IV

Evaluating Style and Content separately, during the design of a user interface

MAGNUS LIF

Center for Human-Computer Studies (CMD), Uppsala University, Sweden.

Abstract

A method for evaluating user interfaces is presented. Here the evaluation of the style is separated from the evaluation of the content of the interface. HCI experts evaluate the style by using sets of guidelines, i.e. heuristics, related to the style. Users evaluate the content with scenarios and sets of guidelines related to the content. The evaluations are separated to make good use of both groups knowledge. The users are experts on the domain and the HCI experts on usability. The method is to be included early in an iterative user centred development process in order guide the design towards an efficient and effective user interface.

Two studies are presented. The first study shows that domain experts identifies more usability problems then HCI experts for certain heuristics. The other study shows that evaluators with limited HCI knowledge obtain better results when using this method than without support from a method.

Keywords

Guidelines, evaluation, content, style, heuristics, users

1. Introduction

How should the process of design be carried out to result in a user interface that effectively support the users in their work? This is one of the key questions within Human-Computer Interaction (HCI). There is of course not just one answer to this question. However, there is no doubt that it is necessary to involve users during the design process. The question is rather how to involve the users in the most beneficial way.

Gould and Lewis (1985) present some key principles for user centred design. These are early focus on users, direct contact with potential users, iterative design and empirical measurement. In the last few years, this approach to design, in different forms, has become quite common. In user centred design, normally at least two groups are involved – software engineers and end-users. The actual design should then be performed in an iterative way where users and software engineers collaborate. When designing an interface according to this approach one would think that the information system, more or less automatically, would be optimised for the end-users since they make all the design decisions. However, during observations performed within different development projects we have seen that this is not always the case. One reason for this is that users seldom have enough knowledge or experience in design. Gould, Boies, and Lewis (1991) have stressed the importance of separating the *content* of an application, (i.e. the substance), from the *style*, (i.e. the look and feel). The content is more related to the users work, e.g. what data that is needed when making a certain decision. The style includes what the screen should look like and what human computer interaction techniques to use. The users know what a useful system should contain in terms of information needs and functionality. However, to create the style of the interface, knowledge in HCI is needed. Therefore it seems like a good idea to let the users decide what the content of the system should be and then let people with HCI knowledge make decisions concerning the style of the interface. One way of solving this problem would be to evaluate the content and the style separately in the iterative design process in order to guide the design towards a user interface that is efficient for the end-user. It is important that this evaluation can be performed early in the development process (Whitefield, Wilson, & Dowell, 1991). The method also has to be inexpensive and easy to use.

The type of work we mainly study is administrative case handling work (e.g. business administration) performed by *skilled professionals*. A skilled user's requirements on a system sometimes differs from a novice user's. For skilled end-users, it is important that the computer support is efficient in daily use, while for novice users, it is more important that it is easy to learn. There are several examples of guidelines for skilled users conflicting with guidelines for novice users. For a novice user, it is important to minimise the number of items of information on the screen (Nielsen, 1993). However, a skilled user requires, and is able to scan, a lot more information if it always has the same spatial location on the screen (Nygren, Johnson, Lind, & Sandblad, 1992; Nygren, Allard, & Lind, In press). Today's evaluation methods often focus on usability problems concerning ease of learning and not on efficiency in daily use.

1.1. Related Work

Methods for evaluation of user interfaces can be separated into, usability testing methods, i.e. where users are involved, and usability inspection methods, i.e. where users are *not* involved.

A traditional method for user testing is *performance measurement* where the purpose is to measure whether a usability goal is reached or not. User performance is almost always measured by having a group of test users perform a pre-defined set of tasks while collecting data on errors and times. The tests are usually carried out in a laboratory. With such a test many usability problems can be found. One advantage of this test method is that the result is given in hard numbers which makes comparison of different design solutions easy. Unfortunately, in most system development projects there are not enough time, money or laboratory expertise to use this kind of method (Nielsen, 1993). Another problem with laboratory tests is that they are difficult to perform early in the design process since the tests demand a running prototype and require a reasonably complete database. Difficulties in sampling, methodological problems in planning, validity and reliability of obtained measures are other pitfalls in usability testing (Holleran, 1991). Also, evaluation of efficiency in daily use requires skilled users.

Thinking aloud (Lewis, 1982) is another method that involves users. Here, the users verbalise their thoughts while using the system. Through this test, users let the evaluator understand how they view the computer system. This is an inexpensive test that identifies users' misconceptions of the system. It is especially useful when applied by the designer of the interface since he can get direct feedback on the design from the users (Jørgensen, 1990). An example of a drawback with this method is that it is not very natural for users to think out loud. Expert users execute parts of their work automatically (Schneider & Shiffrin, 1977; Shiffrin & Dumais, 1981) why it can be hard for them to verbalise their decision processes.

Questionnaires are especially useful for issues concerning users' subjective satisfaction and possible anxieties (Nielsen, 1993). Questionnaires may be distributed to many users and is an inexpensive survey method. However, it is difficult to get objective results when using questionnaires since the users' answers are based on what they *think* they do, not on what they actually do.

One method that include users, developers and usability experts, and may be carried out early in the design process is *pluralistic walkthrough* (Bias, 1991). With this method, representatives from the three categories meet and discuss usability problems that are associated with the dialogue elements in different scenario steps. In pluralistic walkthrough the main focus is on how users react in different situations. One example is when a user claims that he would "Hold down the shift key while pressing Enter" in a certain situation. Pluralistic walkthrough is an effective method in evaluating the learnability of a user interface. It is not as effective when evaluating the efficiency in daily use since the users are not able to predict how they will interact with the system when they become skilled.

There are also several different inspection methods available. One such method is *cognitive walkthrough* (Polson, Lewis, Rieman, & Wharton, 1992). With this method an evaluator examines each action in a solution path and tries to tell a credible story describing why the expected user would choose a certain action. The credible story is based on assumptions about the users background, knowledge and goals, and on understanding the problem solving process that enables a user to guess the correct action. Cognitive walkthrough is an inspection method that focuses on evaluating a design for ease of learning, particularly by exploration. It is more difficult to evaluate efficiency in daily use. Also, this method does not capture many usability problems concerning the content of the interface, due to the evaluator's limited domain knowledge.

Another inspection method is *heuristic evaluation* (Nielsen & Molich, 1990). The evaluator uses sets of guidelines, i.e. heuristics, and compares those with the interface. The heuristics form a checklist that the evaluator uses during his work. It is easy to learn and inexpensive to use. With heuristic evaluation it is possible to identify many usability problems and it is possible to evaluate early on in the design phase. Unfortunately it is difficult for end-users without knowledge in HCI to use this method. However, heuristic evaluation can be useful when evaluating the style of the interface. The heuristics that Nielsen (1993) suggests work for a broad range of interfaces but have an emphasise on ease of learning. The heuristics are not "optimised" for identification of usability problems concerning efficiency in daily use.

Recently a series of methods for measuring usability has been developed in the ESPRIT MUSIC project (Corbett, Macleod, & Kelly, 1993). The usability of a product is defined through analytic measures, performance measures, cognitive workload measures and user attitude measures. Analytic measures is performed early and are based on a dynamic model of the user interface and on the user tasks. It estimates performance parameters for human interaction dependent on the use of specific interface objects. Performance measurement can be enhanced by using the DRUM tool for analysis of video recording. Cognitive workload are measured through heart rate variability and respiration and subjectively by the use of questionnaires. Questionnaires are also used to measure the user attitude. This is an extensive method that can be used for a number of evaluations. However, measuring efficiency in daily use requires that the users have learned how to use the system which can be rather time consuming.

2. The evaluation method

The method presented in this paper is meant to be included early in the iterative design process as a tool to guide the development of the interface towards a design that is efficient for the users. It is easier to improve the prototype if the potential usability problems are discovered early. The method can be applied even if the first prototype is only a paper mock-up. However, the results from an evaluation is more useful if the prototype is interactive (Nielsen, 1990). The evaluation method presented here consists of two parts:

- Heuristic evaluation of the *style* of the interface

- Scenario based evaluation of the *content* of the interface

The first part is an inspection of the interface, performed by HCI professionals. The second part is performed in co-operation with a group of potential end-users. Both parts includes a checklist with heuristics to guide the evaluation. There is a clear distinction between heuristics concerning the content and heuristics concerning the style of the interface. By this separation each group of evaluators can put more effort into the field where they are experts. Users are experts on the work that is to be supported by the system and HCI-people on usability issues. When an HCI expert is inspecting an interface, he is probably able to find out e.g. if default values are needed. However, the users themselves are usually better at telling *which* information to show as default values. By doing this separation the evaluation can be made more cost efficient. We mean that persons with domain knowledge are capable of identifying more usability problems for the heuristics corresponding to the content of the interface than people without domain knowledge.

The heuristics used in this method are primary based on the work done by Nielsen (1993). Other bases for these heuristics are standards and guidelines (e.g., ISO, 1995), controlled psychological experiments, (e.g., Nygren et al., In press), field studies and participation in development projects (e.g., Borälv, Göransson, Olsson, & Sandblad, 1994). Another important input for the selection of relevant heuristics is the experience we gained developing the ADA method (Åborg, Sandblad, & Lif, 1996). The heuristics presented here differs slightly from the ones proposed by Nielsen in that they are separated into style and content and have an emphasis on efficiency in daily use. The purpose of this paper is not to describe the different heuristics in detail why only a brief description is given.

2.1. Evaluation of the style of the interface

In this phase the evaluator identifies usability problems concerning the style of the interface. This is done in a similar way to heuristic evaluation. The heuristics serves as a checklist for the evaluator. During the inspection he comes up with an opinion about what is good and bad about the interface. It is not necessary to follow the checklist in a specific order, but it is important that all aspects are documented.

In order to get the best result in Heuristic Evaluation it is recommended that 3 to 5 evaluators inspect the interface (Nielsen & Mack, 1994). Heuristic evaluation is more effective if carried out by a usability expert than by a non-expert (Desurvie, Kondziela, & Atwood, 1992).

During our work within different projects the following heuristics have shown to be important and useful. The main focus during this inspection is *how* information and tools are presented in the interface, i.e. the look and feel.

1. Transparency
   The interface has to be transparent to the user. It has to be obvious how to use the information system, how different information on the screen is related and how to execute operations (Nielsen, 1993).

2. Orientation and navigation
   It is important that the user is well oriented in the system, i.e. knows where he is in the system, how he got there and how to precede. This might be achieved by showing an overview at the same time as showing details (Henderson & Card, 1986) or by presenting the way a user has “walked” through the system. It must also be clear if there exist information not immediately available and how to reach that information.

3. Disposition of the screen
   The screen area is a limited resource. When a lot of information has to be simultaneously visible it is important to carefully consider how to dispose the screen area. When working with the computer support it should be clear which the main entities on the screen are and how they are related to each other. A skilled user is able to quickly scan a lot of frequently used information, if it always has the same spatial location on the screen (Nygren et al., In press).

4. Readability
   Use fonts, sizes of characters, colours, shapes and forms in a way that makes the information readable on the screen (Nielsen, 1993).

5. Discrete layout
   Do not overdo it. Use colours, fonts and decorations with care (Nielsen, 1993). Do not use too many colours, fonts, frames, boxes and lines in a user interface. This may interfere with the visual codes of information in the interface. Use discrete colours for unimportant information such as the background.

6. Control and feedback
   The user must be able to control the system and to carry out operations in different orders (ISO, 1995). Distinct feedback from the system should be given in reaction to the user's input within reasonable time (ISO, 1995). The user should be given information if the waiting times are longer.

7. Minimise the users' system related input
   Moving around in the system takes a considerable amount of time. Avoid this by using broad instead of deep menus and minimise the number of steps in a dialogue. Input with a mouse is normally slower than input from a keyboard. Therefore it is important to use shortcuts and default options (Nielsen, 1993). Use codes to show if it is possible to chose a certain alternative or not.

8. Errors and on-line help
   Allow the user to make mistakes (ISO, 1995). Support "undo" and "redo" and give constructive error messages. Minimise probability of occurrence of errors. Give on-line help on the user's initiative (ISO, 1995). This help should be easy to access and understand. If no other solution is possible, give help at the initiative of the system.

9. Consistency
   The system should be consistent in functionality and layout (Foley, 1990). This will encourage exploratory learning and is also necessary for efficiency in daily use.

During a typical inspection the evaluator "runs" the prototype by going through the dialogues a number of times, and tries to identify the advantages and disadvantages with the different design solutions. It is important to write down both the opinion and, if it is not obvious, reason for the opinion. The time this takes differs depending on the complexity of the prototype and the evaluators experience, but it should normally not take more than two hours.

2.2. Scenario based evaluation of the content of the interface

The content of the interface is evaluated by a group of users. The evaluation is performed in sessions led by an evaluator. Prior to the actual evaluation a number of scenarios are identified on which a discussion about the prototype is held. The prototype should be displayed either as paper mock-ups or on a screen.

The evaluator that leads the discussion has to be able to understand the users terminology, and keep the discussion on track without inhibiting the free flow of ideas and comments. He must also be observant so that all members in the group have equal opportunity to express their opinion. It may be difficult for the users to understand how the system will function in their normal work environment. Therefore, the discussion has to be focused around the aspects concerning the content of the interface, such as if there is enough information available when performing a task, if the right information is emphasised, if the default values and language is correct, and so on. The main focus here should be on *which* information and tools to supply in the user interface.

The checklist presented below is used to guide the discussion and evaluate how well the system corresponds to the user's requirements.

1. Work related layout and functionality
   The screen layout should fit the needs of a specific work situation. The information entities and the tools must be relevant for the work, and the information should be logically grouped (ISO, 1995). Use metaphors that are familiar to the user. The metaphors may be found by studying the end-users' current work (Nielsen, 1993).

   - Is the screen layout related to the users' work?
   - Is it obvious how the tasks can be performed?

2. Simultaneous presentation of information
   The user should not have to remember information from one screen when working with another. If possible, show all information that is needed in a work task on the screen simultaneously (Lind, 1991). Information that is frequently used should be visible at all times.

   - Is all information needed when performing a task visible?
   - If not, what information is not visible?

3. Emphasise important information
   Use fonts, sizes of characters, colours, shapes, and/or forms to emphasise important information (Nygren et. al, 1992). Do not emphasise unimportant information, e.g. for a skilled user, data is more important than labels etc. Show task related status with highlights, character styles, colour codes and/or by using dedicated positions for application data. This will probably decrease the amount of time a skilled user will need to locate information (Nygren et al., In press).

   - Is the important information emphasised?
   - Is the emphasised information important?

4. Use shortcuts and default values
   Use shortcuts and default values to minimise the user's input (Nielsen, 1993). Sequences of inputs should be possible to perform in a logical order.
   - Are shortcuts used for frequent operations?
   - Is the proper information given as default values?

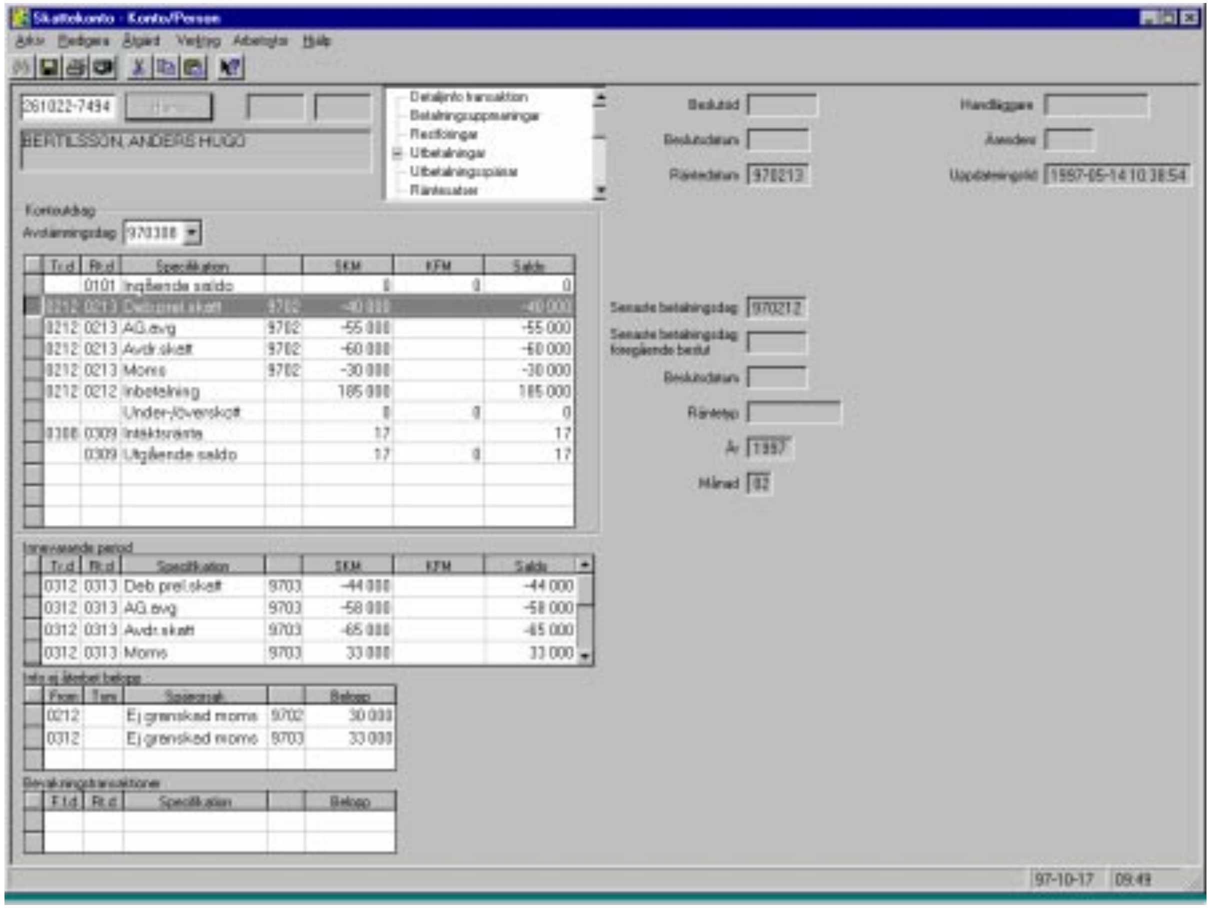
5. Speak the users' language
   System concepts should be expressed in the users' language with words, phrases, symbols, icons, etc. (Nielsen, 1993). Avoid system-oriented terms.
   - Is the language obvious to the user?
   - What terms are difficult to understand?

The results from this discussion is documented in the checklist (see 2.3). The evaluator should also take notes on problems and questions that occur during the session that are not included in the checklist.

2.3. Documentation

The results from the two previous steps are documented according to the structure of the checklists from which it is clear which heuristics are inapplicable, applicable and met, or applicable and not met (ISO, 1995). It is also important to describe the reasons for opinions if they are not obvious.

In the example below we will look at the documentation for one heuristic concerning the style, i.e. "Disposition of the screen area". + is used for applicable and met , - for applicable and not met, and ? for non applicable heuristic. The example shows a prototype from a system development project at the Swedish National Tax Board (Figure 1).



261022-7494

BERTILSSON, ANDERS HUGO

Beslutsid

Beslutsdatum

Räntedatum 970213

Handläggare

Ärendenr

Uppdateringstid 1997-05-14 10:38:54

Kontoutdrag

Avstämningsdag 970308

Tr.d	Rt.d	Specifikation		SKM	KFM	Saldo
	0101	Ingående saldo		0	0	0
0212	0213	Deb.prel.skatt	9702	-40 000		-40 000
0212	0213	AG.avg	9702	-55 000		-55 000
0212	0213	Avdr.skatt	9702	-60 000		-60 000
0212	0213	Moms	9702	-30 000		-30 000
0212	0212	Inbetalning		185 000		185 000
		Under-/överskott		0	0	0
0308	0309	Intäktsränta		17		17
	0309	Utgående saldo		17	0	17

Senaste betalningsdag 970212

Senaste betalningsdag föregående beslut

Beslutsdatum

Räntetyp

År 1997

Månad 02

Innevarande period

Tr.d	Rt.d	Specifikation		SKM	KFM	Saldo
0312	0313	Deb.prel.skatt	9703	-44 000		-44 000
0312	0313	AG.avg	9703	-58 000		-58 000
0312	0313	Avdr.skatt	9703	-65 000		-65 000
0312	0313	Moms	9703	33 000		33 000

Info ej återbet belopp

From	Tom	Spärrorsak		Belopp
0212		Ej granskad moms	9702	30 000
0312		Ej granskad moms	9703	33 000

Bevakningstransaktioner

F.t.d	Rt.d	Specifikation		Belopp

97-10-17 09:49

Figure 1. Screen layout from a prototype at the Swedish National Tax Board

2. Disposition of the screen area

+ The screen area is divided into two parts. The left side shows all important information and the right side shows details. This will give the user a good overview.

– The right side of the screen is not very well utilised. It contains too much white space. This area could be used for showing more details.

– Selections are made in the tree structure on the upper left side. It is possible to make this area bigger in order to show more alternatives at the same time.

Documentation is done in a similar way for all heuristics. The result should be carefully analysed by the designer and function as an input for the redesign of the prototype. The new prototype may then be evaluated in the same way. This iterative process continues until the application is implemented and incorporated in the working environment.

3. Evaluation of the method

Two studies have been performed in order to evaluate the method.

3.1. Study 1

We have separated the heuristics concerning the style of the interface from the heuristics concerning the content. A study has been performed to evaluate if this separation seems appropriate. Our hypothesis is that subjects with domain knowledge are able to identify more usability problems corresponding to the heuristics related to the content of the interface than subjects without domain knowledge, but with HCI knowledge.

Subjects

In total six persons were included in the study. Three of them were domain experts with several years experience from the studied domain. The other three were HCI experts but had only limited knowledge in the domain where the evaluated systems were supposed to be used.

Procedure

The study was performed at the Swedish National Tax Board where five of their prototypes were evaluated by two groups of evaluators. The subjects in group 1 were domain experts with limited usability knowledge, the subjects in group 2 were usability experts with limited domain knowledge. The prototypes differed both in size and completeness. One of the prototypes was in the beginning of the development process and was just realised as bit maps on the screen. The most complete prototype was interactive and had a limited data base.

The subjects were instructed to perform a heuristic evaluation and to identify and document potential usability problems for each prototype. The result of the evaluation was documented as lists of findings. After the evaluation the different findings were categorised according to the heuristics described in this paper.

Results

Heuristic	**Domain experts**	**Usability experts**
Transparency	4	22
Orientation and navigation	10	7
Disposition of the screen	29	23
Readability	26	10
Discrete layout	8	1
Control and feedback	14	16
Minimise the users system related input	18	18
Errors and on-line help	18	2
Consistency	23	36
Work related layout and functionality	37	3
Simultaneous presentation of information	7	2
Emphasise important information	10	0
Use shortcuts and default values	10	0
Speak the users language	43	4

Table 1. Number of identified usability problems for each heuristic.

	Domain experts	**Usability experts**	Total
Style	150	135	285
Content	107	9	116
Total	257	144	401

Table 2. χ^2-test. Number of identified usability problems per group of experts.

	Mean Domain	Mean Usability	U	significance (One-tail)
Work related layout and functionality	12,30	1,00	0	0,05
Simultaneous presentation of information	2,33	0,67	2,5	not sign.
Emphasise important information	3,33	0,00	0	0,05
Use shortcuts and default values	3,33	0,00	0	0,05
Speak the users language	14,30	1,33	0	0,05

Table 3. Mann-Whitney test on heuristics concerning the content.

The number of identified usability problems found by each group were documented for each category (See table 1).

The total number of usability problems identified for style and content were counted for each group of evaluators. A χ^2-test has been applied to verify the hypothesis (See table 2). The test gives a value of $\chi^2 = 56.2$ (df=1, $p<0.001$).

A Mann-Whitney test has been applied to verify the hypothesis for each heuristic concerning the content (See table 3). The mean value has been derived for each heuristic per group of evaluators. The number of subjects in each group were three.

Discussion

Table 1 shows the total number of identified usability problems corresponding to the different heuristics for each group. According to the results the group without domain knowledge were not able to identify many problems concerning the content compared to the group with domain knowledge. This also holds for two other heuristics: "Discrete layout" and "Errors and on-line help". A probable reason for the result corresponding to these two heuristics is that the group with HCI experts seemed to take into consideration that the prototypes were in an early stage of development which made these heuristics less applicable.

According to the result of the χ^2-test presented in Table 2 the domain experts are able to identify more usability problems concerning the content as compared to usability experts with limited domain knowledge. The domain experts identified more than 10 times as many content problems as the usability experts.

In table 3 results from the tests on significance has been presented. A low significance value means that the probability that the result has occurred by chance alone is low. The results are significant for all heuristics except one, "Simultaneous presentation of information”. One reason for this is that the domain experts did not identify so many problems related to this heuristic. However, we still believe that domain knowledge is needed to be able to identify which information that has to be simultaneously present on the screen.

When studying table 1 it seems like the domain experts are better than the HCI experts to identify most of the usability problems as well. However, the findings have not been weighted in the analyses. A closer look shows that the HCI experts has identified more major problems than the domain experts.

3.2. Study 2

The utility of this method were evaluated during a workshop held by the author and his colleague at the Swedish National Tax Board. Our hypothesis was that evaluators with limited HCI knowledge, after a half day tutorial, would be able to identify more usability problems when evaluating a user interface *with* this method than *without* the support of a method.

Subjects

There were 18 subjects employed at the Swedish National Tax Board participating in this evaluation. 16 subjects were software engineers and two were employed for handling the taxes. All of the subjects had limited knowledge in HCI.

In this evaluation, prototypes of systems for reporting working hours were studied. All participants were experienced users of similar systems.

Procedure

This evaluation was performed during a workshop. The subjects were divided into two groups. In the beginning of the work shop the subjects were asked to evaluate a user interface of a system. Two different prototypes were evaluated. Both prototypes were presented as screen dumps.

In the first session group 1 evaluated prototype 1 and group 2 evaluated prototype 2. In this session no particular method was used. The subjects were only instructed to identify and document potential usability problems. The subjects worked together in pairs. In total their were 9 pairs participating. The first session ended after approximately 30 minutes.

After the first session we taught the subjects how to use the first part of the method i.e. how to evaluate the style of the interface. This part of the course lasted for approximately two hours. The subjects were then instructed to evaluate the style of the interface by using the first part of the method. All findings were documented in pre-defined forms. In this session group 2 evaluated prototype 1 and vice verse. This session lasted for approximately 20 minutes.

After this session the subjects were taught how to use the second part of the method, i.e. how to evaluate the content of the interface. This lesson ended after approximately two hours. One main scenario were identified to be used by all participants. The subjects were then instructed to evaluate the content of the interface by using the second part of the method. In this session the groups evaluated the same prototype as in session 2. All findings were documented in pre-defined forms as in session 2. After approximately 20 minutes the evaluation was completed.

After the workshop the results were analysed. The findings reported during each evaluation were given a weight. This was done by an expert with HCI knowledge. Major usability problems were given a weight of 2, minor usability problems were given a weight of 1 and problems of no practical importance were given a weight of 0.

Results

	Without method		With method		W	Significance (one-tail)
Total no		36		125	0	0,005
Mean	4,0		13,9			
Major		11		23	6	0,050
Mean	1,2		2,6			
Minor		25		102	0	0,005
Mean	2,8		11,3			

Table 4. Wilcoxon test. Number of identified problems, totally, major and minor.

	Without method		With method		W	Significance (one-tail)
Weighted		46		148	0	0,005
Mean	5,1		16,4			

Table 5. Wilcoxon test. Score of weighted findings.

The total number of major and minor usability problems identified by each group has been documented (See table 4). A Wilcoxon test has been applied to verify the hypothesis for total number of findings and for major and minor findings respectively. The raw with the total number of findings does not include problems that has been given a weight of 0.

Each finding were given a weight. A Wilcoxon test was applied on the total score of the weighted findings (See Table 5).

Discussion

According to table 4 more usability problems were found when using the evaluation method than without support from a method. There is a rather big difference between the number of problems found when using the method as compared to the case when the method was not used. This is also true for both major problems and minor problems. In table 5 the usability problems were given a weight. Also here the difference is big. When comparing the differences for identified major problems with minor problems (see table 4) it is obvious that the difference is bigger for minor problems. One reason for this could be that the total number of potential minor problems that can be found are much higher than the number of major problems.

Other similar studies show that HCI experts are able to identify more usability problems than non-experts when using heuristic evaluation (Desurvie et al., 1992). There are reasons to believe that this would be the case when using this method as well.

4. Conclusions

This method has been used in systems development projects for administrative case handling work. With this method, usability problems have been discovered early in the user centred design process which have made it easier to improve interfaces. The heuristics are focused on usability problems concerning efficiency in daily use and have shown to be valuable when evaluating interfaces for *skilled* users.

When this method has been used the content of the interface has been evaluated by users already involved in the development process. This has the advantage that they know how the system is intended to work. The disadvantage is that they often are fond of their own design solutions and not willing to make changes.

A relevant question when evaluating content and style separately is how the style corresponds to the content and vice versa. Is it, for instance, possible to get the same results, concerning the content, when evaluating two interfaces with the same content but with different style? Probably not. There is an overlap between content and style. We mean that it is not possible to separate them entirely. A bad design of the style can sometimes correspond to a bad design concerning the content as well, e.g. when information needed for a decision are grouped into several overlapping windows on the screen. However, the important thing is not whether the usability problems are discovered during the evaluation of the content, the style or during both phases. The important thing is that they actually are discovered.

We have presented a method for evaluating content and style separately during the design of a user interface. This separation has shown to be fruitful. We believe that this method could be developed further by identifying aspects in which software engineers are experts, e.g. how the technique will effect the design. By letting users, HCI-experts and software engineers evaluate the interface separately, we assume that most of the potential problems could be discovered early in the development process.

References

BIAS, R.C. (1991). Walkthroughs: Efficient Collaborative Testing. *IEEE Software, 8* (5), 94-95.

BORÄLV, E., GÖRANSSON, B., OLSSON, E., & SANDBLAD, B. (1994). Usability and Efficiency. The HELIOS Approach to Development of User Interfaces. In U. Engelmann, F.C. Jean, & P. Degoulet (Eds.), *The HELIOS Software Engineering Environment, Supplement to Computer Methods and Programs in Biomedicine, 45*, 63-76.

CORBETT, M., MACLEOD, M., & KELLY, M. (1993). Quantitative Usability Evaluation – The ESPRIT MUSiC Project, In G. Salvendy, & M.J. Smith (Eds.), *Proceedings of the Fifth International Conference on Human-Computer Interaction, HCI International' 93* (pp. 313-318). Amsterdam: Elsevier Science Publishers.

DESURVIE, H., KONDZIELA, J., & ATWOOD, M. (1992). What is gained and lost when using evaluation methods other then empirical testing. In A. Monk, D. Diaper, & M.D. Harrison (Eds.), *People and computers VII, Proceedings of the BCSHCI' 92* (pp. 89-102). London: Cambridge University Press.

FOLEY, J.D. (1990). Dialogue Design. In J.D. Foley, A. van Dam, S.K., Feiner, & J.F. Hughes (Eds.), *Computer Graphics: Principles and Practise* (pp. 391-431). Reading, MA: Addison-Wesley.

GOULD, J.D., BOIES, S. J., & LEWIS, C. (1991). Making usable, useful, productivity. Enhancing computer applications. *Communications of the ACM, 34* (1), 74-85.

GOULD, J.D., & LEWIS, C. (1985). Designing for Usability: Key Principles and What Designers Think. *Communications of the ACM, 28* (3), 300-311.

HENDERSON, A., & CARD, S.K. (1986). Rooms: The Use of Multiple Virtual Workspaces to Reduce Space Contention in a Window-Based Graphical User Interface. *ACM Transactions on Graphics, 5* (3), 211-243.

HOLLERAN, P.A. (1991). A methodological note on pitfalls in usability testing. *Behaviour & information technology, 10* (5), 345-357.

ISO/DIS 9241 (Draft). Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) - Part 10: Dialogue Principles, Part 11: Guidance on Usability, Part 13: User Guidance (1995). Geneva, Switzerland: International Organization for Standardization.

JØRGENSEN, A.H. (1990). Thinking-aloud in user interface design: a method promoting cognitive ergonomics. *Ergonomics, 33*, (4), 501-507.

LEWIS, C. (1982). *Using the 'thinking-aloud' method in cognitive interface design.* (IBM Research Report RC 9265, 2/17/82). Yorktown Heights, NY: IBM T.J. Watson Research Center.

LIND, M. (1991). *Effects of Sequential and Simultaneous Presentations of Information* (Rep. No. 19, CMD). Uppsala, Sweden: Uppsala University.

NIELSEN, J. (1990). Paper verses Computer Implementations as Mockup Scenarios for heuristic evaluation. In D. Diaper, D. Gilmore, G. Cockton, & B. Shackel (Eds), *Proceedings of Human Computer Interaction, Interact' 90* (pp. 315-320). Amsterdam: Elsevier Science Publishers B.V.

NIELSEN, J. (1993). *Usability Engineering.* San Diego: Academic Press, Inc.

NIELSEN, J., & MACK, R.L. (Eds.). (1994). *Usability Inspection Methods.* New York: John Wiley & Sons, Inc.

NIELSEN, J., & MOLICH, R. (1990). Heuristic evaluation of user interfaces In J.C. Chew, & J. Whiteside (Eds.), *Proceedings of Human Factors in Computing Systems, CHI' 90* (pp. 249-256). New York, NJ: ACM.

NYGREN, E., ALLARD, A., & LIND, M. (In press). Skilled Users' Interpretation of Visual Displays. *Human Computer Interaction.*

NYGREN, E., JOHNSON, M., LIND, M., & SANDBLAD, B. (1992). The Art of the Obvious. Automatically Processed Components of the Task of Reading Frequently Used Documents. Implications for Task Analysis and Interface Design. In J.P. Baursefeld, J. Bennett, & G. Lynch (Eds.), *Proceedings of Human Factors in Computing Systems, CHI' 92* (pp. 235-239). New York: ACM.

POLSON, P.G., LEWIS, C., RIEMAN, J., & WHARTON, C. (1992). Cognitive Walkthroughs: A Method for Theory-Based Evaluation of User Interfaces. *International Journal of Man-Machine Studies, 36* (5), 741-773.

SCHNEIDER, W., & SHIFFRIN, R.M. (1977). Controlled and Automatic Human Information Processing I, *Psychological Rev., 84,* 1-66.

SHIFFRIN, R.M., & DUMAIS, S.T. (1981). The Development of Automatism. In J.R. Anderson (Ed.), *Cognitive Skills and their Acquisition.* Hillsdale, NJ: Erlbaum.

WHITEFIELD, A., WILSON, F., & DOWELL, J. (1991). A framework for human factors evaluation. *Behaviour & information technology, 10* (1), 65-79.

ÅBORG, C., SANDBLAD, B., & LIF, M. (1996). *A practical method for Evaluation of Human-Computer Interfaces* (Rep. No. 75, CMD). Uppsala, Sweden: Uppsala University.