Breaking down Usability

Martijn van Welie, Gerrit C. van der Veer, Anton Eliëns

Faculty of Computer Science, Vrije Universiteit Amsterdam
de Boelelaan 1081a, 1081 HV
Amsterdam
The Netherlands
{martijn,gerrit,eliens}@cs.vu.nl

ABSTRACT Good usability of a system is the main goal of interface designers. Determining the usability is usually done afterwards by performing usability tests with users or by going through checklists. On the other hand, design guidelines and design heuristics give the designer assistance in improving the usability while designing. In practice the available checklists, tests, guidelines etc. differ in terms of structure, content and terminology and the suggestion is given that one list is more useful than another. This paper breaks down the concept of usability into a layered model that allows comparisons of the available lists and provides better understanding of them. The model also gives a framework for evaluation by showing which usability aspects can be tested empirically and what can be formally checked by analyzing designs during the design process itself.

KEYWORDS usability, evaluation

1. INTRODUCTION

The term usability is used to denote that a design is "good" from a HCI point of view (Hartson 1998). Does a design provide the right functionality in the right way and does it satisfy everyone's needs? A designer or a design team can use guidelines, heuristics or rules as aids in the design process to ensure good usability. On the other hand designers should evaluate their design with users in practice to see if the usability is at the desired or required level. For the evaluation with users checklists or sets of ergonomic criteria and heuristics exist. The problem of all these lists, rules and criteria is that it is unclear how they are related (if at all) and *why* one list may be more useful than others. To understand the various checklists and the relationships between them, the concept of usability needs to be broken down in a way that allows comparisons from both theoretical and practical viewpoints. In section 2 a general background of usability and related knowledge domains are described. In sections 3 and 4 the most well known definitions of usability are discussed together with some common principles and rules. We then propose a layered model of usability which is described in section 5. Section 6 will then discus the usability evaluation process in the light of our model. In section 7 we provide a discussion about the consequences for task modeling and dialogue modeling.

2. KNOWLEDGE DOMAINS

There are many design methods and techniques which all have the goal of designing a usable system.

In iterative design methods the main technique is constant evaluation with users. Other methods take a more structured approach and start with task analysis thereby trying to improve the usability of the initial designs and hopefully to have less iterations than iterative prototyping techniques. This shows two viewpoints on usability in the design process: (a) improving usability by evaluation with users and (b) improving usability already *during* design by applying all available relevant knowledge. Each of the viewpoints is important and an ideal design process uses both viewpoints effectively. UI design is a process that involves knowledge input from several domains. Each of the knowledge domains is necessarily needed but the actual influence may change per project. The knowledge domains are the primary source of information for improving usability and at the same time also a source when evaluating usability and searching for *causes* of sub-optimal usability.

2.1 Knowledge about Humans

The systems we design are being used by humans so we need to know the abilities and limitations of humans. Especially cognitive and perceptual abilities are relevant to design. Humans have serious limitations when it comes to information processing and other tasks such as decision making, searching and scanning (Shneiderman 1998). The fields of cognitive psychology and ergonomics give a theoretical and practical background on these matters. Research in those fields has given useful knowledge that can be used in practice, for instance knowledge about short and long term memory can directly be used to improve learnability of systems. In the past, methods such as GOMS (Card, Moran, and Newell 1983) and CCT (Kieras and Polson 1985) have tried to incorporate cognitive aspects to predict the influence of changes to dialogue aspects of a design. Another important aspect of knowledge about humans is the social and organizational viewpoint. Users perform their tasks in a larger context where they have a social and organizational position that is important to them. In this context they may have to work together or are part of a team. Contextual aspects about users have a less direct impact on the design process and are strongly related to the position of a new system in the organization where it is going to be used. Generally speaking, usability problems are caused by a mismatch between the users' abilities and the required abilities that the system enforces on the users.

2.2 Knowledge about Design

Every designer acquires skills and experiences during projects and that knowledge helps the designer in later projects. This *design knowledge* comes from both practical experience and from literature. Currently the amount of design knowledge available in literature is rather limited which makes the personal experience of the designer an important factor for the usability of the design. Basically the only concrete design knowledge that can be used during design is embedded into *guidelines*. Several guidelines exist but there is no agreement on the form guidelines should have. Some guidelines such as the Macintosh (Apple Computer Inc. 1992) or MS Windows (Microsoft Press 1995) guidelines mainly describe a platform *style* and hardly contain concrete guidelines. The underlying assumption that applications that have been designed according to the guidelines have good usability remains unjustified. Other guidelines such as Mosier's (Smith and Mosier 1986) focus on narrow scoped list of guidelines dealing with detailed design choices and consequently they are quickly outdated by new developments of technology. Despite the differences in guidelines they certainly embody design knowledge and every designer should know them. However, there may be several reasons why the guidelines are not followed during the design process. Even if a designer *tries* to use the guidelines there are still many problems *applying* them. In (Dix, Abowd, Beale, and Finlay 1998) a number of problems with guidelines are discussed such as when to apply a guideline or choose one out of contradicting guidelines. Also the effectiveness of guidelines is under discussion and research has shown that not all guidelines are as practical as desired (Scapin and Bastien 1997). Some older guidelines were designed for designing character based applications and it is not clear in how far they apply to e.g. WIMP interfaces or Virtual Reality interfaces. Another way of capturing design knowledge is in *design patterns* (Bayle 1998). Such patterns describe generalized problems and proven solutions that can be immediately used in practice. Research on design patterns has just started and no concrete results are available yet.

Guidelines deal with both structural (the dialogue) and presentational aspects of a design. For example, guidelines on color use and button sizes refer to the presentation and guidelines on feedback and menu structure deal with dialogue. Usually no explicit distinction between dialogue and presentation is made, although both have a distinguishable impact on usability. Since guidelines often go into depths on describing a platform's style, mainly presentational aspects are covered and there is little guidance for structural aspects. Because design patterns work from a problem to a solution it is more likely to find guidance on structural aspects emphasized in design patterns.

2.3 Knowledge about the Task World

Besides the design knowledge needed for a good design, every project also needs the right information about the specific design case for basing the design on. Both knowledge about humans and knowledge about design is domain-independent but the task world knowledge is different for every design project. Task analysis should provide the information for the requirements of the system both in the functional sense but also in the ergonomic and cognitive sense. The functional side of a task analysis can be transferred quite directly to the design but the ergonomic and cognitive side is very hard to transfer into design. First of all because it is not clear what the relevant information is; what needs to be known in the task model in order to contribute to a more usable design? Secondly, because cognitive aspects are difficult to translate into concrete design decisions. One of the weak points in task analysis research is that it is difficult to justify how task analysis helps to design more *usable* systems as far as this is not directly based on functional requirements of the systems. An answer to this question may be given if it can be defined which properties of a design make the design usable. It can then be seen which information the task model ideally needs to contain.

3. DEFINITIONS OF USABILITY

There is not one agreed upon definition of usability and usability certainly cannot be expressed in one objective measure. Several authors have proposed definitions and categorizations of usability and there seems to be at least some consensus on the concept of usability and they mostly differ on more detailed levels.

In the ISO 9241-11 (Bevan 1994) standard a rather abstract definition is given in terms of efficiency, effectiveness and satisfaction. "Efficiency" is defined as the *resources expended in relation to the accuracy and completeness with which users achieve goals* and "effectiveness" as the *accuracy and completeness with which users achieve specified tasks*. "Satisfaction" is a subjective measure and concerns the *comfort and acceptability of use by end users*. This definition approaches usability from a theoretical viewpoint and may not be very practical. Nielsen (Nielsen 1993) has a slightly different definition that is specified in elements that are more specific. Nielsen only regards expert users when talking about efficiency although learnability is also directly related to efficiency. Memorability mainly relates to casual users and errors deal with those errors not covered by efficiency, which have more catastrophic results. A similar definition is given by Shneiderman (Shneiderman 1998). Sheiderman does not call his definition a definition of usability but he calls it *"five measurable human factors central to evaluation of human factors goals"*. As can be seen from Table 1, Shneiderman's definition is essentially identical to Nielsen's definition and only differs in terminology.

ISO 9241-11	Shneiderman	Nielsen
Efficiency	Speed of performance	Efficiency
	Time to learn	Learnability
Effectiveness	Retention over time	Memorability
	Rate of errors by users	Errors/Safety
Satisfaction	Subjective satisfaction	Satisfaction

Table 1 Usability as in ISO 9241-11, B. Shneiderman and J. Nielsen

Table 2 shows the usability factors as described by Dix (Dix, Abowd, Beale, and Finlay 1998). This categorization looks rather different from the ISO and Nielsen definitions. Dix defines three main groups; learnability, flexibility and robustness suggesting that those concepts are on the same abstraction level. The groups are specified further by factors that *influence* the concept they belong to. For instance, consistency influences learnability positively when a design is consistent within the application and between applications on the same platform. Learnability is subdivided into aspects that are mostly of cognitive nature thereby giving more grip on the important cognitive skills of users in relation to learnability. Robustness corresponds more or less to effectiveness. In flexibility also some lower level concepts such as multi-threading are mentioned but most aspects are mainly related to efficiency.

Learnability	Flexibility	Robustness
Predictability	Dialog initiative	Observability
Synthesizability	Multi-Threading	Recoverability
Familiarity	Task Migratability	Responsiveness
Generalizability	Substitutivity	Task conformance
Consistency	Customizability	

Table 2 Usability categorization by Dix et al.

When comparing these categorizations and definitions it is remarkable that Nielsen and the ISO standard give a concise outline of the term usability while Dix focuses more on the concrete elements that influence usability. From a practical viewpoint, Dix's categorization gives the designer concrete measures for improving the usability of a design. On the other hand, it is odd that Nielsen's notions of efficiency or error rate can not be found in Dix's categorization, as they are clear indicators of usability. The most interesting aspect of Dix's categorization is that it raises the question what the *causes* for sub-optimal usability might be and how it might be improved.

4. PRINCIPLES AND RULES

In addition to definitions of usability, there are also several lists of design principles, heuristics or criteria. Nielsen gives a set of *heuristics* to follow that should have a positive effect on his categories. These heuristics are kind of general guidelines that should be followed, for example: "forgive the user" or "give feedback". Shneiderman gives similar heuristics in his *8 golden rules* for design (Shneiderman 1998):

1. Strive for consistency
2. Enable frequent users to use shortcuts
3. Offer informative feedback
4. Design dialogs to yield closure
5. Offer error prevention and simple error handling
6. Permit easy reversal of actions
7. Support internal locus of control
8. Reduce short-term memory load

From both Dix's categorization and Nielsen's heuristics is shows that the root factors that influence usability need to be found in the cognitive and perceptual abilities of users such as long and short-term memory, problem solving, decision making, searching and scanning (Shneiderman 1998). On the other hand, knowledge about the specific design project expressed in task models is important, especially concerning effectiveness. A similar list is given by the ISO 9241-10 (ISO 1996) standard and is called a set of *dialogue principles*, see Table 3.

Dialogue Principles
Suitability for the task
Self-descriptiveness
Controllability
Conformity with user expectations
Error tolerance
Suitability for individualization
Suitability for learning

Table 3 ISO9241-10 Dialogue Principles

Another interesting list is the list of ergonomic criteria developed by Bastien and Scapin (Scapin and Bastien 1997), see Table 4. Scapin's list of ergonomic criteria mentions "grouping and distinguishing items". Grouping is concerned with the *"visual organization of information items in relation to one another"* and is therefore concerned with presentational aspects. Most other lists mention structural aspects rather than presentational aspects.

1. **Guidance**
1.1 Prompting
1.2 Grouping and distinguishing items
1.2.1 Grouping by location
1.2.2. Grouping by format
1.3 Immediate feedback
1.4 Legibility
2. **Workload**
2.1. Brevity
2.1.1. Conciseness
2.1.2. Minimal actions
2.2 Information density
3. **Explicit control**
3.1. Explicit user actions
3.2. User control
4. **Adaptability**
4.1 Flexibility
4.2. Users' experience
5. **Error management**
5.1. Error protection
5.2. Quality of error messages
5.3. Error correction
6. **Consistency**
7. **Significance of codes**
8. **Compatibility**

Table 4 Ergonomic Criteria by Scapin

Usually there is no explicit distinction between dialogue and presentation level aspects and only the design as a whole is considered. It is useful to realize that measures have both dialogue and presentation aspects. However, often a clear distinction cannot be made. Mullet and Sano (1995) show the importance of presentational aspects and their effect on usability. In addition, they also provide techniques for improving presentational aspects such as grouping, grids etc.

5. A LAYERED MODEL

All the different definitions and principles make usability a confusing concept when actually designing a new system. Usually authors spent a lot effort trying to find out what is the "best" set of principles or to define a "complete set of heuristics". Although these "aids" are useful it remains unclear how they are related and how to judge when an "aid" is useful to improve usability. Figure 1 shows a layered model of usability that helps understanding the various aids. On the highest level, the ISO

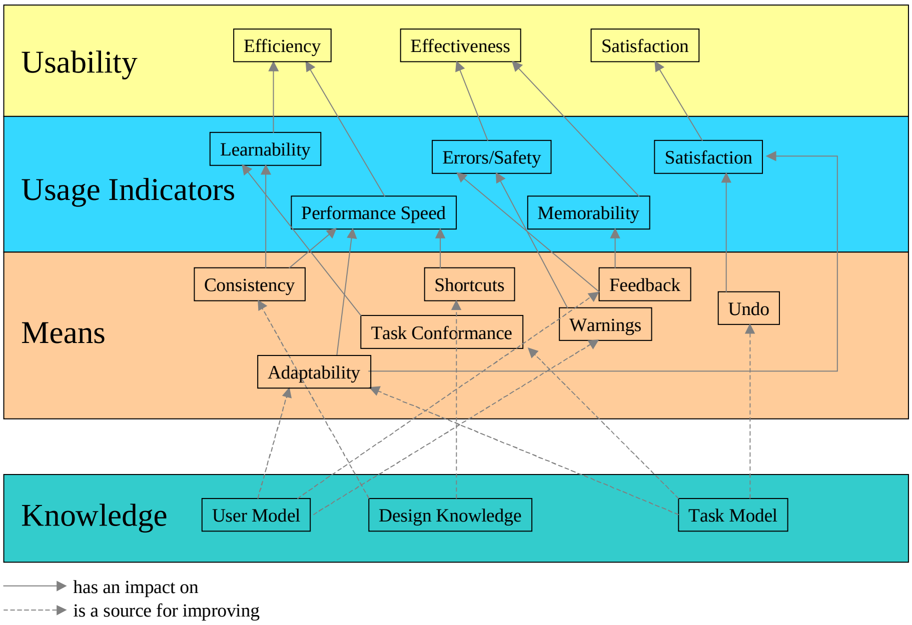

Figure 1 Layered model of usability

definition of usability is given split up in three aspects: efficiency, effectiveness and satisfaction. This level is a rather abstract way of looking at usability and is not directly applicable in practice. However it does give three solid pillars for looking at usability that are based on a well-formed theory (Bevan 1994). The next level contains a number of *usage indicators* which are indicators of the usability level that can actually be observed in practice when users are at work. Each of these indicators contributes to the abstract aspects of the higher level. For instance, a low error-rate contributes to a better effectiveness and good performance speed indicates good efficiency.

One level lower is the level of *means*. Means cannot be observed in user tests and are not goals by themselves whereas indicators are observable goals. The means are used in "heuristics" for improving one or more of the usage indicators and are consequently not *goals* by themselves. For instance, consistency may have a positive effect on learnability and warnings may reduce errors. On the other hand, there may be good reasons for not complying completely with a consistent platform style. Each means can have a positive or negative effect on some of the indicators. The means need to be "used with care" and a designer should take care not to apply them automatically. The best usability results from an optimal use of the means where each means is at a certain "level", somewhere between "none" and "completely/everywhere/all the time". It is up to the designer to find those optimal levels for each means. In order to do that the designer has to use the three knowledge domains (humans, design, and task) to determine the appropriate levels. For example, when design knowledge is consulted by using *guidelines*, it is clear that the guidelines should embody the knowledge of *how* changes in use of the means affect the usage indicators.

The means of Figure 1 are examples of means and usage indicators and the given set is certainly not complete. The different lists and heuristics all give suggestions for useful means. More research is needed to determine which means are most effective for improving usability.

5.1 Comparing Definitions

When comparing the definitions of section 3 using our layered model it is clear that some definitions are on one level and that others have aspects from more than one level. For instance the dialogue principles of the ISO9241-10 standard mention "suitability for learning" (which is learnability) and "error tolerance" which are both usage indicators, but it also mentions "suitability for individualization" (=adaptability) which is a means. Looking at the usability categorization of Dix it shows that mainly means are summed up and the categories are a mixture of means and indicators; learnability is an indicator and flexibility and robustness are means. Scapin's list of ergonomic criteria is essentially a list of means grouped together. Shneiderman's golden rules say to strive for "a certain level" for each of the eight specific means he considers the most important.

When means and indicators are mixed in one list the semantics of the list are ambiguous which causes confusion and makes it more difficult to apply them for actual design decisions. Additionally, it is good to realize that none of the list can be regarded as being complete. Each of the lists has at least one element not mentioned by any of the others. Therefore, all of the lists can be useful but the most important thing is to realize what the semantics of a list is. That way it is clear how a list can be used and what the limitations are.

6. USABILITY EVALUATION

Evaluation of usability can be done *afterwards* and *during* design. The usefulness of all the guidelines, heuristics and other aids is related to the kind of evaluation that is being conducted. Using our model, it is clear that when evaluating with users, evaluation is being done by looking at the *usage indicators*. When evaluating during design, the usage indicators do not provide any data and one has to look at the *means* and make an estimate on their impact.

6.1 Evaluating with Users

Evaluating with users is good method for obtaining data about the actual usage. Using scenarios and other techniques, data about the number of errors or speed of performance can be obtained which should provide a good indication of the usability of the product. However, when usability is not up to par it is important to find out *why* the level of usability is unsatisfactory. In that case, the usage indicators do not help much. One solution is looking at how the *means* were used in the design and another way is by consulting the knowledge from the user and task model.

6.2 Evaluating during Design

Evaluation during the design process is more problematic than evaluating with users. The usage indicators cannot be evaluated directly and therefore do not provide any hard data. What can be done is looking at the *means* that influence the usage indicators. Using walkthroughs and scenarios each of the means can be evaluated by looking at the way they are present in the design and by estimating the positive or negative impact on the usage indicators. For instance, it can be checked if a design is consistent with a platform's style guideline or if in sufficient warnings are given.

Another way of ensuring usability during the design process is by using formal design models. Many models and techniques exist for describing designs using formal notations. State charts, GOMS, ConcurTaskTree's (Palanque and Paterno 1997) and similar notations are being used to describe designs. These kinds of notations are usually strong in describing structural aspects of a design (the dialogue structure) and very weak at describing presentational aspects. In (Payne and Green 1989) Payne says, *"as far as TAG is concerned, the screen could be turned off"*. In relation to the means of our model, this is already a big limitation since a lot of means such consistency, warnings or feedback are strongly related to presentational aspects. Another factor is that most formal models are usually built with the viewpoint of describing "correct" use of the application and therefore do not describe error handling or issuing of warnings.

6.3 Improving Usability

When an evaluation shows that the usability needs to be improved the problem is to find out *which* means need to be changed and *how* they need to be changed. As was mentioned earlier each means may have a positive effect on one usage indicator while having a negative effect on another. In some cases it may be obvious how to improve usability but in cases where problems are of a more structural kind it may not be so simple to solve. In that case, the designer has to take a step back and look at the knowledge domains again. The knowledge domains are the only sources for judging why and how a means is to be changed. For instance, when the task conformance is seen as a problem the task model can give the designer information about what is wrong with the task conformance. Similarly, the user model may give information about the memory limitations which may require the design to have more or better feedback of user actions. Unfortunately the knowledge domains are not always available or written down in a way that makes it easy to use them in practice. Task models may not contain the right information or the available guidelines do not say anything about a particular problem.

7. DISCUSSION

Our layered model of usability gives a somewhat broader perspective on usability and how to achieve good usability in practice. However, it also shows that there is a dependency on the knowledge available. From a theoretical point of view, it is easy to talk about task models but if the task modeling methods available do not produce the task models with the needed information, the task model is not helping to improve usability. Literature on task modeling still has not convincingly shown how the task models contribute to usability other than improving task conformance. The same goes for user models and design knowledge. There are many design guidelines but it is difficult to separate style definitions with real guidelines and the guidelines themselves are defined rather informally. Design guidelines should tell the designer how the means are used most effectively but currently the guidelines are not in such an explicit form. It seems likely that not every means is equally important and that means could be organized into lists ranked by impact under certain constraints.

Beside knowledge aspects, research is still weak on ways for ensuring usability *during* the design process. After all, it is better to do it "right" the first time than having to rely on iterative prototyping with user testing. What is needed is a way to use models for usability evaluation that really give a reasonable indication of usability level. However, most modeling techniques only allow some performance evaluation such as by looking at interaction path lengths. Modeling techniques also need to be able to address issues such as task conformance, warnings, undo and feedback. Although it may be very useful to use those techniques for other purposes, if the set of concepts in modeling techniques is not expanded beyond states and state-changes, those modeling techniques cannot be used to ensure usability during design. The next section will discuss some directions for task and dialogue modeling improvements.

7.1 Addressing task models

Task modeling research has a strong background in cognitive psychology and the focus was on how users perform their work and think about their work from the viewpoint of looking at one user. The strongest link was the fact that if you know more about the user and his work you can build a more usable system. In practice, most modeling methods such as HTA did not model much more than a task hierarchy. Using the task hierarchy only helps to establish *task conformance* and does not help to improve other means such as adaptability or error prevention. However, when the task model is taken as model describing the users, their work, the objects they use and the organization they are part of, it is possible to capture information that can actually *help* to improve usability. The task model should be able to *answer* questions about the task world related to effective use of means. Table 5 shows some question for a task model in relation to a means. As can be observed from Table 5, a task model needs to contain more than a simple task hierarchy. Task analysis methods such as GTA (Van der Veer, Lenting, and Bergevoet 1996,van Welie, van der Veer, and Eliëns 1998) look at much more aspects of the task world such as roles, agents, objects, event and their relationships.

Means	Question for task model
Warnings	What are the critical tasks? How frequent are those tasks performed? Always performed by the same user?
Adaptability	Which types of users are there? Which roles do they have? Which tasks belong to which role?
Undo	Which tasks should be undoable? Which tasks have undoable side effects?
Error prevention	What errors are expected? What are the consequences for users? How can prevention be effective?

Table 5 Questions for task models

Besides these concepts, the right information about the concepts needs to be captured. For instance, when a designer wants to know what the critical tasks are, the task model must be able make a distinction between critical and non-critical tasks, for instance by means of task typing. When the questions from Table 5 need to be answered, all of these aspects and probably even more need to be added. We intend to use the means to check whether our task analysis method GTA (Van der Veer, Lenting, and Bergevoet 1996) contains the necessary information and indeed add missing aspects.

7.2 Addressing dialogue modeling

Dialogue modeling and especially *formal* dialogue modeling (Palanque and Paterno 1997) is gaining interest in HCI research. One problem of most formal methods such as described in (Palanque and Paterno 1997,Payne and Green 1989) is that they are designed to describe the *behavior* of interface and not to enable usability evaluation. Some methods can be used to do verification of systems but this is limited to properties such as state-reachability, deadlocks and interaction path lengths. Although interaction paths can say something about the speed of performance, it is impossible to make predictions about other usability aspects. In the same way as for task models, the means can be used to determine some requirements for dialogue models that enable usability evaluation. A dialogue model also needs to be built using the right concepts and they should be verifiable in some respect.

Means	Questions
Warnings	When are warnings given?
Speed of performance	How many steps needed for accomplishing a task?
Undo	Which functions are undoable?
Feedback	When and how is feedback given?
Consistency	What are similar task-action decompositions?

Table 6 Questions for dialogue evaluation

Looking at Table 6 is it clear that a dialogue model needs to be more than a state-based description. A dialogue model must be able to identify system feedback as either a warning or state feedback and must also contain more detailed information about the functionality as in how far it can be undone or not. In fact there are techniques that partially address these aspects; UAN (Hix and Hartson 1993) deals with explicitly with feedback and TAG (Payne and Green 1989) allowed analysis of consistency by identifying similar task-action decompositions. When such additions are done, a dialogue can be evaluated by looking at how well constraints are satisfied, e.g. "Does the user get a warning before executing a function that is undoable?" or "Given a starting point what is the average number of steps needed to perform this task?"

8. CONCLUSIONS

This paper has discussed several definitions of the concept usability along with heuristics and other guidelines. The proposed layered model of usability gives a division into usage indicators and means that affect them. The model gives a view on usability that can be used in both a practical and a theoretical way. Using this model, the definitions of usability were discussed again and it was shown that some definitions and guidelines are actually a mix of usage indicators and means. In addition, usability evaluation was discussed in the context of our layered model, focussing on usability evaluation during and after the design process. Although the model incorporates several knowledge domains as sources for improving usability, it has been argued that the knowledge domains are in practice hard to use or may not contain the appropriate information.

9. REFERENCES

Apple Computer Inc. (1992), *Macintosh Human Interface Guidelines*, Addison-Wesley Publishing Company,

Bayle, E. (1998), Putting it All Together: Towards a Pattern Language for Interaction Design, *SIGCHI Bulletin*, vol 30, no. 1, pp.17-24.

Bevan, N. (1994), *ISO 9241-11 Ergonomic Requirements for Office Work With VDTs.*

Card, S.K., Moran, T.P. and Newell, A. (1983), *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Ass, Hillsdale.

Dix, A., Abowd, G., Beale, R. and Finlay, J. (1998), *Human-Computer Interaction*, Prentice Hall Europe,

Hartson, H.R. (1998), Human-computer interaction: Interdisciplinary roots and trends, *The Journal of Systems and Software*, vol 43, pp.103-118.

Hix, D. and Hartson, H. R. (1993), *Developing User Interfaces : Ensuring Usability Through Product & Process.* John Wiley & Sons, Inc,

ISO (1996), *ISO 9241-10 Ergonomic Requirements for Office Work With VDTs - Dialogue Principles*.

Kieras, D. and Polson, P.G. (1985), An approach to the formal analysis of user complexity, *International Journal of Man-Machine Studies*, vol 22, pp.365-394.

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

Mullet, K. and Sano, D.K. (1995), *Designing Visual Interfaces: Communication Oriented Techniques*, SunSoft Press (Prentice Hall).

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

Palanque, P. and Paterno, P. (1997), *Formal Methods in Human-Computer Interaction*, Springer-Verlag, London.

Payne, S.J. and Green, T.R.G. (1989), *Task-Action Grammar: the model and its developments*, in: Diaper, D., Task Analysis for Human-Computer Interaction , Ellis Horwood, Cambridge MA.

Scapin, D.L. and Bastien, J.M.C. (1997), Ergonomic criteria for evaluating the ergonomic quality of interactive systems, *Behaviour & Information Technology*, vol 16, no. 4/5, pp.220-231.

Shneiderman, B. (1998), *Designing the User Interface*, Addison-Wesley Publishing Company, USA

Smith and Mosier (1986), *Guidelines for Designing User Interface Software*, MITRE,

Van der Veer, G.C., Lenting, B.F. and Bergevoet, B.A.J. (1996), GTA: Groupware Task Analysis - Modeling Complexity, *Acta Psychologica*, no. 91, pp.297-322.

van Welie, M., van der Veer, G. C., and Eliëns, A. (1998), An Ontology for Task World Models, *Proceedings of DSV-IS98*, Abingdon UK, Springer-Verlag, Wien.