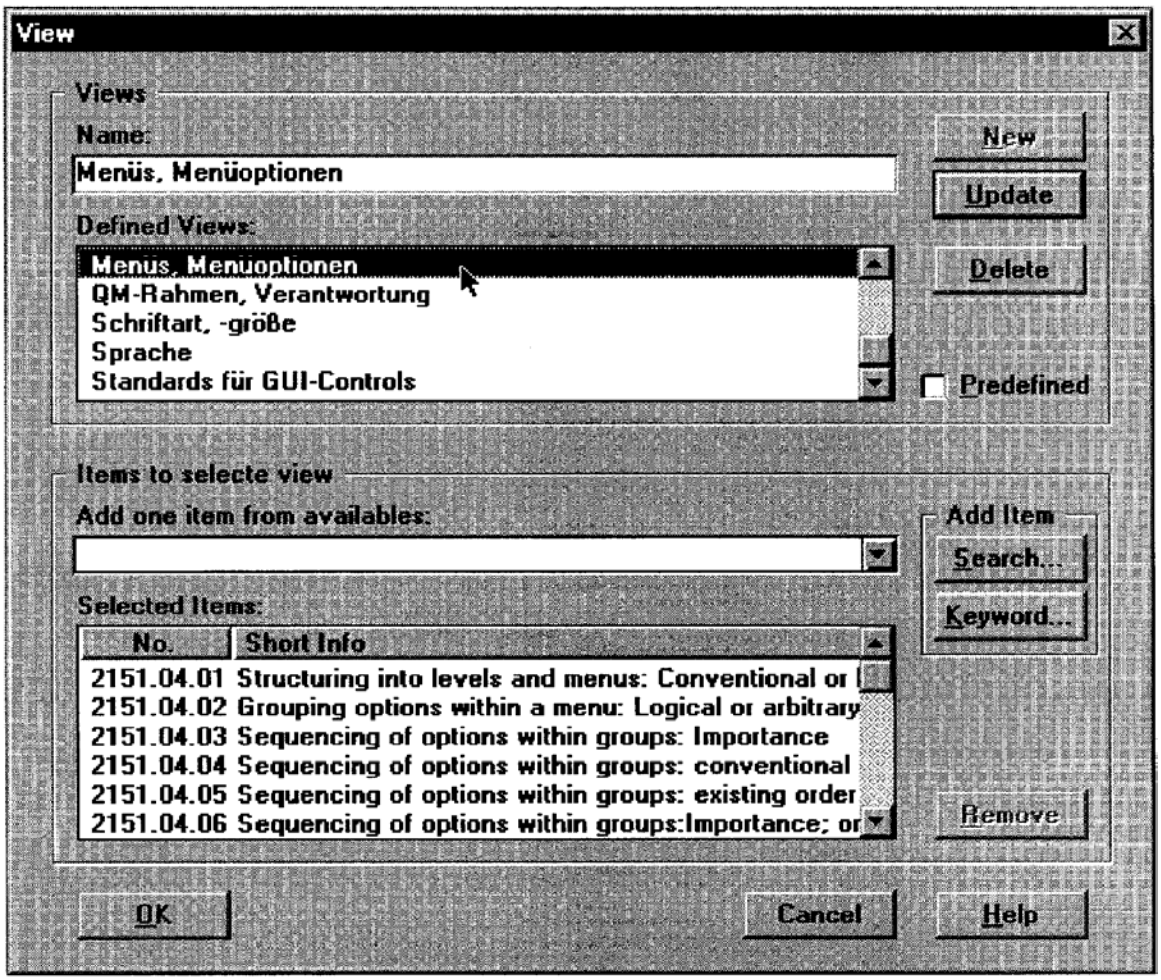

Figure 6. View editor to define own subset(s) of test items.

- the 'Software Usability Measurement Inventory (SUMI)' developed as part of the CEC ESPRIT Programme, Project 5429, Metrics for Usability Standards in Computing (MUSiC) (MUSiC 1992),
- the 'ISONORM 9241 Questionnaire' developed by Prümper (1993).

4.4.1.2. *Interviews:* Interviews are much more time-consuming than questionnaires. They need careful pre-planning, and a good degree of expertise on the part of the interviewer. Since the interviewer can adjust the interview to the situation, interviews are well suited to exploratory studies in which the evaluator does not yet know in detail what he is looking for. Structured interviews with a pre-determined set of precisely phrased questions are required if data are to be analysed statistically. Unstructured or flexible interviews covering pre-specified topics but in a style and order shaped by the responses of the user can elicit more revealing information about the usability of a system. The results are more difficult to analyse, but can be highly informative.

4.4.2. *Objective evaluation methods:* Objective methods — also known as observational methods (Macleod 1992) — involve real people using working systems. They are based on observing users interacting with the system and can range from being almost entirely informal to highly structured. Often, the interaction of the users is recorded in some way for later observation or analysis. To be helpful in evaluating the usability of the system, even the most informal approach requires that the users perform work tasks with the system. Many software-producers run a usability lab in which representatives of real users perform test tasks under the observation of

experts. Observation or video records are often used to confront developers with usability problems of users. This approach seems to be more effective than a (scientific) report generated by a human factors expert (Wiklund 1994).

In all objective evaluation methods, user performance will be measured by having a group of test users performing a predefined set of test tasks while collecting time and data. For this purpose, the general goal 'usability' has to be broken down in different 'usability attributes', which can be measured.

4.4.2.1. *Direct observation:* The simplest form of observation involves having a person familiar with usability issues (for example, a human factors expert) observe individual users performing tasks with the system. If the system is fully implemented and in its natural work setting, then direct observation tells much about the use and usability of the system. Direct observation is an important component of more sophisticated observational methods. For example in a usability laboratory, where observers can view the test person through a two way mirror from an adjacent room.

4.4.2.2. *Video recording:* Video recording allows data to be separated from analysis. The evaluator is free from analysing the data during the observation. Any aspect that has been recorded can be analysed by the evaluator in detail after the session. Most video recording and analysis take place in usability laboratories. By using several cameras, it becomes possible to capture and synchronize data on screen display, hand movements and other user behaviour. Representative users perform specific work tasks with implemented systems or prototypes. A major cost of such evaluation work is the time required to analyse the data. For this purpose, different support tools are available that enable much more rapid analysis of video data.

An example of video recording based observation approach including analytic support tools is the 'Diagnostic Recorder for Usability Measurement (DRUM)' developed as part of the CEC ESPRIT Programme, Project 5429, Metrics for Usability Standards in Computing (MUSiC) (MUSiC 1992).

4.4.2.3. *Interaction monitoring:* The idea is to automatically gather data about how people interact with the system. With the help of monitoring facilities all user inputs (e.g., commands, mouse clicks, menu options) and system outputs (e.g., information displayed, changed states) can be recorded. There are different possibilities for monitoring the interaction:

- Replay based: With the help of a capture tool (for example Lotus ScreenCam), all on-screen behaviour and window sequences together with verbal interaction and users' comments can be captured. Complete transcripts of user sessions are logged either for use in later playback or for analysis of patterns of use, such as what commands are issued next after an error situation (Nielsen 1993). This is a very easy way to monitor the interaction, but it tends to swamp the analyst with raw data.
- Logging based: Logging is usually achieved either by logging low-level parts of the system software, such as keyboard and mouse drivers, or by modifying the software of interest (Nielsen 1993). The latter approach is much preferred, since it makes it easier to log events of interest. The logged information will be analysed by the evaluator, making statistics about the use of special commands (e.g., use of help action, number/placement of mouse clicks, number of error messages during a session).

4.4.2.4. *Co-operative evaluation:* Simple observation lacks access to the thoughts of the user and to the reasons why users did certain things, or failed to do them. Co-operative evaluations take a major step beyond the simple observational methods, because they do not rely solely on observing real users' interaction with working systems; they actively involve those users in the process of evaluation.

In the most simple form, the observer (evaluator) asks questions of the user during performance of a task, for example when the user is encountering a problem, but not providing a comment. The problem with this approach is the interruption of the user's task.

An alternative is to ask people retrospectively what they have done, avoiding interfering with the way people work. Here the difficulty may be in recalling the important problems. A video confrontation of the test users performing tasks can help people describing important problems and reasons for actions. This method is sometimes called 'retrospective testing'.

In addition, the 'thinking aloud' method should be mentioned here. In this method the users perform a task while giving verbal expression to their thoughts, problems, opinions, etc., all of which provides the evaluator with indicators for interpreting the test. This approach may seem artificial to the user, and some test users have great difficulties in keeping up a steady stream of comments as they use a system.

An alternative is the 'constructive interaction' method, in which two users work together on a task and 'tell' each other what they are feeling, doing, or intending to do, etc. The conversation generates data in a more 'natural' manner and users will make more comments when engaged in constructive interaction than when simply thinking aloud.

4.4.3. *Expert evaluation methods:* Expert evaluation methods draw upon expert knowledge to make judgements about the usability of the system for specific end-users and tasks. The expert may be a human factors expert but it could also be a designer with some basic knowledge of human factors. These methods lie at an intermediate stage between subjective evaluation methods and objective ones. These methods are subjective since the expert examines and answers questions pertaining to software ergonomics according to his personal assessment. They are objective since the examination criteria of software ergonomics are operationalized and precisely formulated to an extent which enables the evaluator to answer questions based on clear test rules and traceable conditions (for relative advantages see Whitefield, *et al.* 1991). Detailed instructions in the evaluation guide (e.g., detailed process description, clear notation, structure of the statement) can help reduce the subjectivity of these methods. Hammond *et al.* (1985) report a comparison between expert judgement and user observation and show the superiority of the expert judgement.

The usability inspection methods defined by Nielsen and Mack (1994) can be seen as special kinds of expert evaluation methods. They define usability inspection as the evaluation of the user interface based on the considered judgement of the inspector(s). Usability inspectors can be usability experts, software designers with special expertise, or end users with content or task knowledge. The defining characteristic of usability inspection is the reliance on judgement as a source of evaluative feedback on specific elements of a user interface.

4.4.3.1. *Specialist reports and expert walkthrough:* 'Specialist reports' represent a long established, loosely defined way of evaluating the usability of a system. A human factors expert provides critical evaluation based upon expert knowledge. For their validity, specialist reports rely heavily upon the quality of the expertise, and the skill of the expert in imagining how the system will match the abilities and preferences of the intended end-users.

The 'expert walkthrough' is a variation of the specialist report, but is more methodological. The critical evaluation is generated by the human factors expert on the basis of 'walking through' a number of tasks, which should be representative of the tasks the system is designed to support. However, expert walkthrough relies upon the talent of the expert in anticipating which things the user will find easy or difficult in using the system.

4.4.3.2. *Cognitive walkthrough:* 'Cognitive walkthrough' is a method which gives expert evaluation of usability a more rigorous theoretical basis. In cognitive walkthroughs, a proposed interface in the context of one or more specific user tasks is evaluated. The input to a walkthrough session includes an interface's detailed design description (paper mock-up or working prototype), a task scenario, explicit assumptions about the user population and the context of use, and a sequence of actions that a user should successfully perform to complete the designated task. The cognitive walkthrough method is based on a theory of learning by exploration and on modern research in problem solving (see Wharton *et al.* 1994).

4.4.3.3. *Checklists, guidelines and principles (heuristics):* 'Checklists' are a very popular form of evaluational method, because they can be applied in an easy way. Checklists are composed of a clearly laid out list that can be worked through point by point, to produce a simple but thorough evaluation. However, checklists do require some degree of expert knowledge, both to answer questions and to interpret results.

Checklists are often based on guidelines or principles, as they test how well a user interface of a system complies to them. Today many guidelines are available in different style guides, e.g., Apple Human Interface Guidelines (Apple 1992), IBM Common User Interface Guidelines (IBM 1991), OSF Motif Style Guide (OSF Motif 1992), and Microsoft User Interface Style Guide (Microsoft 1995). Guidelines tend to be system or platform specific. They are useful when an evaluator wants to test the conformance of a user interface with platform specific 'look and feel'.

Usability principles are more general statements about things that affect usability. The most popular ones are the dialogue principles of the ISO 9241 part 10: suitability for the task, self-descriptiveness, controllability, conformity with user expectations, error tolerance, suitability for individualization, suitability for learning. They are not tied to specific systems, but require much interpretation to apply. However, they can provide a useful framework for structuring the questions of a checklist.

An interesting variation of the principle-based evaluation is the 'heuristic evaluation' (Nielsen 1994). Nielsen has derived 10 usability heuristics (principles) from a factor analysis of 249 usability problems. These heuristics can be used by a small set of evaluators (typically three to five) examining a user interface and judging its compliance with them. The evaluators should be usability specialists, preferably persons with a strong background in human factors and with good domain knowledge.

An important measure for each checklist-based evaluation method is the extent to which it is imbedded in a test scheme, (i.e., in a test specification) for the performance of an evaluation. Many of them allow the evaluator to choose how the system being tested should be used to obtain answers to test questions. The disadvantage of this approach is that it is hard to follow the method in which the evaluation results have been

obtained. Along with the test questions checklist-based methods also specify a detailed evaluation procedure, describing the different steps of the evaluation process (e.g., analysing the context of use, building representative test tasks). The method 'ISO 9241 evaluator' described in this article is an example of a checklist-based evaluation approach, which relies on principles.

Examples of guidelines or principles based on checklist evaluation methods that can be practically applied are;

- the 'Evaluation Checklist' developed by Ravden and Johnson (1989),
- the 'EVADIS Checklist' and 'ISO 9241 Evaluator' developed by Oppermann *et al.* (1992), and Reiterer and Oppermann (1995),
- the 'heuristic evaluation' developed by Nielsen (1994).

4.4.4. *Experimental evaluation methods:* In asking scientific questions, soundly designed empirical experiments are necessary in the testing of hypotheses about the usability effects of various factors in a design. Empirical experiments are normally conducted as controlled experimental studies. One problem involved in planning experiments is the correct definition of dependent and independent variables; a second problem is the selection of the proper environment for the study. A third problem is the lack of any underlying theory dealing with man-machine interaction, so that the features to be considered are mostly left to the researcher's imagination and preferences. Monk (1985) provides a thoughtful assessment of the value of controlled experiments and an introduction to key issues in designing experiments specifically for evaluation.

4.4.5. *Summary of important characteristics of evaluation methods:* Table 1 shows that each evaluation approach has its advantages and disadvantages. The choice of a method for a specific evaluation depends upon the stage of system development, the kind of user involvement, the type of data necessary for the kind of results required, the available expertise, the possible place of the evaluation and the available resources (time and money).

The characteristics of each method shown in Table 1 can be used to compare the ISO 9241 evaluator with other evaluation methods. The ISO 9241 evaluator is an expert-based evaluation method used as a means of assuring standard conformance. It incorporates a checklist and considers general usability principles derived from ISO 9241 part 10.

The ISO 9241 evaluator requires actual software in its use, so that the *timing of the evaluation* in the development process can occur after designing a prototype and analysing the tasks and the user characteristics. Therefore, the ISO 9241 evaluator cannot be used during the specification stage of the product development. The ISO 9241 evaluator can typically be used for ex-post evaluation purposes, like evaluating standard software products for purchasing decisions, while approaches like the expert and the cognitive walkthrough are better suited for evaluation purposes in the early stages of the software development process.

As an expert based evaluation approach, the ISO 9241 evaluator does not need real users during the evaluation process. However, the evaluator must have access to real users before the evaluation, in order to gather typical user and tasks' characteristics in defining the context of use. *User-based methods* like questionnaires are better suited to show the user's satisfaction with the interface of the software. Compared with the ISO 9241 evaluator, observation methods (i.e., direct, video recording) have the advantage of their judgement being based on objective data. In practice, each expert-based approach should be supplemented with a user-based approach.

The primary *scope of evaluation* of the ISO 9241 evaluator is on the user interface of the software. Therefore, the type and number of problems one can detect are mostly related to software usability and not to the quality of work or to the user's behaviour. To evaluate the quality of work, special task and workplace analysis methods have been developed (for example, KABA, see Dunckel *et al.* 1993). To evaluate the user's behaviour, observational or experimental methods have to be used.

The *types of data* that can be gathered with the ISO 9241 evaluator are primarily qualitative and diagnostic. If quantitative data are needed, observational or experimental evaluations are better suited.

An expert with good *expertise* about human factors is required for the use of the ISO 9241 evaluator. It is not a method that can be used by those without a background in human factors. If such expertise is not available other methods as questionnaires or heuristic evaluations are better suited.

The evaluation with the ISO 9241 evaluator can take *place* in house (at the users' workplace) or at the expert's workplace. There is no need for complex or expensive equipment. The primary tool for the evaluation is a PC running the ISO 9241 evaluator software. The flexibility in location and the equipment for the evaluation are important advantages of this approach compared with other evaluation approaches.

The ISO 9241 evaluator supports expert judgement, so the *costs* imposed by the evaluation can be restricted. The available computer support reduces routine work. Information about the tasks and the user characteristics is needed for the context of use. If this is not specified (for

Table 1. Important characteristics of evaluation methods (based on Macleod 1992).

Type of Method	Subjective	Objective	Expert	Experimental
Timing of use in product development	prototype or later	prototype or later	any stage	any stage
User-based	yes	yes	no	yes
Scope of evaluation	broad	broad	broad	very narrow
Type of data:				
• quantitative	yes	yes	no	yes
• qualitative	yes	yes	yes	yes
• diagnostic	yes	yes	yes	narrowly
Expertise required	medium	medium	high	high
Place	in-house	in-house or laboratory	in-house or at the expert's workplace	laboratory
Costs:				
• time	low/medium	medium/high	low	high
• money	low/medium	medium/high	low	high
Main advantages	Fairly quick User-based User's view Diagnostic Broad scope	User-based and involves user performance on work tasks Diagnostic Broad scope Can bring together designers and users	Low cost Quick Diagnostic Broad scope	User-based Rigorous Can produce results with validity and reliability
Main disadvantages	Less effective early in the design cycle Only retrospective views of usability	Less effective early in the design cycle May interfere with what is being observed	Not-user based Depends on quality of the expert Questionable reliability	Narrow scope High cost Requires careful design by competent theoretician
Main disadvantages	Less effective early in the design cycle	Less effective early in the design cycle	Not-user based Questionable reliability	Narrow scope High cost

example during the analysis process of the software development), the evaluation could be very time consuming because the evaluation has to take into account all possible contexts of use.

The *main advantages* of the ISO 9241 evaluator are its flexibility, modest equipment requirement, low costs, and that it is a method that can be fairly quickly used especially when the evaluator has good experience in applying the method and the necessary information about users and task are available in documented form.

The *main disadvantages* of the ISO 9241 evaluator are that the judgement is not user-based and the open reliability. Variations in assessment between different evaluators are reduced by a detailed evaluation guide, which describes the whole evaluation process. Nevertheless, the final statement can be biased to a certain degree by the judgement of the expert concerning the relevance and rating of the evaluation items. Empirical tests are in progress to show the validity and the reliability of the ISO 9241 evaluator.

5. Summary

Many different evaluation methods are currently available. The choice of a method for a specific evaluation depends upon the stage of system development, the kind of user involvement, the type of data necessary for the kind of results required, the available expertise, the possible place of the evaluation and the available resources (time and money). There is an increasing need for practical and comprehensive evaluation methods and tools for conformance testing with ISO standards. Practical means that the amount of time and resources must be manageable in software projects. Comprehensive means that the context of use has to be considered during the evaluation of user interfaces. For a comprehensive evaluation of usability, it is necessary to use more than one evaluation method. For example, an expert method for diagnosis of general usability principles in a user interface, combined with the use of a subjective method to evaluate the user interface to show the users' level of satisfaction. Another example may be

an expert method for quick diagnosis of specific usability problems in a prototype, combined with the use of an objective method to measure the redefined prototype to evaluate the user performance with the help of representative test tasks.

The ISO 9241 evaluator presented in this article is a practical expert-based evaluation method that can be integrated into a comprehensive evaluation approach. In particular, it takes the context of use into consideration and provides extensive computer support for the use of the evaluation procedure. To reduce the variations in assessment between different evaluators, detailed instructions have been developed that supports the evaluator during the evaluation process.

Acknowledgements

Special thanks to Thomas Geis who has given us many useful hints about the actual status of the different parts of the ISO 9241 standard. We would also like to thank the partners of the EVADIS team for their contribution to the development of the ISO 9241 evaluator: Thomas Geis (TÜV Rheinland, Cologne), Manfred Koch (Ernst & Young GmbH Vienna), Jochen Prümper (Prof. Prümper & Partner, Munich/Berlin), Wilhelm-Wolfgang Strapetz (University of Vienna), and Christoph Wick (GMD, St. Augustin).

Notes

[1]Note that in Dzida's notation the term criterion is used differently from our terminology: we perceive a criterion as an abstraction of different measures for an ergonomic quality, e.g., flexibility, what Dzida calls sub-factor, where he uses a criterion for the measurable operationaliza tion of a sub-factor.

[2]In former versions the attributes were called principles for the presentation of information. Both terms are correct. It is a matter of the view of the reader. The designer's view is the principles-view, as he is looking for guidance in the design process. The evaluator's view is the attributes-view, as he is looking for attributes of a product to be evaluated

[3]The software was developed by Wilhelm–Wolfgang Strapetz, Ernst & Young Consulting GmbH Vienna and University of Vienna.

References

Bodart, F. and Vanderdonckt, J. 1995, Using recommendations and data base tools for evaluation by linguistic ergonomic criteria, in Y. Anzai, K. Ogawa, and H. Mori (eds) *Symbiosis of Human and Artifact*. (Elsevier Science B.V., Amsterdam), 367 – 372.

EEC, 1990, The minimum safety and health requirements for work with display screen equipment. Directive of the European Economic Community, 90/270/EEC.

Hammond, N., Hinton, G., Barnard, P., MacLean., Long, J. and Whitefield, A. 1985, Evaluating the interface of a document processor: a comparison of expert judgement and user observation, in B. Shackel (ed). *Human – Computer Interaction—INTERACT '84*, (Elsevier Science Publishers B.V., Chichester), 725 – 729.

Hix, D. 1995, Usability evaluation: how does it relate to software engineering? in: Y. Anzai, K. Ogawa, and H. Mori (eds) *Symbiosis of Human and Artifact*, (Elsevier Science B.V., Amsterdam), 355 – 360.

IBM, 1991, Systems Application Architecture, Common User Access, Advanced Interface Design Reference.

ISO 9241, 1994, Ergonomic Requirements for Office Work with Visual Display Terminals, Part 8, Requirements for displayed colours, Draft International Standard.

ISO 9241, 1994, Ergonomic Requirements for Office Work with Visual Display Terminals, Part 10, Dialogue Principles, International Standard.

ISO 9241, 1995, Ergonomic Requirements for Office Work with Visual Display Terminals, Part 11, Guidance on specifying and measuring usability, Draft International Standard.

ISO 9241, 1996, Ergonomic Requirements for Office Work with Visual Display Terminals, Part 12, Ergonomic requirements for presentation of information, Committee Draft.

ISO 9241, 1995, Ergonomic Requirements for Office Work with Visual Display Terminals, Part 13, User guidance, Draft International Standard.

ISO 9241, 1996, Ergonomic Requirements for Office Work with Visual Display Terminals, Part 14, Menu dialogues, International Standard.

ISO 9241, 1994, Ergonomic Requirements for Office Work with Visual Display Terminals, Part 15, Command dialogues, Proposed Draft International Standard.

ISO 9241, 1996, Ergonomic Requirements for Office Work with Visual Display Terminals, Part 16, Direct manipulation dialogues, Committee Draft.

ISO 9241, 1994, Ergonomic Requirements for Office Work with Visual Display Terminals, Part 17, Form filling dialogues, Committee Draft.

Macleod, M. 1992, *An Introduction to Usability Evaluation*, (National Physical Laboratory, Teddington).

Microsoft, 1995, *The Windows Interface Guidelines for Software Design*, (Microsoft Press).

Monk, A. 1985, How and when to collect behavioural data, in: A. Monk (ed) *Fundamentals of Human – Computer Interaction*, (Academic Press, London), 69 – 79.

MUSiC, 1992, *Metrics for Usability Standards in Computing (ESPRIT II Project 5429)*, Product information, National Physical Laboratory, UK.

Nielsen, J. and Mack, R. (eds), 1994, *Usability Inspection Methods*, (John Wiley & Sons, New York).

Nielsen, J. 1993, *Usability Engineering*, (Academic Press, San Diego).

Nielsen, J. 1994, Heuristic Evaluation, in: J. Nielsen and R. Mack (eds) *Usability Methods*, (John Wiley & Sons) 25 – 62.

Norman, K. and Shneiderman, B. 1989, *Questionnaire for User Interaction Satisfaction (QUIS 5.0)*, University of Maryland: HCI-Lab, College Park.

OPPERMANN, R., MURCHNER, B., PAETAU, M., PIEPER, M., SIMM, H. and STELLMACHER, I. 1988, *Evaluation von Dialogsystemen, Der Software-Eronomische Leitfaden EVADIS,* (Walter de Gruyter, Berlin).

OPPERMANN, R., MURCHNER, B., REITERER, H. and KOCH, M. 1992, *Software-ergonomische Evaluation, Der Leitfaden EVADIS II*, (Walter de Gruyter, Berlin).

OSF MOTIF, 1993, *Open Software Foundation, OST/MOTIF Style Guide*, Revision 1.2, (Prentice Hall, London).

PRÜMPER, J. 1993, Software-evaluation based upon ISO 9241 Part 10, in T. Grechenig and M. Tscheligi (eds) *Human Computer Interaction, Vienna Conference, VCHCI'93, Fin de Siècle Vienna, Austria, September 1993, Proceedings, Lecture Notes in Computer Science*, (Springer Verlag, Wien), 255 – 265.

RAVDEN, S. and JOHNSON, G. 1989, Evaluating usability of human – computer interfaces, a practical method. (Ellis Horwood, John Wiley & Sons, New York).

REITERER, H. and OPPERMANN, R. 1993, Evaluation of user interfaces, EVADIS II – a comprehensive evaluation approach, *Behaviour & Information Technology*, **12,** 137 – 148.

REITERER, H. and OPPERMANN, R. 1995, Standards and software-ergonomic evaluation, in Y. Anzai, K. Ogawa, and H. Mori (eds) *Symbiosis of Human and Artefact: Human and Social Aspects of Human – Computer Interaction*, (Elsevier Science B.V., Amsterdam), 361 – 366.

WHARTON, C., RIEMAN, J., LEWIS, C. and POLSON, P. 1994, The Cognitive Walkthrough Method: A Practitioner's Guide, in J. Nielsen and R. Mack (eds) *Usability Inspection Methods*, (John Wiley & Sons, New York), 105 – 140.

WHITEFIELD, A., WILSON, F. and DOWELL, J. 1991. A framework for human factors evaluation, *Behaviour & Information Technology*, **1,** 65 – 79.

WIKLUND, M.E. 1994, *Usability in Practice, How Companies Develop User-Friendly Products*, (AP Professional, Boston).