Behaviour & Information Technology
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713736316

Dogmas in the assessment of usability evaluation methods

Kasper Hornbæk [a]
[a] Department of Computer Science, University of Copenhagen, Copenhagen, Denmark

First published on: 29 January 2009

To cite this Article Hornbæk, Kasper(2010) 'Dogmas in the assessment of usability evaluation methods', Behaviour & Information Technology, 29: 1, 97 — 111, First published on: 29 January 2009 (iFirst)
To link to this Article: DOI: 10.1080/01449290801939400
URL: http://dx.doi.org/10.1080/01449290801939400

Behaviour & Information Technology
Vol. 29, No. 1, January–February 2010, 97–111

Taylor & Francis
Taylor & Francis Group

Dogmas in the assessment of usability evaluation methods

Kasper Hornbæk*

Department of Computer Science, University of Copenhagen, Copenhagen, Denmark

(*Received June 2006; final version received January 2008*)

Usability evaluation methods (UEMs) are widely recognised as an essential part of systems development. Assessments of the performance of UEMs, however, have been criticised for low validity and limited reliability. The present study extends this critique by describing seven dogmas in recent work on UEMs. The dogmas include using inadequate procedures and measures for assessment, focusing on win–lose outcomes, holding simplistic models of how usability evaluators work, concentrating on evaluation rather than on design and working from the assumption that usability problems are real. We discuss research approaches that may help move beyond the dogmas. In particular, we emphasise detailed studies of evaluation processes, assessments of the impact of UEMs on design carried out in real-world systems development and analyses of how UEMs may be combined.

Keywords: usability evaluation methods; downstream utility; formative evaluation; methodology; usability testing

1. Introduction

Usability evaluation methods (UEMs) are widely recognised as an essential part of systems development. Much work has focused on developing more accurate, more widely applicable and more cost-effective evaluation methods, for example by iteratively refining descriptions of evaluation methods using data from systematic assessments and comparisons of their performance (for reviews see Gray and Salzman 1998, Hartson *et al*. 2001, Cockton *et al*. 2003a). This study focuses, in particular, on formative evaluation with the use of techniques such as think-aloud testing, heuristic evaluation and cognitive walkthrough.

Such assessments of UEMs, however, suffer from problems concerning validity, reliability and practical utility. Gray and Salzman (1998) pointed out that an often-cited selection of studies that compare evaluation methods suffers from low validity, because of limitations not only in statistical tests and in the conclusions passed on to practitioners and researchers, but also in the measures used to compare methods. The reliability of comparisons of UEMs has also been brought into question. Work on the evaluator effect (Hertzum and Jacobsen 2001) shows that different evaluators find markedly different sets of problems as a result of applying a particular UEM. A study of usability evaluation in industry also found that different teams of evaluators identified different usability problems (Molich *et al*. 2004). Recently, Wixon (2003) made the case that comparisons of UEMs do not appreciate that the real goal of such methods is to impact design, not to generate problems. Thus, comparisons may not properly assess the practical utility of UEMs.

This study extends the above critique by presenting seven dogmas in work on UEMs. We term certain beliefs and methods 'dogmas' because they are widely present, yet rarely questioned, in many recent assessments of UEMs, in the comments by reviewers upon studies with which we are familiar, or in studies discussing evaluation of UEMs. In particular, the dogmas persist in recent research, despite the fact that some were alluded to in the critical reviews mentioned above and that they unnecessarily constrain assessments of UEMs. In using the notion of dogmas, we do not wish to show disdain for the studies we discuss, nor do we intend any of the pejorative connotations that dogma carries. Rather, we use the notion of dogma merely descriptively to convey that conventions in research on usability evaluation – the way studies are typically done – may be problematic. The aim of this study, then, is to synthesise the dogmas and to be more direct than existing reviews in suggesting how to move beyond them.

2. Typical activities in recent assessments of UEMs

In discussing assessments of UEMs, we propose Figure 1 as a sketch of a typical study. This figure is synthesised from a selection of 25 assessments of UEMs appearing from 1999 to 2004 in well-known human–computer interaction conferences and journals,

*Email: kash@diku.dk

ISSN 0144-929X print/ISSN 1362-3001 online

DOI: 10.1080/01449290801939400
http://www.informaworld.com

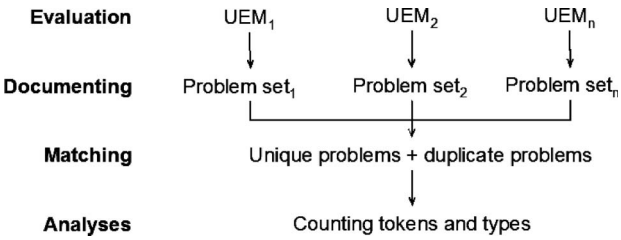

Figure 1. Reference model of comparisons of usability evaluations.

comprising the ACM Conference on Human Factors in Computing Systems (CHI), IFIP International Conference on Human–Computer Interaction (INTERACT), *Behaviour & Information Technology, Human Factors, Interacting with Computers, International Journal of Human–Computer Interaction* and *International Journal of Human–Computer Studies*. The aim of presenting the figure is to give the reader an overview of the assessments of UEMs performed since the review by Gray and Salzman (1998); this overview and the studies that underlie it will be discussed in the following sections. It is not the intent of the present study to present a formal or quantitative review of research.

As shown in Figure 1, recent assessments of UEMs typically included four activities. During *evaluation*, a number of UEMs are applied (UEM_1, UEM_2, etc.), which may be either distinct evaluation techniques (e.g. heuristic evaluation, cognitive walkthrough) or conditions of use of a particular UEM (e.g. expert evaluators, extended task descriptions as input to evaluation). For simplicity, the rest of this study will use the term UEMs to denote both. Usually individual evaluators apply one UEM at a time. Think-aloud testing is often part of the evaluation and is used to produce a set of known usability problems. When *documenting* the evaluation, evaluators individually create a set of descriptions of usability problems; sometimes these are documented using a structured format, sometimes using a free-text format. The evaluator often attaches a severity estimate to each problem. During *matching*, persons other than the evaluators (typically the researchers running the assessment) compare the sets of usability problems. The aim of this comparison is to identify duplicate problems. During *analysis*, problems found using the UEMs under assessment are counted, typically both as problem instances (tokens) and in relation to some kind of classification, say, into problems concerning different areas of the interface (types). The results of this counting are typically used to select one of the UEMs as the best. Note that studies typically do not move beyond analysis to consider how problems may be solved or whether they can be solved.

While the model is certainly simplified, it captures well the main elements of most of the 25 studies appearing between 1999 and 2004 (e.g. Bastian *et al.* 1999, Sears and Hess 1999, Andre *et al.* 2001, Kjeldskov and Skov 2003, Mankoff *et al.* 2003). The studies that do not fit the model will be discussed individually in subsequent parts of the study.

Figure 1 suggests the overall structure of the remainder of the study. In Section 3, we discuss the use of counts of problems in assessments of UEMs; Section 4 discusses the widespread use of matching in studies of usability evaluation; Section 5 discusses one common understanding of how evaluators use UEMs to produce their problem sets; Section 6 discusses the widespread focus on individual problems as the unit of analysis; Section 7 discusses the focus on evaluation in isolation from design; Section 8 discusses the insistence on finding a single best method; and Section 9 discusses the treatment of usability problems as something real and definite.

3. Dogma No. 1: problem counting as the main approach to assessing UEMs

The most common analysis of the outcomes of evaluators' application of a UEM is to count the number of problems identified. Sometimes counting is done on the level of problem tokens; that is, the usability problems listed by evaluators, sometimes on the level of problem types, for example as derived from a classification of the usability problem. The UEM that helps evaluators identify, on average, the most usability problems is then typically argued to be the most effective method.

This approach is surprisingly common. Among the studies reviewed by Gray and Salzman (1998), counts of problems were used in four studies to find the most effective UEM or condition for UEM use (Jeffries *et al.* 1991, Desurvire *et al.* 1992, Karat *et al.* 1992, Nielsen 1992). More recent studies of UEMs also use this approach (e.g. Bastian *et al.* 1999, Mankoff *et al.* 2003, Markopoulos and Bekker 2003). Some of these studies state the dogma in straightforward terms, for example: 'The primary basis for evaluating a technique should be the number of usability problems it helps identifying' (Kjeldskov and Stage 2004).

The above approach has at least four limitations. Already Gray and Salzman (1998) discussed what they termed the 'problem counting approach'. They argued that just counting the number of problems conflates naming of potential problems with identifying real problems, because counts of potential problems will include problems that are not true usability problems. Thus, the predictive validity of UEMs, which is 'the relevance of the problems detected under a particular

method' (De Jong and Schellens 2000), is not assessed through problem counts. The counter-argument that finding many problems may be seen as a sign of the discovery strength of a particular UEM, and that high discovery strength should help find also the important problems, appears unconvincing when not backed up by data.

A second limitation of the problem counting approach is that different kinds of problems – for example with respect to generality, type, aspects of the user interface covered, or clarity – are given equal weight when counted. This limitation becomes serious if different UEMs lead evaluators to produce different kinds of problem description. Studies have suggested that problems found by different UEMs may vary with respect to, for example, the content of problems (Sears and Hess 1999), the level of performance at which a problem is found (Fu and Salvendy 2002), and the complexity of the solutions required (Hornbæk and Frøkjær 2004a). Additionally, the descriptions of problems discovered by different UEMs may be more or less clear, contain different number of design suggestions, and be at varying levels of specificity. Problem counting ignores this evidence.

Dennis Wixon (2003) argued that most comparisons of UEMs fail the practitioner. According to Wixon, the literature on evaluation of methods assumes that the '[n]umber of problems detected is the most appropriate criterion for evaluating a method' (p. 30). He argued that problem identification is only the first step towards improving the product; finding ways to fix the problems identified is equally important. Following this view, a third limitation of the problem counting dogma is that finding a large number of usability problems is not necessarily an indication of the quality of a UEM if those problems do not, or perhaps cannot, lead to changes in the problematic software. Underlying the dogma of counting usability problems is the contention that UEMs can be assessed in isolation from design (see Section 7).

Finally, a severe limitation of the problem counting dogma is that it ignores the straightforward observation that evaluators may learn more about usability issues than they document in their lists of usability problems. In the context of systems development, Grudin (1991) discussed various gaps in the communication between users and developers/designers imposed by different development contexts. Similarly, in the context of usability evaluation, gaps may exist to different degrees between usability evaluators and system developers. In some cases, the reporting of problems may be the only point of contact between the two groups; in others, usability evaluators will be actively involved in redesigning the system under development or developers will be involved in the evaluation. In the latter context, the counting of problems ignores several fruitful, but subtle, outcomes of usability evaluations. For example, it has been argued that one positive aspect of the think-aloud method is the experience of seeing users struggle with an interface (Helms Jørgensen 1990). This would not be captured by a list of problems, but routinely takes place in practical UEM use.

Thus, using raw counts of problems to reason about UEMs has limited validity as an indicator of the quality of a UEM, except as a coarse measure of how easily problems may be discovered with a particular method. However, several possibilities for extending comparisons of UEMs beyond the problem counting approach exist. First, problem counting may form part of an analysis of what De Jong and Schellens (2000) called congruent validity, that is 'the similarities and differences between the problems found under different methods' (p. 244). However, to do so problem counting must be combined with some form of analysis of the content of the problems. This has been done in a number of studies, for example using severity ratings or classification of problems according to phase in the interaction cycle; some of the studies doing this predate 1999 (e.g. Jeffries *et al.* 1991, Karat *et al.* 1992). Among more recent studies, more comprehensive classification has been attempted. Sutcliffe *et al.* (2000) analysed problems using taxonomies of underlying causes of observed user problems, intended to facilitate discussion about possible design solutions. Fu and Salvendy (2002) classified problems according to the level of information processing they required. Their classification suggested that user testing is more effective than heuristic evaluation in finding usability problems that novice users encounter. Analyses as the above are interesting in pointing out some of the different qualities of problems identified with different UEMs.

Second, instead of counting usability problems, we can look at alternative dependent measures, including whether developers get ideas for improvements or whether the problems identified are corrected. Section 7 will discuss those measures related to design recommendations.

Third, akin to the ISO tripartition of usability into measures of effectiveness, efficiency, and satisfaction (ISO 1998), we may consider also evaluators' comments on and satisfaction with the UEMs under evaluation. For instance, de Angelli *et al.* (2003) supplemented counting of problems and problem categories with questionnaires on the evaluators' perception of the UEM they used and of the application under evaluation. Sutcliffe and Gault (2004) asked evaluators to assess on a scale from 1 to 7 the applicability of 12 heuristics for evaluation of virtual reality applications. As with all *ad hoc*

satisfaction measures, such data can hardly stand on their own, but may capture orthogonal dimensions to the effectiveness of a UEM.

4. Dogma No. 2: matching problem descriptions is straightforward

In most assessments of UEMs, the usability problems found by the evaluators are compared in order to identify similar problems. In addition, problems are often matched to a list of known usability problems to help assess the realness of the problems found. Sometimes problems that are considered similar are replaced with a general description of the problem, but most often the individual descriptions of the problems are kept for analysis.

Several of the comparisons of UEMs reviewed by Gray and Salzman (1998) used some kind of matching of usability problems. Nielsen (1992), for example, compared usability problems found with a telephone operated interface to a collection of known problems with that interface. Many studies appearing since the Gray and Salzman review also match problems; see, for instance, Andre *et al*. (2001) or Molich *et al*. (2004). Often, however, little or no information is given on how matching is performed. Nielsen (1992), for example, did not explain the procedure he used to match usability problems or the criteria for treating two problems as similar. Newer studies are similarly uninformative about the procedure used for matching. Mankoff *et al*. (2003), for instance, did not explain how known usability issues were related to problems identified using their new heuristics for usability inspection. While solid procedures for matching and definitions of what constitutes a match may have been used, they are not described in the paper. Thus, it seems that matching of usability problems is generally considered straightforward.

There are a number of reasons why this is not so. First, the lack of rigour in describing matching procedures opens the way for disparate interpretations of what constitutes a match. Lavery *et al*. (1997) illustrated the absence of clear descriptions of matching rules in a selection of studies. As mentioned above, studies appearing since Lavery *et al*.'s study are similarly brief about their matching procedures. In addition, Lavery *et al*. described shortcomings of basing matching on a specific evaluation method, for example when using the questions from cognitive walkthrough to match problems. They also listed problems associated with studies – such as Mack and Montaniz (1994) – that do contain descriptions of how problems were matched.

Second, matching is made difficult because of the brief, context-free descriptions that comprise most usability reports. The 46 problems described by John and Mashyna (1995, Appendix 1), for example, is on average about 28 words long; Keenan *et al*. (1999) showed three problems consisting of an average of 23 words. Hartson *et al*. (2001) suggested that 'because usability problems are usually written in an *ad hoc* manner, expressed in whatever terms seemed salient to the evaluator at the time the problem is observed, it is not unusual for two observers to write substantially different descriptions of the same problem' (p. 387). In general, matching of usability problems often proceeds with little information as a fragile, interpretive process, containing many of the uncertainties of interpreting short utterances in general (Frøkjær and Hornbæk 2002).

Third, matching may be done on both problem types and problem tokens. John and Mashyna (1997) provided examples taken from studies comparing evaluation methods at the problem type level, without data on problem tokens or where problem types cover disparate problem tokens. They argued that these studies find a large overlap between sets of problems, possibly leading to an overrating of the success of methods. Similarly, Connell and Hammond (1999) showed that counting by problem type rather than by problem token reduces the number of evaluators needed to find three-quarters of the usability problems in an interface. Thus, the level at which matching is done may impact evaluation results.

Fourth, a tradeoff exists between whether liberal or strict matching is used. On the one hand, too liberal matching leads to large overlaps between evaluation methods: when more problems are matched there will, on average, be more matches among problems and hence larger overlaps. On the other hand, too strict matching of problems may lead to relatively little overlap between evaluation techniques or evaluators. As an illustration, consider the low levels of agreement found in some studies of the evaluator effect (Hertzum and Jacobsen 2001). These levels of agreement are contingent upon the matching criteria used. For example, one study found an average overlap between evaluators of merely 9%, even though evaluators appeared to be in consensus when discussing the problems (Hertzum *et al*. 2002). The study by Hertzum *et al*. does not describe their criteria for matching in detail, but did have two evaluators perform and discuss the matching. Nevertheless, the use of overly strict matching criteria could be part of the reason for the apparent difference between the overlap data and the evaluators' opinions.

Fifth, an extensive psychological literature on the perception and assessment of similarity exists. This literature suggests a number of further difficulties for matching of usability problems. For instance, Tversky

(1977), in a pioneering study on similarity, suggested that similarity judgments may not be symmetrical and that they may be affected by diagnosticity. Given two usability problems, A and B, a lack of symmetry of similarity judgment implies that a different similarity is found when A is compared with B than when B is compared with A. In terms of Tversky's theory this could be explained by the salience of the problems being assessed: usability problem B could be a more commonly experienced problem leading us to assess similarity to it higher – the same way as subjects rate the number 103 to be more similar to 100 than the other way around. The principle of diagnosticity suggests that features of an object may be more or less salient according to how well they help classify an object, or, in Tversky's (1977) words 'the similarity of objects is modified by the manner in which they are classified' (p. 344). Thus, the matching of usability problems is likely to be affected by the matches made. Another problem is pointed out by Klein (1999), in a discussion of the work of Weitzenfeld (1984). Klein suggested that similarity does not make sense without some notion of purpose; that is, a notion of the use to which the similarity of objects are to be put. By analogy to the matching of usability problems, matching appears to have an unclear and unrealistic purpose, simply because it is not mirroring activities in real usability work. None of the studies surveyed attempts to create a realistic context for the matching or to provide matchers with a clear idea of why matching is done, apart, of course, from the rather obvious use for assessing evaluation methods.

A couple of attempts have been made to circumvent the above problems. One is to match usability problems to varying degrees. John and Matshyna (1997) distinguished a precise and a vague hit between usability problems; Connell *et al.* (2004) similarly distinguished a hit from a possible hit. In neither case, however, did the authors discuss the benefits of using multiple degrees of matching. While matching in degrees reduces part of the uncertainty of matching, it is not a general solution because it does not remove most of the difficulties discussed above.

Another suggestion has been to use structured reporting of usability problems to facilitate their matching (Lavery *et al.* 1997). The rationale behind this suggestion is that the format of usability problem reporting is crucial to successful matching of problems. Lavery *et al.* (1997) proposed a reporting format for usability reports that separates causes, breakdowns, outcomes and solutions. They showed examples of usability problems in the proposed format, but did not document whether it will improve matching of problems, or whether evaluators find the format too laborious.

Building upon the work of Lavery *et al.* (1997), Cockton *et al.* (2003b) showed how a extended reporting format made evaluators using heuristic evaluation make fewer predictions that were false positives, compared with a previous study using a simpler reporting format. The extended reporting format asked evaluators to report problem descriptions, likely/actual difficulties, the context of a problem, and assumed causes. In addition, evaluators reported how they discovered problems, for example if they scanned the system for problems and which heuristic they used to find a particular problem. The study is indicative only, because the use of the extended reporting format is compared only to a previous study, with many differences to the study in which the extended reporting format was being used. The study by Cockton *et al.* does not describe which aspects of the extended format work: one obvious hypothesis is that evaluators simply wrote longer problem descriptions, and thus gave more information for the matching process, or simply thought a little deeper about the problem. Cockton *et al.* (2004) provided further evidence for the impact of the structured reporting format. However, their studies appear not to be true experiments – in that no comparable control group was used – so no clear quantification of the benefits of structured reporting can be given. Nevertheless, structured reporting appears promising because it forces evaluators to be more careful in describing usability problems.

Connell *et al.* (2004) reported a qualitative analysis of problems with a ticket vending machine predicted by two analytic evaluation techniques and observed at the London Underground. They suggested describing usability problems in terms of their behavioural outcomes, even for analytical evaluation techniques. Though reporting behavioural outcomes is only a minor modification to problem reporting, it appears to facilitate matching.

Overall, a number of difficulties regarding structured reporting remains: the evidence that more reliable or better matching can result from structured problem reports need to be strengthened, and studies that claim an advantage for structured reporting can seldom pinpoint the factors behind that advantage. Research on how to report and match usability problems appears necessary.

Another suggestion for how to improve matching has been to use some kind of classification for describing usability problems (Keenan *et al.* 1999; Sutcliffe *et al.* 2000; Andre *et al.* 2001). The aim is to assist matching by supporting either the description of usability problems or the matching process. In the words of Hartson *et al.* (2001) '[t]he User Action Framework allows usability engineers to normalise

usability problem descriptions based on identifying the underlying usability problem types within a structured knowledge base of usability concepts and issues' (p. 405).

Studies have shown that it is possible, with little training, to learn to classify problems within at least some of these classification schemes. Andre *et al.* (2001) experimented with the User Action Framework (UAF), a comprehensive extension of Norman's Stages of Action model of human–computer interaction. The UAF contains several hundred nodes organised in a hierarchical structure, with top nodes based on Norman's model and deeper-level nodes being increasingly specific about a particular type of usability problem. Andre *et al.* showed that reliable classification was possible at the top levels of the User Action Framework (Cohen's kappa = 0.583). Though overall reliability was statistically significant, the agreement between evaluators was low compared with commonly agreed-upon requirements for agreement. Sutcliffe *et al.* (2000) described how Model-Misfit Analysis could be used to analyse the usability problems identified with two information retrieval products. Though their aim with the model is to provide explanations of the problems and identify solutions, an obvious use would be to match problems using their position in the taxonomies. Sutcliffe *et al.* provided no evidence that their model improved analyses over alternative approaches. In addition, the goal of making a classification of usability problems is not only to assist reliable classification, but more importantly to support description and matching of problems. This goal, however, still needs some empirical underpinning.

Finally, one idea would be to model matching more closely to how usability practitioners work with problems: this could give the matching a much clearer purpose. However, the literature does not contain solid empirical studies of how matching and prioritising is done in real-life settings; Section 6 discuss this idea further.

5. Dogma No. 3: usability evaluation proceeds as prescribed and directly identifies problems

A UEM functions in a way like any other method: it suggests a way of proceeding and some activities to be undertaken. Some UEMs are detailed on the procedure and the questions to ask (e.g. by providing detailed steps to walk through and sets of guidelines to consider), others are less extensive. Some studies of UEMs, however, appear to be based on the assumption that the methods under assessment prescribe the evaluation activity so strongly that decisions on part of the evaluator play a small role. Thereby, it is assumed that the evaluation process is orderly, proceeding as prescribed by the method, and that it directly leads to identification of problems by deliberate thinking about the questions to be asked. This assumption is seen in studies of both analytic and empirical UEMs. We consider this assumption a dogma, but compared with the earlier discussions in this study, this dogma does not show itself in an easily identifiable manner and the discussion below is therefore somewhat indirect.

In the studies reviewed by Gray and Salzman, it appears that at least two studies implicitly attribute most of the problem generation to the methods. Jeffries *et al.* (1991) asked evaluators to 'report problems they found even if the technique being used did not lead them to the problem, and to note how they found the problem' (p. 121). They classify evaluators' answers as to whether the problem was found 'via technique', 'side effect (e.g. when applying a guideline about screen layout, a problem with menu organisation might be noted)', or 'from prior experience with the system'. Jeffries *et al.* showed that evaluators using guidelines report many problems found as side effects (23%) and from prior experience (34%). Thus, the sources for problems differ between the UEMs being evaluated. For some reason, however, Jeffries *et al.* (1991) wrote that 'by definition, all of the problems found by heuristic evaluators were found by heuristic evaluation' (p. 121), apparently assuming that evaluators using heuristic evaluation only find problems through the heuristics. Nielsen (1992) made a more indirect assertion about the nature of the evaluation process in writing 'Since heuristic evaluation is based on judging interfaces according to established usability principles, one might expect that problems violating certain heuristics would be easier to find than others' (p. 379). Here Nielsen appears to assume that the discovery of problems is done directly through the heuristics, and that evaluators somehow will have more or less difficulty finding certain problems because those problems may be found by a particular heuristic. Of course, problems of a particular kind may be more or less hard to uncover, but the absence of the evaluator in the statement quoted (and his or her understanding of the heuristics) seems simplistic.

Similar examples may be found in later studies. Several studies (e.g. Hornbæk and Frøkjær 2004a, Huart *et al.* 2004) relate the heuristics and questions that evaluators report having used to uncover problems in an interface to counts of the problems found by those heuristics or questions. Apparently these studies assume (a) that heuristics and questions in this way can be directly used to uncover problems and (b) that evaluators' reports are valid; that is, when an evaluator writes that problem X has been found using heuristic Y, this is the case. Compare these

assumptions to the description of Hertzum and Jacobsen (2001):

> We conjecture that the heuristics principally serve as a source of inspiration in that they support the evaluator in looking over the interface several times while focusing on different aspects and relations of it. The quality of this stream of new aspects and relations is that it leads the evaluator to consider still new questions about the usability of the interface.

Here, evaluation appears a much more complex process that to a lesser extent is assumed to be directly driven by the evaluation method.

Another example of the dogma mentioned above may be found in the study by Cockton and Woolrych (2002). They argued that 'In our research on HE, predictions attributable to HE rattled around within a big box of predictions based on analysts' common sense' and 'Thus, it appeared that the analyst themselves rather than the heuristics they were supposedly applying provided the discovery resource' (p. 15). While Cockton and Woolrych go on to present some interesting data on heuristic evaluation, their critique of the evaluation process assumes that evaluators must use heuristics to discover problems. In essence ' ... it must be established that predictions are due to the UIM [usability inspection method, i.e. heuristic evaluation] and not to expert judgment or an analyst's intuitions' (Cockton and Woolrych 2001).

In contrast to the above remarks, data accumulate that suggest that the use of UEMs is much more complex. Studies of the evaluation process suggest that evaluators find problems in many ways, and not just as prescribed by the technique. Jacobsen and John (2000) reported a diary study of the use of cognitive walkthrough. Evaluators in that study identified problems even before they actually performed the evaluation proper. Hornbæk and Frøkjær (2004b) similarly showed how usability problems were discovered before, during, and after the evaluation itself: evaluators, for example, identified usability problems even on their first visit to the web site they were going to evaluate, a visit that often took place before evaluators had even looked at the evaluation technique they were going to use. Studies have also suggested that the extent to which evaluators are guided by heuristics is uncertain (Doubleday *et al.* 1997): 'It is difficult to know how far the heuristics guided the evaluators, even though they were given a detailed synopsis of the meaning of each heuristic' (p. 107). John and Mashyna (1997) found, similarly to the study by Jeffries quoted above, that only about two-thirds of the problems found during a cognitive walkthrough were actually that the evaluator attributes to the technique. Regarding empirical evaluation methods, several studies suggest that think-aloud practice differs markedly from prescriptions (e.g. Boren and Ramey 2000, Nørgaard and Hornbæk 2006). Thus, the role of UEMs in finding problems appears complex.

The above observations have several implications. First, we need to better understand the role of methods in usability evaluation. Studies that look at the process of conducting an evaluation are a good start (e.g. John and Mashyna 1997, Hornbæk and Frøkjær 2004b). Looking at such processes will teach us more about how problems are discovered and analysed, and should give important input on when the resources provided by an evaluation method are useful and when they are not.

Second, given that evaluators identify problems with a number of means and in many different stages of the evaluation activity, perhaps the need for support during evaluation should be rethought. Most of the studies reported since the review of Gray and Salzman have focused on validated new evaluation methods (e.g. Bastian *et al.* 1999, Sears and Hess 1999, Hornbæk and Frøkjær 2004a). Rather than devising new methods, we should perhaps develop support for evaluators while performing an evaluation and when reporting its results. One instance of an evaluation method where such support has been carefully developed and validated is evaluation by cognitive models such as KLM and GOMS. John *et al.* (2004), for example, showed how a tool for developing cognitive models made developing such models faster and the predictions more accurate compared with previous models. Because development of cognitive models is known to be difficult and time-consuming, explicit support for these activities seems a sensible and worthwhile development of that kind of evaluation method.

Third, evaluators' ability to perceive problems may be underestimated. Mack and Montaniz (1994) provided several observations on the role of method descriptions in the inspection process. From analysing usability problems, they suggested that usability inspectors sometimes simply self-report a problem they themselves experience. They also notice that '[o]ne objective of inspection heuristics or guidelines is to stimulate inspectors to notice things about the software interface that might lead, on further reflection, to identifying a potential problem' (p. 321) and that 'one important factor in generating inspection results is the extent to which inspectors can draw on their own experiences as users and as problem experiencers' (p. 323). These quotes, in concert with the one above by Hertzum and Jacobsen, suggest that we may further study what triggers the perception of a problem in the first place.

Fourth and finally, we need further insight into the evaluation process of experts, especially in analytical

evaluations. Jeffries *et al.* (1991) concluded their study by writing 'We believe that heuristic evaluation and usability testing draw much of their strength from the skilled UI professionals who use them. The importance of these people's knowledge and experience cannot be underestimated' (p. 124). Several studies have shown that expertise improves evaluators' performance. Nielsen (1992), for instance, showed that persons with knowledge on usability evaluation found about two to three times as many usability problems with a spoken language bank interface as novices did. Most of such studies, however, present only coarse details on the role of expertise in evaluation. We need to understand better how expert evaluators perform their evaluations so as to be able to develop tools and inspection methods that support evaluators who are beyond the stage of novices.

6. Dogma No. 4: the individual usability problem as the unit of analysis

In most assessments of UEMs, individual usability problems identified by the evaluators form the unit of analysis, for example by being counted, matched, or categorised. Sometimes individual problems of a certain type are considered together in the analysis, but this typically happens based on the individual problems. Only rarely are entire sets of problems analysed, let alone prioritised or synthesised, for example by listing the most critical issues to correct given the evaluation results. Thus it appears that most research studies consider as simple (or just ignore) the move from individual problems to a larger unit of analysis. This move, however, appears a necessary part of the practical uptake and use of the results of any usability evaluation.

Four of the studies reviewed by Gray and Salzman compared UEMs at the level of individual problems. Nielsen (1992), for example, looked at and counted separately every problem named by an evaluator. Nielsen's classification into major and minor problems was apparently based on personal judgment, and not on an analysis of the evaluators' results. Though group evaluations were reported in some studies (e.g. Desurvire *et al.* 1992, Karat *et al.* 1992), they also use the individual problems reported by groups as their unit of analysis, and do not include an overall view of the entire problem set. Studies appearing since 1999 also use only the individual usability problem as their unit of analysis (e.g. Bastian *et al.* 1999, Cockton and Woolrych 2001, Fu and Salvendy 2002, Mankoff *et al.* 2003, Law and Hvannberg 2004).

Treating the individual usability problem as the unit of analysis has many advantages from a methodological and statistical viewpoint. For example, it is easy to count problems because evaluators' reports typically mention them separately; more observations can be analysed statistically when individual problems are the focus; and individual problems are typically shorter and more distinct than recommendations at the level of a problem set.

Yet, there are at least three reasons why we need to supplement the individual usability problem as the unit of analysis. One of these reasons is that in practice considering the full problem set is crucial. Jeffries (1994) wrote eloquently that 'adding an additional step to the evaluation process, where the full set of problems are considered together, is extremely valuable' (p.288) and continued:

> This final step, which needs to be done by a trained individual, can ensure that the individual problem reports are not based on misunderstandings of the application, that they don't contradict each other, that the full impact of any trade-offs are taken into account, and that the recommendations are applied broadly (e.g. to all scrolling lists, not just to the one that the evaluator noticed) (p. 290).

Keenan *et al.* (1999) also addressed the issues of large scale analysis and prioritisation. They presented the Usability Problem Taxonomy (UPT), a classification system for usability problems. One of the ideas of the UPT is to support the prioritisation of usability problems and to allow high-level or global analysis of the problems. Similarly to the claim made here, Keenan *et al.* suggested that prioritisation and high-level analysis of usability problems are poorly supported by current methods for describing usability problems.

A second reason why we need to focus not only on individual problems is that individual problems may not be the only, or the most important, kind of feedback from usability evaluations. Some studies have looked at synthesised results as the outcome from usability evaluation; that is, at feedback from evaluations at a higher level of abstraction. Molich *et al.* (2004), for example, studied usability reports as the outcome of usability evaluation. Though the reports analysed in Molich's study included individual problems, some did include summaries or other syntheses of the problems found. It seems that more studies could investigate how to combine low- and high-level information from usability tests and how to design feedback so as to facilitate developers in understanding it and acting on it – for the latter purpose, individual descriptions of problems seem inappropriate.

A third reason why we should not focus only on individual problems is that studies of how an understanding of a set of usability problems develops seem important. This understanding process may happen for individual evaluators, independent raters of, say, the

severity of problems, and evaluators working in groups to perform an evaluation. Consider the situation where an independent rater scores a large number of usability problems with respect to severity; such scoring takes place in a number of the studies discussed above. Usually, this is done by going through the problems and scoring them individually. The assigned scores are then used to compare the severity of problems found by different UEMs. However, during this process the rater learns a lot about the entire problem set, about frequent and less frequent problems, and about where in an interface problems may also be seen. However, this is not captured with the individual problem as the level of analysis. Another example is found in studies of group evaluations. Mostly such studies either aggregate the results of individual evaluations to simulate a group (e.g. de Angelli *et al.* 2003) or, less often, report only the results of a group's evaluation (e.g. Karat *et al.* 1992). No study that we know of presents data on both individual evaluations and the result of meeting in a group to discuss those evaluations, which would help describe how evaluators' understanding develops. Likewise, studies that assess the utility of the results of usability evaluation, for example by asking developers (e.g. Hornbæk and Frøkjær 2004a), also stick to analysis of the practical utility at the level of individual problems.

The above studies suggest that we should look in more detail at how prioritisation of problems is done in practice. We know of only one study that comes close to doing that, namely Hassenzahl (2000). Such studies would help investigate if the processes of prioritisation in practice have any implications for how we should assess UEMs. Finally, studying how the understanding of usability problems develops when they are seen not as individual problems, but at some higher level, also warrants research.

7. Dogma No. 5: look at evaluation in isolation from design

UEMs offer procedures that help evaluators analyse the usability of a system. From that analysis, the evaluator forms various insights about the design. While these insights may take many forms, successful methods aim to help improve, or at least understand better, the usability of the system. To do so, methods must help identify aspects of the design that can be improved and ideally enable the evaluator to suggest solutions for how to do that. Thus, the true utility of methods lies in their ability to influence the design of the application being evaluated. This, however, is widely ignored in assessments of UEMs as most assessments look at evaluation in isolation from design.

All studies in the review by Gray and Salzman (1998) look at evaluation in isolation from design. They do not consider, for example, if usability problems may be corrected or which methods give the most useful input towards a redesign. As already argued, most studies appearing since 1998 conclude their analysis by counting and classifying usability problems. This step may be necessary before it can be assessed whether an UEM may impact design (although that is controversial), but it is definitively not sufficient. Only a few studies look at evaluation in the context of design or with some link to design (e.g. Hornbæk and Frøkjær 2004a).

Several problems are associated with assessing UEMs in isolation from their influence on design. Smith and Dunckley (2002), for example, argued that:

> A number of studies have been carried out to compare usability evaluation methods ... However all these have focused on evaluation methods themselves rather than on their influence on design. The effectiveness of the different methods has been compared in terms of the usability problems identified with an assumption of a direct link to design improvements (p. 832).

The assumption of a link to design improvements remains, however, an assumption. Cockton *et al.* (2003a), in a review of usability inspection techniques, similarly pointed out that 'Current UIMs [usability inspection methods] provide little, if any, support for the generation of recommendations for fixing designs to avoid predicted problems' (p. 1120). Wixon (2003) was even more harsh in arguing that '[t]he literature on usability evaluation is fundamentally flawed by its lack of relevance to applied usability work' (p. 34). He sees the focus on finding problems – rather than fixing them – as one of these flaws. In summary, identifying and listing problems remains an incomplete attainment of the goal of evaluation methods, both in assessing evaluation methods and in devising new methods. It should be noted, however, that we here are focusing on the formative use of evaluation, as all uses of evaluation methods might not require or aim at redesign. This would, for instance, be the case for summative evaluations.

One way to deal with this dogma is simply to conduct studies that go closer to the context in which the results of usability evaluations are being applied. Hartson *et al.* (2001) discussed the notion of downstream utility; that is, the 'usefulness in the usability engineering process after gathering usability problem data (e.g. quality of usability problem reports in helping practitioners find solutions)' (p. 389). Some work in this direction has aimed to develop metrics of impact. Sawyer *et al.* (1996) made an early influence by proposing impact ratio as a metric. The impact ratio is

simply the amount of usability problems that are addressed in the next version of the design of the system under evaluation over the number of usability problems found. Another metric is to ask developers about their perception of the utility of the usability problems in their development work (Hornbæk and Frøkjær 2004a, 2005). Still another metric is based on the effect of making design changes informed by problems identified in usability evaluations. John and Marks (1997) looked at design-change effectiveness; that is, the effectiveness (as measured by user testing) of changes derived from predicted usability problems.

Another approach is to find alternatives to usability problems as the main form of feedback from the evaluation process. Design proposals have been discussed as one such possibility. Hornbæk and Frøkjær (2005), for example, presented a study of how developers of a large web application assess statements of usability problems and statements of redesign proposals as input to their systems development. Developers found redesign proposals to have higher utility in their work than usability problems. In interviews they explained how redesign proposals gave them new ideas for tackling well-known problems. Redesign proposals were also seen as constructive and concrete input.

A natural objection to the use of redesign proposals is that sometimes evaluators may be able to characterise a problem, but not to point out a solution. It also appears that several differing redesign proposals may be created in response to the same problem description. Another objection is that redesign proposals should be formed not at the level of single problems, but as a coherent design encompassing all or most problems. Cockton *et al.* (2003a) argued that studies looking at the impact on design of UEMs have their own problems. They argued that in some cases it is not known whether the proposed redesigns are improvements; sometimes it is not known, either, from which usability problems the proposed changes were derived. We will not attempt to address these objections here. However, we contend that looking at evaluation in the context of design makes assessments of UEMs more realistic and more relevant to practical usability work.

8. Dogma No. 6: a single best UEM exists

Many assessments of UEMs use a simple setup. Few dependent measures are obtained, usually only one system is evaluated, each evaluator uses only one method and little or no data is provided about the kinds of problems identified. Within this setup, many studies conclude that one of the methods assessed is superior to the others. This conclusion is often based on the number of problems identified by each UEM and not on an in-depth analysis of the kinds of problems found or the utility of the UEM in evaluating the particular kind of system.

Most of the studies reviewed by Gray and Salzman (1998) appear *not* to have looked to find a single best UEM. They often include several applications (e.g. Nielsen 1992), analyse the content of the problems found (e.g. Karat *et al.* 1992), and use several dependent measures (e.g. Desurvire *et al.* 1992). However, none of them investigates evaluators that use combinations of methods. Only the study by Nielsen and Phillips (1993) stands out as being somewhat one-dimensional in focusing mainly on monetary gain from using a particular method.

Recent studies appear to be much more focused on finding a single best UEM, as judged from the measures they employ, the language in which they state their conclusions, the methods they examine, the analysis of the problems they perform, and so forth. Yet, this dogma is quite difficult to pinpoint with certainty in the studies and is in some cases only indicated by a general impression in the mind of the reader that the study reported is rather simplistic. Consider the following examples as indicators of this dogma. Most of the studies appearing since 1998 look at only one application. In many ways this is reasonable because of resource constraints. Some studies, however, reach conclusions that would have gained a lot from being replicated across different systems. Fu and Salvendy (2002), for example, described interesting data suggesting that heuristic and think-aloud evaluation were most likely to identify problems that affect users at a particular level of expertise. It would have been useful to know whether this was in part an artefact of a particular application being tested and it would at least have been relevant to get the authors' reflections on how their choice of application may have impacted their results. Not a single study among the 25 studies forming the basis of the present discussion has looked at evaluators using combinations of methods. As mentioned in the discussion of the problem counting dogma, many studies also rely mostly on a count of usability problems, rather than on characterising differences in the kinds of problems found (e.g. Mankoff *et al.* 2003, Kjeldskov and Stage 2004).

We consider the search for a single best UEM unfortunate for three reasons. First, practitioners appear to use a combination of methods, rather than relying on the results of just one (Borgholm and Madsen 1999, Gulliksen *et al.* 2004). In Molich *et al.* (2004), for example, three of nine teams performing usability tests of Hotmail chose to combine user testing with usability inspection. The dogma that an assessment of UEMs should identify a 'winner' is not

providing helpful information for the practice of combining UEMs.

Second, the choice of which UEM to use depends on what kind of information it is likely to give. The focus on problem counting and less on analysis of the contents of the problems found, then, does not give the most pertinent information for this choice. In some situations it might be desirable to use a method that is sub-optimal in terms of number of problems identified if that method usually identifies a particular kind of problem that is of particular relevance.

Third, contextual factors (such as system fidelity, evaluator–developer gap, phase in development cycle, kind of system, etc.) are all relevant to understanding the results obtained. While many studies manipulate such variables (or at least present details on them), some studies focus exclusively on finding a best method. In our opinion it is unlikely that such a method would work across all contexts. Finding a single best method in a particular context does not help practitioners design an evaluation in a different context.

A number of possibilities besides attempting to find a single best UEM exist. One approach could be to look not at individual techniques, but at combinations of techniques. While this appears a simple idea we only know of a few studies that have tried it. Frøkjær and Larusdottir (1999), for example, examined the effectiveness of performing heuristic evaluation, cognitive walkthrough or no evaluation before running a think-aloud test. Their results show that heuristic evaluation in combination with think-aloud testing detects more usability problems than other combinations of UEMs. However, much more systematic data on how UEMs may be combined seems to be needed.

Another direction to take research beyond comparisons based on the belief that a single best UEM exists consists in characterising differences in the kind of problems found. While many studies have done this for severity and for what functions are affected in the user interface, more detailed analysis of the contents of problems would provide valuable information. Recently, a few studies have tried to do this. In Fu and Salvendy (2002), for example, usability problems found by user testing and heuristic evaluation were mapped to a three-level performance classification. Aided by that mapping, the authors were able to show that heuristic evaluation finds fewer problems concerned with the knowledge-based performance than user testing does. Consequently, they argue that '[u]ser testing is more effective in discovering usability problems that novice users encounter' (p. 141). Such information appears useful as an aid to selecting a UEM in a particular setting. Other, related attempts to move beyond win/lose outcomes have used the notion of scoping (Cockton and Woolrych 2001, Blandford *et al.* 2004). The basic idea is to pursue qualitative descriptions of differences between UEMs in the problems they identify. For example, Blandford *et al.* (2004) compared the problems predicted by seven analytical UEMs. They found that the problems predicted could be divided into five categories and that each UEM identified only problems belong to one or two of those categories. Their detailed analysis can serve as a model for how to describe and reflect on qualitative differences in problems identified with different UEMs.

9. Dogma No. 7: usability problems are real

Many studies of usability evaluation appear to be based on a view – mostly implicit – of usability problems as something definite, unambiguous, and unchanging; that is, that usability problems are real in some sense of that word. This view influences how assessments of UEMs are designed and how results are interpreted. It implies, for example, that think-aloud testing is often treated as a gold standard against which to compare other methods. It is also closely related to the idea that a fixed number of usability problems exists in an interface. Below we discuss some alternative views of usability problems, arguing that the assumption that usability problems are real may not always be conducive for valid and informative assessments of UEMs.

This dogma is more of a guiding principle than an easily identifiable practice. In the studies reviewed by Gray and Salzman, only weak indications of this dogma may be found. Nielsen (1992), for example, used a set of known usability problems as benchmark. However, he failed to explain why those problems should be considered more real than other problems. As mentioned earlier, Karat *et al.* (1992) and Jeffries *et al.* (1991) matched problems. Below we argue that such matching is related to the view that usability problems are real, because matching takes very literally the problem descriptions and, occasionally, aims to infer underlying problems.

In studies appearing since the Gray and Salzman review, the belief that usability problems are real appears widespread. A number of studies, for example, rely on a notion of known problems; that is, usability problems that are supposed to be more real than other descriptions of problems (e.g. Mankoff *et al.* 2003). Other studies assume in one way or the other that a fixed number of usability problems exists in an interface (e.g. Mankoff *et al.* 2003, Law and Hvannberg 2004). Finally, almost all studies use matching of usability problems, with the assumption that it makes sense to compare problems and assess their likeness.

The belief that usability problems are real has a number of implications for doing assessments of UEMs that appears suboptimal. First, this dogma is very much related to expectations of what matching can do for assessments of usability problems; this was already touched upon in the earlier discussion of matching (see Section 4). An implicit assumption in most procedures for matching is that a notion of similar problems makes sense. Hartson *et al*. (2001), for example, wrote that ' ... to perform set operations on usability problem sets, one needs the ability to determine when two different usability problem descriptions are referring to *the same underlying problem*' (p. 387, my italics). Understanding the relation between descriptions of usability problems and underlying problems appears to be part of the reason for many of the methodological difficulties facing studies that assess UEMs. Despite the simple notion of similar problems, we still do not have a clear and convincing procedure for assessing whether two problems are similar.

The dogma that usability problems are real is also related to the belief that any user interface contains an exact, fixed number of usability problems that we may model and predict. As a consequence, we may try estimating how many evaluators or users are needed to detect a particular percentage of the problems in the interface. However, this assumption appears doubtful. For any authoritative list of usability problems, how can we ever be sure that we have not missed any problems? Moreover, general models of the number of users needed to detect a certain percentage of problems have been attacked (Spool and Shroeder 2001, Woolrych and Cockton 2001). It is claimed that such models underestimate the influence of user and task variability, and thus provide predictions of limited utility. In a few cases, studies infuse usability problems into an interface or do particular forms of testing that do not depend on the abovementioned belief that a particular number of problems exists in an interface.

This dogma also seems to be related to the belief that descriptions of usability problems uncovered by think-aloud testing are truer than problems found in other ways, leading to the insistence on usability testing as the infallible benchmark of all UEMs. This view fails to take into account that results of think-aloud studies are shaped by the tasks users do, interpretations on the part of the evaluator, individual differences among users, and so forth. It has been shown, for example, that even think-aloud studies are influenced by the evaluator effect (Hertzum and Jacobsen 2001). Further, Chi (1997) cited studies in which thinking aloud improved performance, possibly leading to an underestimation of usability problems in an interface. Some studies of UEMs have recognised that usability problems found by think-aloud testing may not be as special as usually assumed. Molich *et al*. (2004), for example, concluded their study of how different organisations conduct usability evaluations by stating 'Usability testing by itself can't develop a comprehensive list of defects. Use an appropriate mix of methods' (p. 74).

Let us turn to some alternatives to the view that usability problems are real, and the implications for assessments of UEMs. One of these views is that it is simply not possible (or at least not fruitful) to talk about same problems, underlying problem, correct match of problems, or the like. Instead, descriptions of usability problems should be seen merely as incomplete expressions of observations and inferences from usability evaluations, which may be useful in some practical activities such as improving an interface. We should then look to the implications of usability-problem descriptions, in particular whether they will help developers and designers change the system that was evaluated. Among other things, this view should lead to much greater focus on how results from usability evaluations are used.

A second view is that usability evaluation is really idea generation. The basic focus here is not whether or not usability problems are real. Rather the aim is to give system developers ideas about how to tackle the problems they are facing in their systems development. This view implies that we should focus on novelty and surprise in usability test results, and that ideas for key design problems should be valued over information of minor importance to the design and development group's current concerns. Especially the key concern becomes whether a UEM helps generate design ideas (see Tohidi *et al*. 2006).

It should be noted that the discussion above is related to a debate that has gone on for decades, in particular in the philosophy of science (Kukla 1998), about whether various phenomena in science can be said to be real. It is outside the scope of this study to review this debate or to draw analogies between it and the point raised here. We simply note that this kind of discussion is not unique, and that the insistence that usability problems are real may be just one viable position among many. We are not attempting to settle the discussion, only to begin it. Some of the alternative views sketched above might be useful in generating new directions for how to assess UEMs. However, fleshing out the consequences of those views remains an open task.

10. Conclusion

Methods for evaluating the usability of a computer application are a key contribution of human–computer

interaction research. However, the assessment of the relative effectiveness of different UEMs remains a daunting task. We have discussed seven assumptions in recent assessments of UEMs that are rarely questioned, but seem dubious. In particular we described a focus on counts of problems, a lack of attention to procedures for matching problems, unrealistic assumptions about the role played by method prescriptions in evaluations, an exclusive focus on individual problems, a lack of focus on how problems are taken up in design, a belief that a single best UEM exists, and an assumption that usability problems are real.

We have argued that these assumptions are problematic and presented alternatives to each of them. In particular, we have argued for more careful analysis of the contents of usability problems, for detailed studies of the process of usability evaluation, for paying attention to real-life prioritisation and working with usability problems, and for looking at how usability evaluation influences design.

In sum, this and other reviews have identified many methodological challenges in assessing UEMs. Yet, many recent studies are still guided by the dogmas outlined in the present study. We suggest that moving beyond those dogmas will strengthen usability research and enable better advice to be passed on to practitioners.

Acknowledgements

Erik Frøkjær has contributed immensely to this study by discussing with me most of its content. I wish to thank the USE project group for input on a draft of this study. The work was supported by the Danish Research Councils (grant number 2106-04-0022).

References

Andre, T.S., *et al*., 2001. The user action framework: a reliable foundation for usability engineering support tools. *International Journal of Human-Computer Studies*, 54 (1), 107–136.

Bastian, J.M.C., Scapin, D.L., and Leulier, C., 1999. The ergonomic criteria and the ISO/DIS 9241-10 dialogue principles: a pilot comparison in an evaluation task. *Interacting with Computers*, 11, 299–322.

Blandford, A., *et al*., 2004. Scoping analytical usability evaluation techniques: a case study. *CASSM Working paper*, available from http://www.uclic.ucl.ac.uk/annb/CASSM/CASSMpubls.html

Boren, T. and Ramey, J., 2000. Thinking aloud: reconciling theory and practice. *IEEE Transactions on Professional Communication*, 43 (3), 261–277.

Borgholm, T. and Madsen, K.H., 1999. Cooperative usability practices. *Communications of the ACM*, 42 (5), 91–97.

Chi, M.T.H., 1997. Quantifying qualitative analyses of verbal data: a practical guide. *The Journal of the Learning Sciences*, 6 (3), 271–315.

Cockton, G., Lavery, D., and Woolrych, A., 2003a. Inspection-based evaluations. *In*: J.A. Jacko and A. Sears, eds. *The Human-Computer Interaction Handbook*. Mahwah, NJ: Lawrence Erlbaum Associates, 1118–1138.

Cockton, G. and Woolrych, A., 2001. Understanding inspection methods: lessons from an assessment of heuristic evaluation. *In*: A. Blandford, J. Vanderdonckt, and P.D. Gray, eds. *Proceedings of people and computers XV: joint proceedings of HCI 2001 and IHM 2001*. Berlin: Springer-Verlag, 171–192.

Cockton, G. and Woolrych, A., 2002. Sale must end: should discount methods be cleared off HCI's shelves? *Interactions*, 9 (5), 13–18.

Cockton, G., *et al*., 2003b. Changing analysts' tunes: the surprising impact of a new instrument for usability inspection method assessment. *In*: *Proceedings of people and computers XVII: designing for society*. Springer Verlag, 145–162.

Cockton, G., Woolrych, A., and Hindmarch, M., 2004. Reconditioned merchandise: extended structured report formats in usability inspection. *In*: *Extended abstracts of ACM Conference on Human–Computer Interaction*. New York: ACM Press, 1433–1436.

Connell, I., Blandford, A., and Green, T., 2004. CASSM and cognitive walkthrough: usability issues with ticket wending machines. *Behaviour & Information Technology*, 23 (5), 307–320.

Connell, I.W. and Hammond, N.V., 1999. Comparing usability evaluation principles with heuristics: problem instances vs. problem types. *In*: *Proceedings of IFIP TC.13 international conference on human-computer interaction*, Amsterdam: IOS Press, 621–629.

de Angelli, A., *et al*., 2003. On the advantages of a systematic inspection for evaluating hypermedia usability. *International Journal of Human-Computer Interaction*, 15 (3), 315–335.

De Jong, M. and Schellens, P.J., 2000. Towards a document evaluation methodology: what does research tell us about the validity and reliability of evaluation methods? *IEEE Transactions on Professional Communication*, 43 (3), 242–260.

Desurvire, H.W., Kondziela, J.M., and Atwood, M.E., 1992. What is gained and lost when using evaluation methods other than empirical testing. *In*: *Proceedings of people and computers VII*, Cambridge University Press, 89–102.

Doubleday, A., Ryan, A., and Sutcliffe, A., 1997. A comparison of usability techniques for evaluating design. *In*: *Proceedings of ACM conference on designing interactive systems*. New York: ACM Press, 101–110.

Frøkjær, E. and Hornbæk, K., 2002. Metaphors of human thinking in HCI: habit, stream of thought, awareness, utterance, and knowing. *In*: *Proceedings of HF2002/OzCHI 2002*.

Frøkjær, E. and Larusdottir, M., 1999. Predicting of usability: comparing method combinations. *In*: *Proceedings of 10th international conference of the information resources management association*.

Fu, L. and Salvendy, G., 2002. Effectiveness of user-testing and heuristic evaluation as a function of performance classification. *Behaviour & Information Technology*, 21 (2), 137–143.

Gray, W.D. and Salzman, M.C., 1998. Damaged merchandise? A review of experiments that compare usability evaluation methods. *Human-Computer Interaction*, 13 (3), 203–261.

Grudin, J., 1991. Interactive systems: bridging the gaps between developers and users. *IEEE Computer*, 24 (4), 59–69.

Gulliksen, J., *et al*., 2004. Making a difference – a survey of the usability profession in Sweden. *In*: *Proceedings of proceedings of the third nordic conference on human-computer interaction*. New York: ACM Press, 207–215.

Hartson, H.R., Andre, T.S., and Williges, R.C., 2001. Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction*, 13 (4), 373–410.

Hassenzahl, M., 2000. Prioritising usability problems: data-driven and judgement-driven severity estimates. *Behaviour & Information Technology*, 19, 29–42.

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

Hertzum, M. and Jacobsen, N.E., 2001. The evaluator effect: a chilling fact about usability evaluation methods. *International Journal of Human-Computer Interaction*, 13, 421–443.

Hertzum, M., Jacobsen, N.E., and Molich, R., 2002. Usability inspections by groups of specialist: perceived agreement in spite of disparate observations. *In*: *Proceedings of extended abstracts of the ACM conference on human factors in computing*, New York: ACM Press, 662–663.

Hornbæk, K. and Frøkjær, E., 2004a. Usability inspection by metaphors of human thinking compared to heuristic evaluation. *International Journal of Human-Computer Interaction*, 17 (3), 357–374.

Hornbæk, K. and Frøkjær, E., 2004b. Two psychology-based usability inspection techniques studied in a diary experiment. *In*: *3rd Nordic conference on human-computer interaction (Nordichi 2004)*, New York: ACM Press, 3–12.

Hornbæk, K. and Frøkjær, E., 2005. Comparing usability problems and redesign proposals as input to practical systems development. *In*: *Proceedings of ACM conference on human factors in computing systems*. New York: ACM Press, 391–400.

Huart, J., Kolski, C., and Sagar, M., 2004. Evaluation of multimedia applications using inspection methods: the cognitive walkthrough case. *Interacting with Computers*, 16, 183–215.

ISO, 1998. Ergonomic requirements for office work with visual display terminals (VDTs)-part 11: guidance on usability.

Jacobsen, N.E. and John, B.E., 2000. *Two case studies in using cognitive walkthroughs for interface evaluation* (Carnegie Mellon Tech Rep. No. CMU-CS-00-132).

Jeffries, R., 1994. Usability problem reports: helping evaluators communicate effectively with developers. *In*: J. Nielsen and R.L. Mack, eds. *Usability Inspection Methods*. New York: Wiley, 273–294.

Jeffries, R., *et al*., 1991. User interface evaluation in the real world: a comparison of four techniques. *In*: *Proceedings of ACM conference on human factors in computing*, New York: ACM Press, 119–124.

John, B., *et al*., 2004. Predictive human performance modelling made easy. *In*: *Proceedings of ACM conference on human factors in computing systems*, New York: ACM Press, 455–462.

John, B. and Mashyna, M.M., 1995. Evaluating a multimedia authoring tool with cognitive walkthrough and think-aloud user studies (Rep. No. CMU-HCII-95-105/CMU-CS-95-189).

John, B.E. and Mashyna, M.M., 1997. Evaluating a multimedia authoring tool. *Journal of the American Society of Information Science*, 48 (9), 1004–1022.

John, B.E. and Marks, S.J., 1997. Tracking the effectiveness of usability evaluation methods. *Behaviour & Information Technology*, 16 (4/5), 188–202.

Karat, C.-M., Campbell, R., and Fiegel, T., 1992. Comparison of empirical testing and walkthrough methods in usability interface evaluation. *In*: *Proceedings of ACM conference on human factors in computing*. New York: ACM Press, 397–404.

Keenan, S.L., *et al*., 1999. The usability problem taxonomy: a framework for classification and analysis. *Empirical Software Engineering*, 4 (1), 71–104.

Kjeldskov, J. and Skov, M., 2003. Creating realistic laboratory settings: comparative studies of three think-aloud usability evaluations of a mobile system. *In*: *Proceedings of 9th IFIP TC13 international conference on human-computer interaction*. Amsterdam: IOS Press, 663–670.

Kjeldskov, J. and Stage, J., 2004. New techniques for usability evaluation of mobile systems. *International Journal of Human-Computer Studies*, 60, 599–620.

Klein, G., 1999. *Sources of power: how people make decisions*. Cambridge, MA: MIT Press.

Kukla, A., 1998. *Studies in scientific realism*. Oxford University Press.

Lavery, D., Cockton, G., and Atkinson, M.P., 1997. Comparison of evaluation methods using structured usability problem reports. *Behaviour & Information Technology*, 16 (4/5), 246–266.

Law, E. and Hvannberg, E., 2004. Analysis of combinatorial user effect in international usability tests. *In*: *Proceedings of ACM conference on human factors in computing systems*, New York: ACM Press, 9–16.

Mack, R.L. and Montaniz, F., 1994. Observing, predicting, and analyzing usability problems. *In*: J. Nielsen and R.L. Mack, eds. *Usability Inspection Methods*. Wiley, 295–339.

Mankoff, J., *et al*., 2003. Heuristic evaluation of ambient displays. *CHI letters, CHI 2003, ACM conference on human factors in computing systems*, 5 (1), 169–176.

Markopoulos, P. and Bekker, M., 2003. On the assessment of usability testing methods for children. *Interacting with Computers*, 15, 227–243.

Molich, R., *et al*., 2004. Comparitive usability evaluation. *Behaviour & Information Technology*, 23 (1), 65–74.

Nielsen, J., 1992. Finding usability problems through heuristic evaluation. *In*: P. Baursfield, J. Bennet, and G. Lynch, eds. *Proceedings of ACM CHI'92 conference on human factors in computing systems*. New York: ACM Press, 373–380.

Nielsen, J. and Phillips, V., 1993. Estimating the relative usability of two interfaces: heuristic, formal, and empirical methods compared. *In*: *Proceedings of ACM conference in human factors in computing systems*, 214–221.

Nørgaard, M. and Hornbæk, K., 2006. What do usability evaluators do in practice? An explorative study of think-aloud testing. *In: ACM symposium on designing interactive systems, DIS2006*. New York: ACM Press, 209–218.

Sawyer, P., Flanders, A., and Wixon, D., 1996. Making a difference – the impact of inspections. *In*: M.J. Tauber, ed. *Proceedings of ACM conference on human factors in computing*. New York: ACM Press, 376–382.

Sears, A. and Hess, D., 1999. Cognitive walkthroughs: understanding the effect of task description detail on evaluator performance. *International Journal of Human-Computer Interaction*, 11, 185–200.

Smith, A. and Dunckley, L., 2002. Prototype evaluation and redesign: structuring the design space through contextual techniques. *Interacting with Computers*, 14, 821–843.

Spool, J. and Shroeder, W., 2001. Testing web sites: five users is nowhere near enough. *In*: *Proceedings of CHI '01 extended abstracts on human factors in computing systems*, New York: ACM Press, 285–286.

Sutcliffe, A. and Gault, B., 2004. Heuristic evaluation of virual reality applications. *Interacting with Computers*, 16, 831–849.

Sutcliffe, A., *et al*., 2000. Model mismatch analysis: towards a deeper explanation of users' usability problems. *Behaviour & Information Technology*, 19 (1), 43–55.

Tohidi, M., *et al*., 2006. Getting the right design and the design right. *In*: *Proceedings of ACM conference on human factors in computing*. New York: ACM Press, 1243–1252.

Tversky, A., 1977. Features of similarity. *Psychological Review*, 84 (4), 327–352.

Weitzenfeld, J., 1984. Valid reasoning by analogy. *Philosophy of Science*, 51, 137–149.

Wixon, D., 2003. Evaluating usability methods: why the current literature fails the practitioner. *Interactions*, 10 (4), 29–34.

Woolrych, A. and Cockton, G., 2001. Why and when five test users aren't enough. *In*: J. Vanderdonckt, A. Blandford, and A. Derycke, eds. *Proceedings of IHM-HCI 2001 conference: Volume 2*. Toulouse: Cépadèus Éditions, 105–108.