Int. J. Human-Computer Studies (2001) **54,** 107–136
doi:10.1006/ijhc.2000.0441
Available online at http://www.idealibrary.com on IDEAL®

The user action framework: a reliable foundation for usability engineering support tools

TERENCE S. ANDRE
Air Force Research Laboratory, 6030 South Kent Street, Mesa, AZ 85212, USA

H. REX HARTSON
Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

STEVEN M. BELZ AND FAITH A. MCCREARY
Department of Industrial and Systems Engineering, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061, USA

(Received 24 March 2000 and accepted in revised form 8 September 2000)

Although various methods exist for performing usability evaluation, they lack a systematic framework for guiding and structuring the assessment and reporting activities. Consequently, analysis and reporting of usability data are *ad hoc* and do not live up to their potential in cost effectiveness, and usability engineering support tools are not well integrated. We developed the User Action Framework, a structured knowledge base of usability concepts and issues, as a framework on which to build a broad suite of usability engineering support tools. The User Action Framework helps to guide the development of each tool and to integrate the set of tools in the practitioner's working environment. An important characteristic of the User Action Framework is its own reliability in term of consistent use by practitioners. Consistent understanding and reporting of the underlying causes of usability problems are requirements for cost-effective analysis and redesign. Thus, high reliability in terms of agreement by users on what the User Action Framework means and how it is used is essential for its role as a common foundation for the tools. Here we describe how we achieved high reliability in the User Action Framework, and we support the claim with strongly positive results of a summative reliability study conducted to measure agreement among 10 usability experts in classifying 15 different usability problems. Reliability data from the User Action Framework are also compared to data collected from nine of the same usability experts using a classic heuristic evaluation technique. © 2001 Academic Press

KEYWORDS: user action framework; usability evaluation; tool support; evaluation techniques.

1. Support for usability engineering activities

1.1. THE NEED FOR STRUCTURED USABILITY TOOL SUPPORT

Because of a growing awareness of its importance, organizations are expending ever-increasing resources for "doing usability"—building enviable usability laboratories, training developers in usability engineering methods (Hix & Hartson, 1993), and

1071-5819/01/010107 + 30 $35.00/0 © 2001 Academic Press

conducting usability evaluations. Usability practitioners now have effective (though often expensive) methods to guide them in gathering raw observational usability data via usability testing and usability inspection. But frequently they are not achieving acceptable *returns on this investment* in usability development effort.

The process is mainly thwarted by information losses, losses that occur because of the lack of a framework for organizing activities and reporting the results. The result is poor-quality usability problem reporting. Having reviewed hundreds of usability problem descriptions from real-world usability laboratories (e.g. Keenan, 1996), and having consulted in numerous real-world usability development environments, we know that evaluators usually record what they believe salient about an observed usability problem, based on what they notice at the time. Even though evaluators often use standardized report forms to ensure inclusion of contextual information, the resulting problem descriptions are more often than not inconsistent, vague and incomplete. These *ad hoc* laundry lists of raw usability problems require much of the content to be conveyed by memory and word of mouth to designers responsible for fixing the problems.

Unfortunately, further analysis and redesign are often performed after a delay in time, by different people and sometimes at a different location, causing information losses that leave developers to interpret the reports and reconstruct the missing usability information. The value and overall utility of the expensive usability data is significantly reduced within the process.

This poor communication of usability data in the iterative process is directly due to the lack of a framework to guide complete and accurate usability problem reporting and the other iterative usability development activities. Additionally, few software tools exist to support usability problem analysis, classification and reporting, and the few such tools that do exist are mostly *ad hoc* and have not been proven effective. Very few projects, for example, have employed usability database facilities to track of the life history of each usability problem and to compare trends across projects, allowing practitioners to build on lessons learned. Practitioners in these projects must continually reinvent usability problem analysis and design. Further, no community memory exists to leverage all our usability efforts by growing an industry-wide usability knowledge base, which would advance the science to everyone's benefit.

Finally, many existing user interaction development methods (e.g. inspection heuristics, design guidelines) are limited to graphical user interfaces (GUIs). The expansion of applications to additional new interaction styles, such as those found in virtual environments, Web-based applications, pen-based interaction, and voice I/O has meant that GUI-specific guidelines, methods and tools are often not applicable.

In sum, the practice of usability engineering can benefit from the following.

- A reliable framework to guide interaction development activities and to facilitate high-quality usability problem reporting.
- Integrated sets of tools to support interaction development activities downstream from usability testing, including for usability problem classification and usability data maintenance.
- Tools, including usability inspection tools, that can adapt easily to new interaction styles beyond GUIs.

The focus of this paper is the first item, the framework and its reliability of usage, but we will also indicate how the framework is to be applied in the tools and extended beyond GUI interaction styles.

1.2. INTRODUCING THE USER ACTION FRAMEWORK

Through iterative work on classification frameworks at Virginia Tech, we have developed a structured knowledge base of usability concepts, called the User Action Framework, that addresses the above problems by being as follows.

- The framework needed to guide interaction development activities.
- Designed specifically to facilitate high-quality usability problem reporting.
- A foundation on which a range of usability engineering support tools can be integrated.
- Built on a very general model of how humans interact with machines, affording easy adaptation to new interaction styles beyond GUIs.

Figure 1 illustrates our vision of how the User Action Framework integrates software development tools (i.e. Usability Design Guide, Usability Problem Inspector, Usability Problem Classifier and Usability Problem Database) to support specific parts of interaction design life cycle activities. Two of the support tools, the Usability Problem Inspector and the Usability Problem Classifier, are currently in the formative evaluation stage, while the Usability Design Guide and the Usability Problem Database tools have yet to be developed.

The Usability Design Guide tool at the left provides guidelines-based help for interaction designs, which are then (moving to the right) subjected to formative evaluation via either traditional usability lab-based testing (Hix & Hartson, 1993) or by usability inspection methods (Nielsen & Mack, 1994) such as the Usability Problem Inspector. Usability problems uncovered by evaluation are classified by type in the Usability

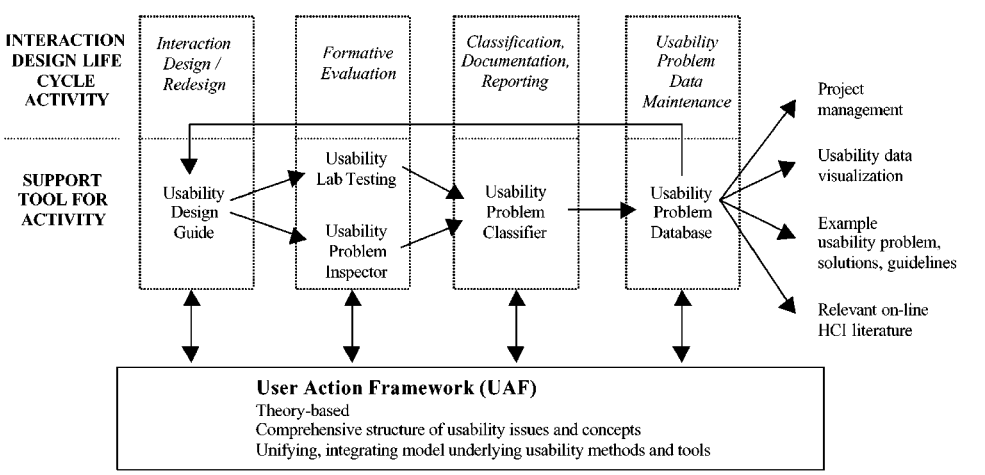

FIGURE 1. Scope of Usability Tool Integration provided by the User Action Framework.

Problem Classifier and entered into the Usability Problem Database as part of the development life cycle record.

The usability support tools are integrated by sharing User Action Framework content and structure as a common underlying framework. Each tool thus locates a given usability situation in the same place within the User Action Framework structure. Usability information flows laterally from tool to tool in the development process, as envisioned in Figure 1. For example, when the developer uses the Usability Problem Inspector to find a specific usability problem, that problem will be located within the usability problem space of the User Action Framework as it is implemented within the Usability Problem Inspector tool. To document the usability problem for the project, the developer follows a lateral link to the Usability Problem Database tool, arriving in the same location within the same User Action Framework structure, without any need for searching.

1.2.1. Importance of classification. The essence of the User Action Framework is an organizing framework of usability concepts, issues, principles and guidelines. The organization of the User Action Framework facilitates classification by providing structured, unique identification paths for the usability concepts and issues in the knowledge base. Classification plays an important role in usability engineering because classification aids description and good description is essential to high-quality problem reporting. Poor-quality reporting can lead to misdirected solutions and wasted resources, but classification helps accurate, complete and consistent problem reporting. Usability problems can look similar on surface but have different underlying causes and vice versa, but classification helps extract and isolate individual usability problems from observed usability situations (Cockton & Lavery, 1999), and classification helps pinpoint the underlying essence of problem causes. Given equal design processes, the input of higher quality usability problem reports, to include accurate and reliable descriptions, should lead to higher quality design solutions.

Because we believe classification to be an important part of the usability reporting process, classification is an inherent characteristic within the User Action Framework. Classification serves description in usability, much as it does in any other scientific area having structured frameworks for description (e.g. classification taxonomies of biology). For classification, the User Action Framework can be viewed as a decision structure within which classifiers make systematic decisions at nodes in each level of the structure, each node establishing a usability attribute of an observed usability problem. Classification determines a sequence of usability attributes accumulated, one per node in the full classification path within the hierarchically structured knowledge base. This sequence of specific usability attributes constitutes a descriptive "encoding" in a kind of "standard" usability language, locating the problem in a structured usability problem/design space. The net effect of these encodings is more complete and precise problem descriptions and provably higher value usability data within problem reports. If this classification process is reliable in addition, it means that the same complete and precise problem descriptions will be produced consistently and independently of the individual classifier.

Finally, classification of usability problems by type is not only valuable within the usability development process, but is also necessary for characterizing the strengths and

weaknesses of usability evaluation methods within usability evaluation method comparison studies (John & Marks, 1997; Lavery, Cockton & Atkinson, 1997; Gray & Salzman, 1998; Cockton & Lavery, 1999; Keenan, Hartson, Kafura & Schulman, 1999;). The User Action Framework answers all these needs for structured, complete, precise and reliable usability problem classification.

1.2.2. Importance of reliability. Usability engineering support tools offer the most value to interaction development groups if they are used consistently and predictably, from practitioner to practitioner. Without this kind of usage reliability, one evaluator using a usability tool can obtain one result and another a different result, and the quality of the usability data for the project will depend on the individual using the tools. One such example of this kind of variation is seen in the popular heuristic evaluation technique developed by Nielsen and Molich (1990) in the early 1990s. Heuristics, intended as a cheap, fast and easy to use method for inspection of user interfaces, do not provide a structured framework to separate out the fine differences between various usability problems (Dutt, Johnson & Johnson, 1994). Thus, some researchers (e.g. Jeffries, Miller, Wharton & Uyeda, 1991; Doubleday, Ryan, Springett & Sutcliffe, 1997) have noted that results from heuristic evaluations can lead to problem identification that is not distinct in terms of separating one problem description from another. Heuristics are often too general for detailed analysis, with each heuristic covering a broad range of usability factors (Sears, 1997). The overlaps among categories and gaps between them make it difficult to obtain reliable results from methods based on general lists of usability heuristics. Such overlaps and gaps in the categories can contribute to mis-classification; potentially influencing the process for focusing on a specific design solution.

We developed the User Action Framework on the premise that reliable tool usage depends on a consistent shared understanding of an underlying framework and how it is applied. Because the purpose of the structure is to provide quick and reliable access to the content, we consider reliability of its structure traversal the most important criterion for evaluating the User Action Framework as an underlying framework for tools.

1.3. RELATED WORK

1.3.1. Model-based framework. As explained in Section 2, we built the User Action Framework on a structure from Norman's (1986) theory of action model. Our work is not the first to use Norman's model as a basis for usability inspection, classification and analysis. Several other researchers (e.g. Cuomo, 1994; Lim, Benbasat & Todd, 1996; Sutcliffe, Ryan, Springett & Doubleday, 1996; Rizzo, Marchigiani & Andreadis, 1997; Garzotto, Matera & Paolini, 1998) have used Norman's model in various ways and found it helpful for communicating information about usability problems, identifying frequently occurring problem types and providing guidance for diagnosis of usability problems. In the work perhaps most similar to ours, Cuomo and Bowen (1992) used Norman's theory of action model with some success to assess the usability of graphical, direct-manipulation interfaces. Cuomo and Bowen concluded that the model showed promise, especially for problem classification in the usability testing environment, but

that more work was needed to modify Norman's theory of action model for use as part of an inspection technique.

1.3.2. Classification approaches. Classification of usability problems by their underlying characteristics directly benefits analysis. Once the salient attributes of a usability problem are identified, the nature of the problem can be communicated and solutions from previous problems similar in nature can be considered in the current context. It is our working hypothesis that a hierarchically structured framework of usability attributes helps to minimize individual differences in reporting. Such a framework provides practitioners with a standardized process for developing usability problem descriptions, which (1) are complete in terms of the attributes applicable to a problem type, and (2) distinguish a problem of one type from a problem of another.

Design guidelines (Mayhew, 1992; Shneiderman, 1998) and heuristics (Nielsen & Molich, 1990) offer a basis for high-level classification of usability problems, but little of this kind of classification is done in practice. Although in theory guidelines and heuristics (Nielsen & Molich, 1990) could provide a basis for high-level classification of usability problems, most of the related literature casts them as support for design and for identifying usability problems. In defining heuristics, Nielsen did some classifying and aggregating of usability problems found by several evaluators (Nielsen & Molich, 1990). Additionally, Brooks (1994) and Nielsen (1993) explored specific classification patterns based on their source and location in the human–computer dialog. This approach to classification, though, relies more on characteristics of how or where a usability problem is incorporated into the user interface than on characteristics of the usability problem itself. We believe a classification scheme for usability problems based upon the type of problem in terms of its cause within the interaction cycle will better aid developers in documenting and reporting usability problems and in finding strategies for addressing those that are similar in nature.

Butler (as reported in Nielsen, 1994*c*) classified problems into categories that are then rated for severity. Simple schemes are used for classifying problems by severity or importance (Nielsen, 1993; Desurvire, 1994; Rubin, 1994). Coupled with a cost/benefit assessment, this scheme is useful in prioritizing the order in which to address the problems. However, different evaluators using the same classification scheme based on severity may rate two instances of the same usability problem very differently, depending on their effects upon the users. While severity ratings may be useful in meeting an immediate need, they do little to increase our understanding of the underlying characteristics and seldom aid in finding solutions.

Vora (1995) classified usability problems based on the type of user error, which proved useful in identifying the source of problems within an interface but did not provide adequate guidance for improving the interface. Such a method of classification proved helpful in organizing the usability data, but still did not provide a consistent framework to guide and structure the process of collecting data. Jeffries (1994) concluded that research is needed to determine if clustering of usability problems by type can provide a more thorough understanding of their characteristics.

In sum, approaches to classification have been *ad hoc*, incomplete, unstructured and rather unhelpful for finding solutions to usability problems in an interaction design. In any case, studies comparing usability evaluation methods (Muller, Dayton & Root, 1993;

Desurvire, 1994; Karat, 1994) and reviews of those studies (Gray & Salzman, 1998) have concluded that some means is needed for classifying usability problems by type, so that methods can be assessed in terms of what kinds of problems they are best at identifying.

1.3.3. Reliability measures. Reliability of a tool or framework is a measure of the consistency of results across different users. As a formal measure, reliability is an index of agreement between two or more sets of nominal identification, classification, rating or ranking data. Cohen's (1960) kappa, a measure of the proportion of agreement beyond what would be expected on the basis of chance, is one example of a reliability measure.

There are other ways to compute a reliability measure. Sears (1997) measures reliability by using the ratio of the standard deviation of the number of problems found to the average number of problems found. Nielsen (1994*b*) used Kendall's coefficient of concordance to assess agreement among evaluators making severity ratings.

2. User Action Framework development history

2.1. THE USABILITY PROBLEM TAXONOMY AND THE USABILITY PROBLEM CLASSIFIER

Our current work in usability engineering methods and support tools began with the Usability Problem Taxonomy (Keenan, 1996; Keenan *et al*., 1999), postulated on the view that each usability problem possessed attributes in both a task- and an artefact-related dimension (Carroll, Kellogg & Rosson, 1991). The artefact dimension contained three major categories (visualness, language and manipulation) while the task dimension contained two major categories (task-mapping and task-facilitation). The resulting taxonomy consisted of four levels of problem types and one level of specific examples of usability problems. Keenan (1996) conducted an empirical study to evaluate the reliability of the Usability Problem Taxonomy. The study included seven participants and 20 problem descriptions to classify using the Usability Problem Taxonomy. Summative evaluation indicated that the Usability Problem Taxonomy yielded acceptable reliability at the first classification level on the artefact dimension ($\kappa = 0.403$), but only marginal reliability on the task dimension ($\kappa = 0.095$). We concluded that we needed better reliability if this was to be the foundation for integrating a suite of usability engineering tools.

Building on the Usability Problem Taxonomy, van Rens (1997) expanded the work begun by Keenan, creating the Usability Problem Classifier, with the addition of new content and adjustment of the structure. A "peel-off" mechanism was created up-front to rule out several less common but bothersome cases. The most significant new feature in this version was the classification of usability problems related to an action on an object in terms of the timing of the usability problem relative to the user action (i.e. occurring before, during or after the user action).

In a subsequent version of the Usability Problem Classifier, the current authors moved the "before, during or after" decision earlier in the classification process to include more of the task-based context, which was easier to examine at the beginning stages of classifying a problem. This conceptually simplified problem description and classification, but did not eliminate disagreements about classification results among our users.

At about this same point in the development of the Usability Problem Classifier, our work expanded to include other usability engineering support methods and tools (e.g. usability inspection and usability data maintenance tools). We noted that each tool required a structured way to organize usability concepts and issues in the context of the purpose of that tool. Rather than develop this structure separately for each new support tool, we began to unify their designs on a central structured model of usability concepts and issues, allowing for a consistent structure, content and "standard" usability language across all the tools. What began as development of a classification tool grew into the search for a unifying conceptual model to serve as a foundation for all the tools, and the question of reliability became even more important, but was being limited by three factors.

1. The underlying conceptual model was not consistently successful in helping users distinguish high-level cases to get started in the right direction early in the classification task.
2. Our user population continually exhibited an inherent variation in the way they interpreted the basic usability concepts and terminology.
3. There were a few cases where our purely hierarchical structure imposed an unnatural ordering on usability attributes that were not hierarchically related. We could represent only one possible ordering in a hierarchical structure, and classification with a different ordering resulted in a mismatch.

In the first several versions of the Usability Problem Classifier, we attempted to address the first problem, the conceptual model, in various ways but without complete success. In addressing the second problem, to overcome the variation in interpretation of concepts among users, we spent considerable time and effort making local adjustments in wording, but reliability gains were very limited. At this point in the project, we were hitting a frustrating "reliability wall". Our inability to achieve a level of reliability high enough for the Usability Problem Classifier to be *the* model to integrate our envisioned usability engineering tools continued to be a barrier to working on the usability engineering tools we so much wanted to develop.

In the end, two concepts emerged to solve the problems and lead us to the extremely high reliability result reported here, paving the way for development of the tools. First, we established a very effective conceptual model by adapting and extending Norman's (1986) theory of action model, a model that highlights issues about the way people interact with machines. Second, we introduced a small number of parallel paths in the structure. Divergent classification instances due to variation in interpretation or lack of hierarchical ordering could now reconverge, an impossible outcome in the purely hierarchical structure. In the next section, we discuss these two successful avenues in the quest for high reliability.

2.2. THE QUEST FOR HIGH RELIABILITY

2.2.1. A new conceptual model. Without a doubt the framework structure turned out to be the single most significant factor affecting reliability. The history of User Action Framework development has been largely the history of a search for an effective

structure that would organize the usability knowledge base in the most natural (and, therefore, hopefully the most reliable) way. The key to our success in finding such a structure was our adaptation of Norman's theory of action model of user interaction (Norman, 1986). Norman's cycle of interaction provided a starting point for the User Action Framework that gave immediate and dramatic increases in reliability. Because Norman's model applies to the interaction between a human and almost any machine, its adaptation within the User Action Framework also gets most of the credit for the broad applicability of the User Action Framework to many different interaction styles (e.g. Web, virtual environments, voice) and even beyond computers to elevators, ATMs, bus stop signs and refrigerators. Until our adaptation of this model, classification consistency had been persistently elusive.

In developing his concept of "cognitive engineering", Norman (1986) proposed a theory of action model, based on seven stages that occur during interaction by a human user with any kind of machine. Once we recognized the similarities between Norman's model and the structure of the Usability Problem Classifier, it was easy to visualize Norman's model as the starting point for generalizing our own model. In particular, our "before user action" and "after user action" portions of the Usability Problem Classifier corresponded approximately to the execution and evaluation sides of Norman's model, respectively. Similarly, our "during action" portion of the Usability Problem Classifier was very similar to Norman's physical activity component. We adapted and extended Norman's theory of action model into what we call the *Interaction Cycle*. The seven stages of Norman's model are shown in Table 1, along with corresponding Interaction Cycle categories. Table 1 represents our view of how to match the Interaction Cycle areas with Norman's seven stages. "Planning—high level" maps to the first two stages of Norman's model. In our Interaction Cycle, establishing the goal and forming the

TABLE 1

Correspondence of interaction cycle parts to Norman's theory of action

Norman's term	Interaction cycle term	Meaning in terms of usability issues
Establishing the goal	Planning—high-level	Can user determine the general requirements for getting started?
Forming the intention	Planning—high-level	Can user determine what to do to get started?
Specifying the action sequence	Planning—translation	Can user determine how to do it in terms of physical actions?
Executing the action	Physical action	Can user do actions (easily)?
Perceiving the resulting system state (change)	Assessment—understanding feedback	Can user see feedback?
Interpreting the state	Assessment—understanding feedback	Can user understand feedback?
Evaluating the system state with respect to the goals and intentions	Assessment—evaluating outcome	Can user determine success of outcome (of planning and actions)?

intention are high-level planning issues. Not until users specify the action sequence do they begin the translation process. "Physical action" maps directly to Norman's execution stage while "Assessment" maps to the last three stages in Norman's model. In the assessment part of the Interaction Cycle, understanding feedback requires first perceiving the feedback presentation—what we view as the almost instantaneous transition from physical actions to assessment.

The Interaction Cycle includes the concepts from all of Norman's stages, but organizes them pragmatically in a slightly different way. Like Norman's model, the Interaction Cycle is a picture of how interaction between a human user and *any* machine happens in terms of cognitive and physical user actions. This generality has served us well. We have also added a dual view, the machine (system) view of the same interaction and extended the concept to include interaction initiated by the system, by the environment or by the Interaction Cycles of other collaborating users (Kaur, Maiden & Sutcliffe, 1999). We reorganized and extended the concepts and issues that had been the content of the Usability Problem Classifier into a detailed, tool-independent, usability knowledge base we call the *User Action Framework*. Figure 2 shows the relationship between the Interaction Cycle and the User Action Framework.

The Interaction Cycle is both a cycle of user actions representing interaction with a computer (or any machine), in Figure 2(a), and the categories of the top level of the hierarchical User Action Framework knowledge base structure [Figure 2(b)]. The User Action Framework content, described in Section 3.1, is about the interaction design and how it supports and affects the user and task performance during interaction as the user makes cognitive or physical actions in each part of the cycle. The basic flow of user interaction is sequential around the cycle, but instances of actual interaction can include many alternative paths and variations, which are discussed in Section 3.2.

The real basis in Norman's (1986) model is the fact that we follow a cycle of interaction that includes the cognitive and physical actions of the user as they plan what to do, do it

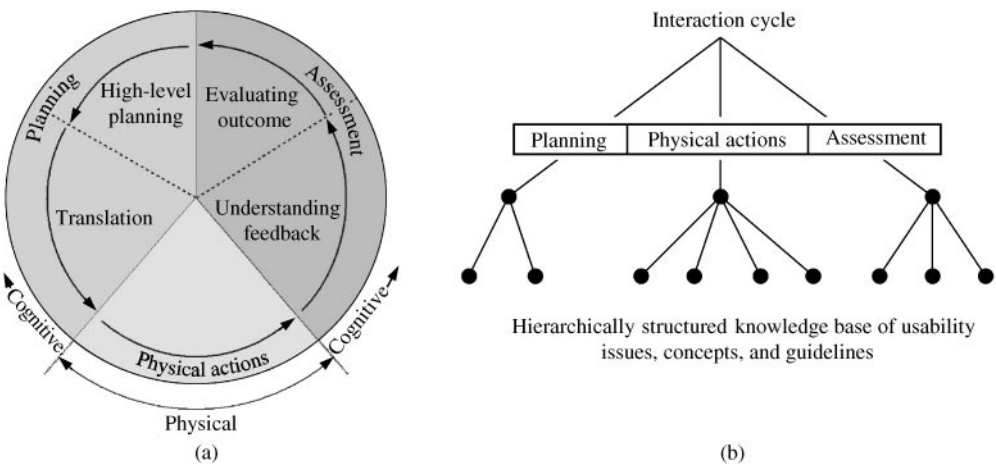

FIGURE 2. The Interaction Cycle Parts (a) and building the User Action Framework upon the Interaction Cycle (b).

and assess the outcome. The view of the Interaction Cycle in Figure 2(a) is not intended as a one-to-one mapping of Norman's seven stages to the content of the User Action Framework. Norman's stages helped us think about our own organization of the Interaction Cycle and the content of the User Action Framework. In the end, we were able to merge the concepts proposed by Norman and our own organization structure into a cycle that would help think about usability problems in a pragmatic way.

With the Interaction Cycle as an underlying structure, we achieved dramatic gains in reliability, and it remained only for us to tune the User Action Framework design to address some remaining variation among classifiers. This tuning focused on an approach to allow classifiers to reach an end-node description with more than one possible path.

2.2.2. Parallel classification paths. Formative evaluation observations made over several years during iterative refinements of the User Action Framework have led us to an inescapable conclusion: some kinds of variation will occur among classifiers. We have found it is impossible to design a classification scheme to avoid all variation. We must, rather, design to accommodate it.

For example, in our original view of classification reliability, we envisioned, with just the right wording at each node, every classifier could be led to take exactly the same classification path for a given usability situation (e.g. usability problem) being classified. So, we used formative evaluation as a strong input to design. When an evaluator subject failed to classify a problem on the expected path, we determined the rationale for the errant classification by interviewing the subject. We made changes in the wording of the User Action Framework content, determined by formative evaluation in a manner similar to the way Good, Whiteside, Wixon and Jones (1984) developed "user-derived interfaces". If one of the user-participants of Good *et al.* made an "error" in a command, for example, they used a "Wizard of Oz" technique to modify the interaction design so that it would have worked for that user. In a similar manner, at each node where the user's classification path deviated from our expected choice, we added a "semantic attractor" (which we also called "semantic flypaper") to our "correct" choice and "semantic deflectors" to "incorrect" choices made by users. (We put the terms "correct" and "incorrect" in quotes here because we do not view classification choices as absolutely correct or incorrect; we were seeking consistency in the choices.)

While the attractors and deflectors gave us an initial boost in reliability, they failed as a sole strategy for steering all users into agreement on classification paths. We soon reached a limit where additional changes to avoid divergence by one user would work against previous changes made to avoid divergence in other users. We realized that it would be impossible to converge on a single overall set of deflectors and attractors that would work the same for all users. Further adaptations to our users would only cause local oscillations that would not improve overall reliability. We wonder if the Good *et al.* group experienced the same kind of limits in their work.

Subsequently, we reasoned that classification reliability required only consistent final classification results, not identical classification paths. So we deviated from a purely hierarchical structure and provided alternative paths for some classification choices. When classification paths for the same usability situation occasionally diverged, we were thus able to secure reconvergence on the same final classification node. This reconvergence was most effective where two usability attributes were more or less orthogonal.

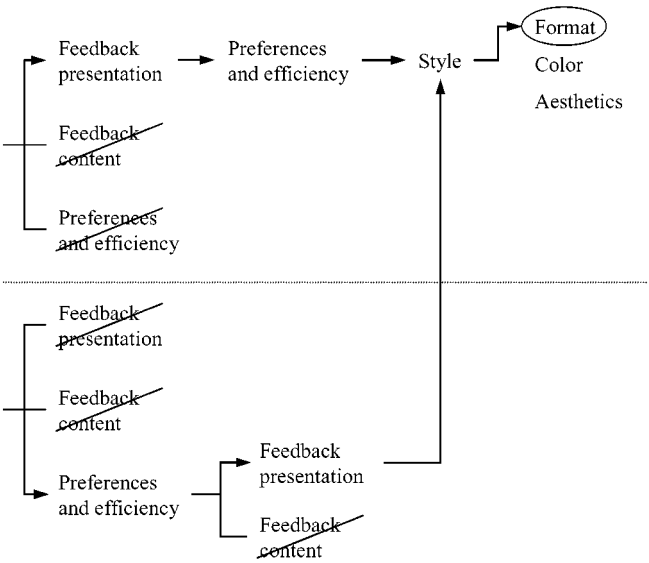

FIGURE 3. Alternative paths to classify a usability problem.

Consistency within a pure hierarchy had forced us (and User Action Framework users) always to put the same attribute first in the classification sequence when, in fact, no such natural ordering existed. That meant adding artificial rules about which attributes to consider first and so on. An example clarifying this concept involves the usability attribute "preferences and efficiency". Preferences and efficiency attributes are often independent of (especially in terms of ordering) many other usability concepts. As an example, consider the situation illustrated in Figure 3.

In this example, an error message (feedback) might have attributes that describe its presentation or appearance and other attributes that describe its meaning or content. If a usability situation being classified involves preferences and efficiency issues about the presentation of an error message, then some users might choose feedback presentation first and preferences and efficiency second (top of Figure 3). Others might choose preferences and efficiency followed by feedback appearance (bottom of Figure 3). In either case, the user eventually selects an attribute relating to the format of the message and neither path is more correct than the other in arriving at this end point. The User Action Framework became a quasi-hierarchical structure with some parallel alternate classification paths, both of which lead to the same final combination of attributes, avoiding an artificial source of inconsistency.

Until the study reported here, lack of reliability in the User Action Framework had been an insurmountable barrier to continuing our work on usability engineering support tools. Now that we finally have a design for the framework with provably high reliability in usage, we are able to move forward with tool development.

3. Description of the User Action Framework

For several years we have been compiling usability concepts from guidelines, published literature, large amounts of real-world usability data, and our own experience into

a structured usability knowledge base organized on user cognitive and physical actions within the Interaction Cycle and structured into levels of abstraction within the User Action Framework.

The User Action Framework, which has now become our stock-in-trade basis for organizing, discussing, classifying and reporting usability problems, derives a powerful generality and flexibility in its application across many tools and interaction styles from its theory base in Norman's (1986) model. The User Action Framework also derives essential practical and applied qualifications from its empirical roots in the usability data on which its content is based.

Like Norman's theory of action model, the User Action Framework describes user activities and experiences at "run time" (i.e. when the user interacts with the computer). In addition, *and more importantly for usability engineering methods and tools*, the User Action Framework also supports developers' and evaluators' design-time analysis of the effects of an interaction design on users proceeding through the Interaction Cycle. Figure 4 shows in a bit more detail of the top-level User Action Framework categories, corresponding to cognitive and physical actions users make while performing a task using a computer.

3.1. USER ACTION FRAMEWORK CONTENT

Planning includes both high-level planning and translation. High-level planning is concerned with the user's ability to understand the overall computer application in the perspective of work context, problem domain and environmental requirements and constraints. High-level planning has to do with the system model and metaphors, and the user's knowledge of system state and modalities. High-level planning includes user work goal decomposition across a hierarchy of plan entities: goals, task and intentions, all expressed in cognitive problem-domain language.

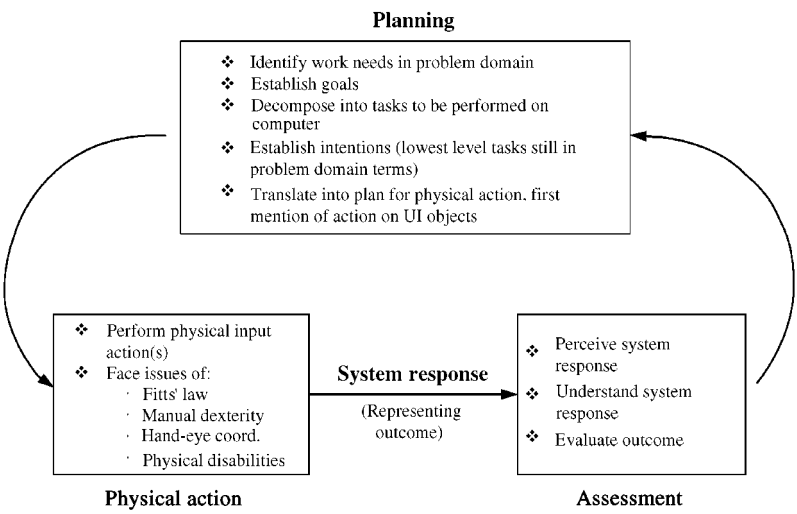

FIGURE 4. Representation of process flow through Interaction Cycle parts.

In translation, the final stage of planning, the user determines what physical actions to take in order to accomplish an intention, translating intentions into plans for physical action(s). Because users often depend on cognitive affordances in the interaction design (e.g. visual cues) to make this translation, usability issues in the User Action Framework under translation include those that pertain to their cognitive affordance presentation, and their content or meaning.

The physical action part of the User Action Framework is about perceiving and manipulating user interface objects, including issues of interaction complexity, I/O devices, interaction styles and techniques, manual dexterity, layout (Fitts' law) and physical disabilities. Physical action is all about the user's execution of the plan. In graphical user interfaces, this mainly involves clicking, dragging, selecting, etc.

The assessment part includes issues about feedback and how it supports the user's ability to gauge the outcome of physical actions. The structure of the assessment part parallels that of the translation part in that it has to do with presentation of feedback, meaning of feedback and preferences and efficiency. Just as in Norman's (1986) model, assessment has primarily a cognitive demand on the user as they try to understand the feedback and determine success of the outcome.

3.2. MOVING THROUGH PARTS OF THE INTERACTION CYCLE

The typical user behavior for users in planning involves work goal decomposition across a hierarchy of plan entities: goals, task and intentions. Users establish a goal in the work domain (e.g. produce business letter). Goals break down into tasks (e.g. formatting the page) that subsequently spawn intentions (e.g. user intends to set left margin). Not every user rigidly follows this sequence of steps for every goal or task. Exceptions and variations occur in several ways. Planning often involves skipping steps, changing plans (e.g. intention shifts for error recovery or exploration), working without conscious planning, planning only as a response to the interaction situation, or even as scripted behavior acquired via rote learning (without understanding).

GUI-based interaction, especially by new users, is often based on initial planning to get started and then a pattern of user action, system response and the user making the next action in reaction to that system response, and so on. This kind of turn-taking interaction pattern is variously called situated interaction (Suchman, 1987; Kaur *et al*., 1999) or display-based interaction (Payne, 1991) or incremental or opportunistic planning (Weller & Hartson, 1992). In such situations, users rarely work out the plans for many tasks or intentions in advance.

Highly practiced expert users can perform tasks in an "automated" manner, without conscious planning. In the context of activity theory (Bodker, 1991), this is called an automated operation, in contrast to a planned, controlled action. According to another theory of action (Lim *et al*., 1996), such users raise the level of abstraction at which they perceive such tasks, not thinking about the details below that level.

All of these aspects of planning within user interaction are accommodated (or potentially accommodated) by various different categories within the User Action Framework. Further discussion of these cases is beyond the scope of this paper.

3.3. PUTTING THE USER ACTION FRAMEWORK TO WORK

3.3.1. Mapping the User Action Framework to usability engineering support tools. Application of the User Action Framework is in its mapping to usability engineering support tools. A mapping to a given tool retains the content and structure of the User Action Framework, but changes the way the content is expressed; each knowledge item is rephrased into an *expression* reflecting the *purpose* of the tool. When mapped to the Usability Problem Classifier tool, for example, each concept is expressed as a classification-oriented question about an *observed usability problem.* In the Usability Problem Inspector tool each concept maps to a question asking for an expert judgment concerning a *potential usability problem* in the system being inspected. The same concepts are expressed in the Usability Design Guide tool as prescriptive *advice for avoiding usability problem* of each type. Finally, the Usability Problem Database tool contains usability data related to (classified under) each usability concept. The Usability Problem Database supports usability data maintenance (stored by problem type) within a project life cycle, and supports sharing and reuse of usability analysis and usability problem information and solutions that have worked in other similar situations. The Usability Problem Database also allows visualization of usability data populations and clusters by problem type to help focus the development process.

3.3.2. Mapping the User Action Framework to interaction styles. The original orientation of the User Action Framework was for GUIs, the most common interaction style presently. The value of the generality of the underlying model, acquired from Norman's (1986) model, is realized when one considers extending the User Action Framework to other interaction styles. All other interaction styles share the same underlying Interaction Cycle process based on what the user thinks, sees and does. Applying the User Action Framework to other interaction styles simply involves different terminology and different emphases. For example, mappings to Web and other hypermedia applications emphasize critical issues such as navigation and "getting lost in hyperspace". Mappings to virtual environments highlight such usability issues as gestural interaction and 3-D visualization.

To examine the issues involved in this kind of mapping, we manually constructed a mapping from GUIs to voice-driven interaction for a usability inspection of a commercial voice I/O email system (Hartson, Andre, Williges & Van Rens, 1999). The mapping involved such changes as substituting "auditory cues" for "visual cues" and emphasizing issues related to human memory, which is relied on more with spoken menus than with visual menus. We were encouraged by finding that this mapping to a voice interaction style required relatively little time and effort and the corresponding usability inspection process produced effective results.

4. Reliability of the User Action Framework

With ample baseline data from our reliability studies of the Usability Problem Taxonomy and the Usability Problem Classifier, we set out to measure the reliability of our new model: the User Action Framework. Understanding of the meaning and use of the

User Action Framework is well represented by the ability to correctly and consistently classify usability situations within the structure. Therefore, the goal of the reliability study was to document the level of agreement among experts when classifying a given set of usability problem case descriptions using the User Action Framework as a classification tool. Along with our baseline reliability data from the Usability Problem Taxonomy, we also wanted to compare our results with classification reliability obtained from the same experts using Nielsen's (1994*a*) revised set of heuristics.

4.1. OVERVIEW OF RELIABILITY MEASURE

One important performance measure of a usability engineering tool is reliability, which is a measure of the consistency, or the extent of agreement, among evaluators with respect to their results in using the tool. Although there are several methods to measure reliability (Meister, 1985), the kappa (κ) statistic (Cohen, 1960) is commonly used to examine observer agreement for categorical lists or taxonomies; it is especially useful when the occurrence of chance agreement needs to be considered. Kappa is scaled between -1 and $+1$. Positive values of kappa correspond to greater than chance agreement, zero represents only chance agreement, and negative values correspond to less than chance agreement. Kappa is approximately normally distributed and can be used to test the null hypothesis of whether agreement exists beyond the chance level. Kappa is traditionally used to assess agreement between two observers. In the present study, more than two observers were used, requiring an extension to the kappa statistic provided by Fleiss (1971) to measure the level of agreement among several observers.

4.2. PARTICIPANTS

The participants for the User Action Framework reliability study consisted of 10 usability professionals recruited from government and commercial organizations. Nine of the ten usability professionals participated in the follow-up heuristic reliability study. The participants were selected from five different organizations where usability engineering (design, test or evaluation) was a formal part of their daily job experience. All participants possessed at least a bachelor's degree in computer science, human factors, psychology or industrial engineering. A majority (6 of 10) of the participants possessed an advanced degree (masters or Ph.D.). The average age of the participants was 35 years, ranging from 25 to 47 years. All participants had a minimum of 3 years experience in user interface design, test, and/or evaluation ($M = 7.9$). Participants were equally split in terms of their self-reported usability specialty with half from the design perspective and the other half from the test and evaluation perspective.

4.3. MATERIALS

Materials for the User Action Framework reliability study included a local Website containing the User Action Framework content linked together to facilitate traversal of the knowledge base. Figure 5 shows the User Action Framework start page with three primary areas of the Interaction Cycle (planning, physical actions and assessment) represented as hypertext links. By selecting a particular link (e.g. physical actions), participants went further into the structure as shown in Figure 6.

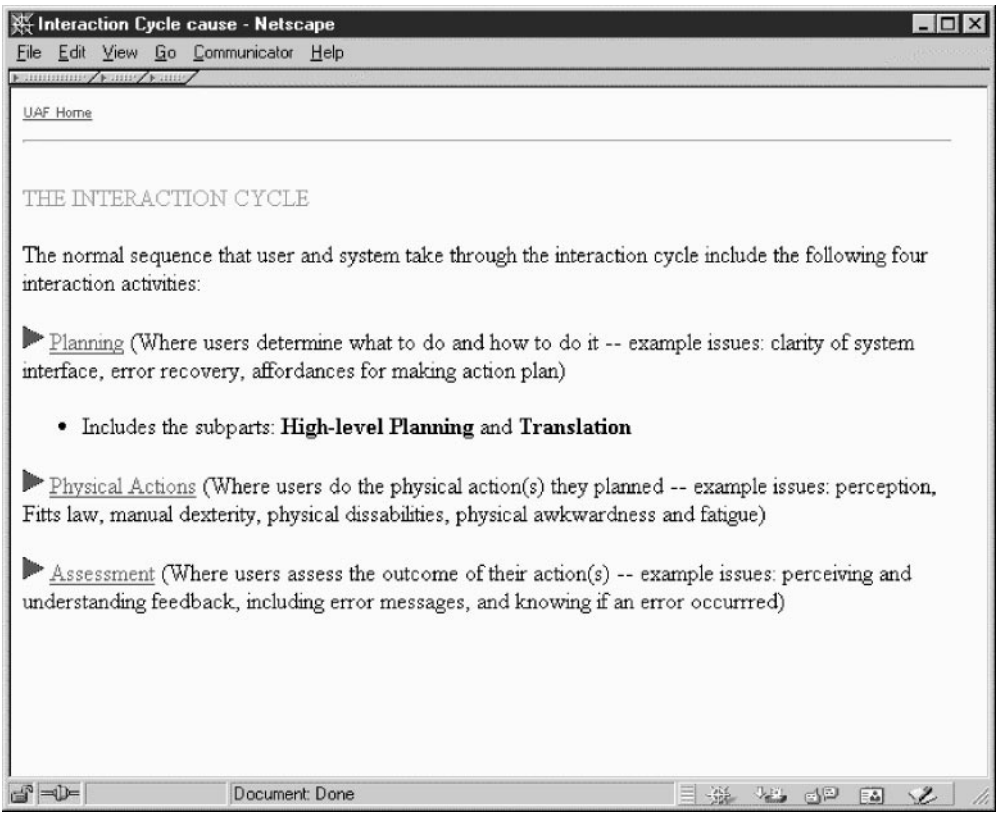

FIGURE 5. Start page for the User Action Framework.

Fifteen usability problem case descriptions were selected from a larger database containing over 100 usability problem cases. These descriptions were collected from various software development projects and personal experience. Case descriptions employed in this study were limited to those representing only a single usability problem or concept; case descriptions representing multiple, related usability problems were not considered. The 15 case descriptions were also selected on the basis of their real-world expected frequency of occurrence relative to their location within the Interaction Cycle. Based on a pilot study (Hartson *et al*., 1999), a majority of usability problems were found to exist under translation within the planning portion of the Interaction Cycle. Assessment contained the second largest portion of the usability problems; the fewest problems were found to occur within the physical actions portion of the Interaction Cycle. Table 2 provides a summary of the 15 cases used in the reliability study, and how they are distributed within the User Action Framework.

Materials for the heuristic reliability study consisted of Nielsen's (1994*a*) revised set of 10 heuristics shown in Table 3 along with the same 15 usability problem descriptions used in the User Action Framework reliability study.

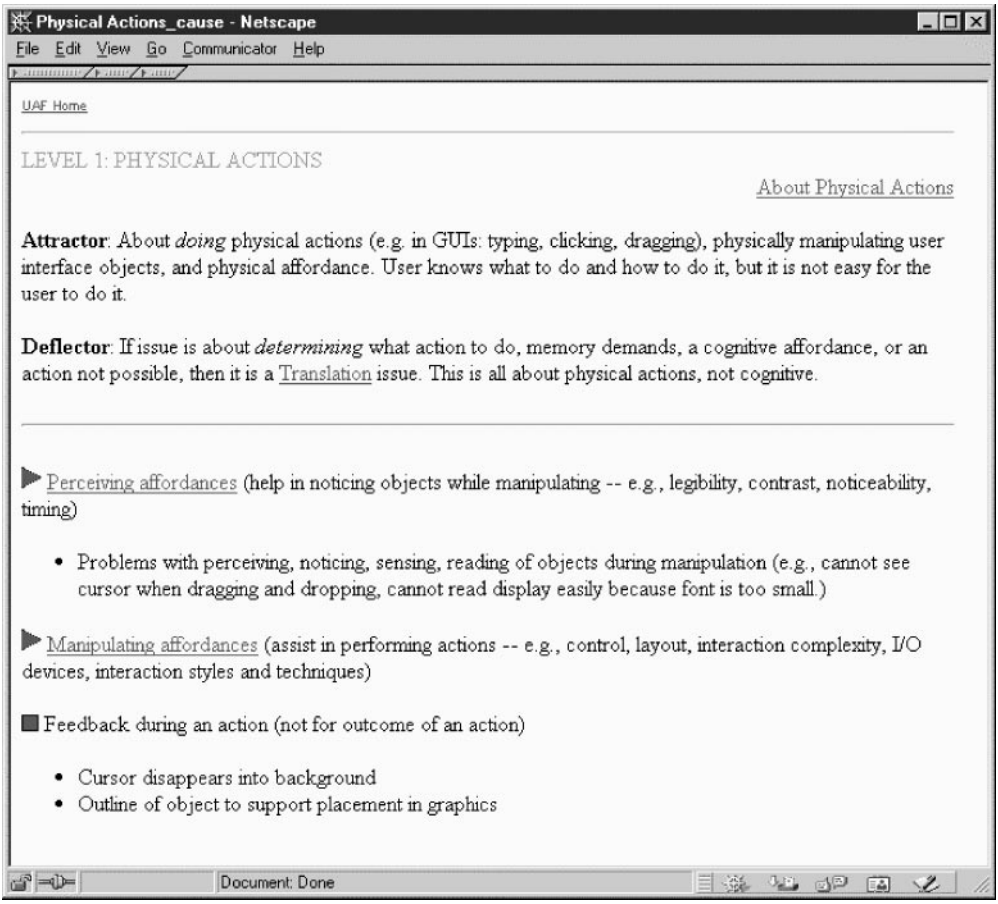

FIGURE 6. Example of physical actions page in the User Action Framework.

4.4. PROCEDURE

For the User Action Framework study, each participant viewed a 20-min tutorial on the User Action Framework. This tutorial involved an on-line description of the Interaction Cycle components, the structure of the User Action Framework, and an example of how to use the Web-based User Action Framework to classify a usability problem. Participants then read the 15 case descriptions and used the Web-based User Action Framework to classify the problem. Even though problem descriptions were selected on the basis of having only one usability issue, we directed participants to classify only the primary problem on the chance they interpreted a problem description as having two or more usability problems. Participants could traverse any number of paths of the User Action Framework before noting their final classification of the usability problem.

The same experts were asked 1 month later to participate in the heuristic reliability study using the 15 usability problem case descriptions from the User Action Framework reliability study. One of the original 10 experts was not able to participate in the heuristic

TABLE 2

Usability problems used in the reliability study

Case no.	Type of usability problem	Relevant area in User Action Framework
1	Unreadable error message	Assessment
2	User does not understand master document feature	Planning (high-level)
3	User cannot find a feature to support re-using document numbers in a document retrieval system	Planning (translation)
4	User clicks on wrong button	Physical actions
5	User cannot directly change a file name in an FTP program	Planning (translation)
6	User cannot tell if system is performing requested operation	Assessment
7	User wants to fix database error but is confused by button labels for appropriate action	Planning (translation)
8	Program does not provide a Ctrl-P shortcut for printing	Planning (translation)
9	User cannot understand error message provided by system	Assessment
10	Unnecessarily long error message	Assessment
11	Unwanted confirmation message	Assessment
12	User does not see way to select odd font size	Planning (translation)
13	Data entry format not provided	Planning (translation)
14	Uncontrollable scrolling	Physical actions
15	Vision-impaired user needs preference options for setting larger font size	Planning (translation)

TABLE 3

Revised set of usability heuristics (from Nielsen, 1994a)

Heuristic
Visibility of system status
Match between system and real world
User control and freedom
Consistency and standards
Error prevention
Recognition rather than recall
Flexibility and efficiency of use
Aesthetic and minimalist design
Help users recognize, diagnose and recover from errors
Help and documentation

evaluation, thus leaving us with nine participants for the heuristic evaluation study. Participants were mailed a heuristic evaluation packet with Nielsen's (1994*a*) revised set of 10 heuristics, a 20-min training package adapted from Nielsen's (1993) *Usability Engineering* book, and paper forms to record the primary heuristic they would apply to each usability problem description.

Time to complete the classification for each method was not controlled since the primary focus was on the utility of the final classification decision for each problem.

Instructions to the participants indicated the process would take approximately 90 min. Participants using the User Action Framework did not go beyond 90 min, but generally needed the full time to complete the classification. Feedback from the participants using the heuristic evaluation indicated that 90 min was more than enough time. We did not collect completion data on the heuristic evaluation process since these participants completed the classification from their sites and mailed their answers back to the experimenters.

4.5. HYPOTHESES

Although a primary goal of the study was to document the reliability of the User Action Framework, we did establish two hypotheses arising from our previous work, and expectations regarding the results of the heuristic evaluation. Our first hypothesis was the User Action Framework would result in a higher overall reliability score (kappa) than was found in our previous work on the Usability Problem Taxonomy (Keenan *et al*., 1999). Keenan *et al*. showed classification within the Usability Problem Taxonomy was reliable at the first classification level (the level of the five primary categories) on the artefact dimension ($\kappa = 0.403$, $p < 0.001$), but marginally reliable on the task dimension ($\kappa = 0.095$, $p > 0.10$). Our expectation for higher agreement is based on extensive formative evaluation and on the assumption that the Interaction Cycle, based on Norman's (1986) theory of action model, provides a more natural way to think about the types of interaction problems users encounter. In addition, the provision of alternative paths for some classification choices rather than a pure hierarchical structure allows for reconvergence to agreement on the same final classification node even though the users of the tool may take slightly different paths.

Our second hypothesis was the User Action Framework would result in higher reliability than results obtained from experts using heuristics to classify the 15 usability problem cases. Although not originally intended as a classification framework, heuristics have been used by practitioners as labels for both finding and discussing problems. As a result, their reliability in classifying usability problems is important to both researchers and practitioners.

4.6. DATA COLLECTION AND ANALYSIS

The primary data for User Action Framework reliability study were the participant's path through the User Action Framework in classifying a problem. For each case description, we recorded the classification path taken by the participant and documented their selection of end-node descriptions. Consider the example illustrated in Figure 7 where the usability problem involves a user having a difficult time reading a feedback message.

In this example, the first four levels comprise the classification nodes where choices within each node were designed to be orthogonal. The choices within a node at the lowest level of the hierarchy, level 5 in this example, were not intended to be orthogonal for classification. Rather, these end-node descriptors were developed to augment the classification with a description of several possible causes related to the final classification node. Font size and contrast were the descriptors chosen to portray the nature of the

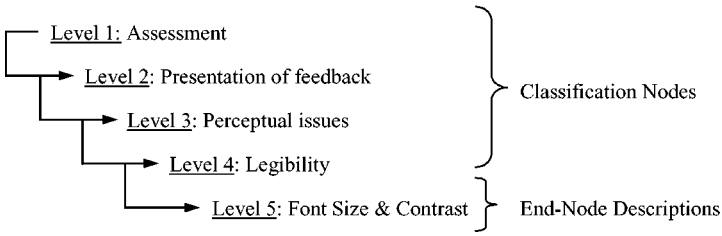

FIGURE 7. Example path for a usability problem involving a feedback message.

legibility problems in the above example. Agreement at the end-node descriptions was defined as two or more experts having an element in common. For example, if one expert selected *Size and Color* while another expert selected *Size and Contrast*, then these two experts are in agreement since *Size* is a common element.

Because the User Action Framework is comprised of a number of classification nodes at various levels (as many as six levels), we calculated expert agreement at each of the different levels within the hierarchical structure as well as overall agreement at all end-node descriptions. For each usability case description, the participant using the User Action Framework is presented with a range of choices that are dependent upon the path taken to describe the problem. At the top levels of the User Action Framework, the number of choices are usually small; typically the choices are between two or three items. The User Action Framework broadens at deeper levels, presenting the user with as many as eight choices at the lowest classification nodes. Therefore, the small differences in choices made early on result in large differences in terms of the number and kinds of choices faced later. As a result, the hierarchal structure of the User Action Framework essentially holds up a higher standard for reliability because once two classifiers disagree, there is little or no chance for them to later reconverge to agreement.

Data from the heuristic reliability study were relatively straightforward in terms of measuring agreement since there were only 10 categories and no hierarchical levels. The nine participants in the heuristic reliability study indicated their primary choice of a heuristic that applied to each case description on paper forms where Nielsen's (1994*a*) 10 heuristics were listed.

4.7. RESULTS

Reliability measures, such as kappa, are intended to measure classifier agreement across a fixed number of categories. To measure classifier agreement in the User Action Framework, we analysed the agreement in three ways: (1) reliability at each level within the hierarchical structure, (2) reliability within the respective parts of the Interaction Cycle (i.e. planning, physical actions, assessment) and (3) overall reliability for end-node descriptions. We analysed classifier agreement for the heuristic reliability study by treating the 10 heuristics as one level of end-node descriptions.

4.7.1. Agreement at levels in the User Action Framework. Figure 8 shows an example of the data from one usability case description; case 10, which is about an unnecessarily

CASE 10	LEVEL IN UAF				
Participant	LEVEL 1	LEVEL 2	LEVEL 3	LEVEL 4	LEVEL 5
1	Assessment	Content	Clarity	Complexity	Volume & Verbosity
2	Assessment	Content	Clarity	Complexity	Volume & Verbosity
3	Assessment	Content	Clarity	Complexity	Volume & Verbosity
4	Assessment	Content	Clarity	Complexity	Volume & Verbosity
5	Assessment	Content	Clarity	Complexity	Volume & Verbosity
6	Assessment	Existence	Level inappropriate		
7	Assessment	Content	Clarity	Complexity	Volume & Verbosity
8	Assessment	Content	Clarity	Complexity	Volume & Verbosity
9	Assessment	Content	Completeness	Level of detail	
10	Assessment	Content	Clarity	Complexity	Volume & Verbosity
Agreement	**10 out of 10**	**9 out of 10**	**8 out of 9**	**8 out of 8**	**8 out of 8**

FIGURE 8. Example summary of participant categorization of a usability case description.

long error message displayed to the user. Level 1 shows that all 10 participants agreed that this particular usability case description was an *Assessment* problem because it involved a feedback message. At the next level (level 2), nine of the 10 participants agreed that the case description was about the *Content* of the feedback message. One participant felt the issue was related to the *Existence* of the feedback message. To continue measuring agreement accurately, we had to eliminate participant #6 from further reliability measures since this person was now taking a different path than the remaining nine. Thus, at level 3, eight of the remaining nine participants agreed that the issue was about the *Clarity* of the feedback message. Participant #9 felt the problem was related to the *Completeness* of the feedback message. As a result, we eliminated participant #9 from reliability analysis at the next level. At level 4, all eight remaining participants agreed that the issue was about the *Complexity* of the feedback message. At the end node for this path (level 5), all eight participants selected *Volume & Verbosity* as the usability cause for this case description. The example illustrated in Figure 8 shows how we approached calculating reliability at different levels by eliminating participants who proceeded down a different path from the majority. This helped us avoid continuous penalties for further disagreement at lower levels when a participant was on a different path and had no opportunity to see the same choices as the other participants. Although elimination of participants at lower levels inflates the agreement among the reduced number of participants, such an approach allowed us to represent the actual agreement at lower classification levels.

Table 4 shows the reliability results for each level within the User Action Framework. Column 2 indicates the number of cases analysed for each level within the User Action Framework. Depending on the case, classifiers had to traverse a number of hierarchical levels before reaching the final page with end-node descriptions. For example, some cases used in this study required navigation down to only the fourth level in the User Action Framework before end-node descriptions were presented. As shown in Table 4, nine cases required level 5 classification while only three cases required level 6 classification. One case only required classification to level 3 in the User Action Framework. Values in the P_o column indicate the proportion of observed agreement while values in the

TABLE 4

Results of reliability analysis at each level

Level	Cases at this level	Average agreement				
		P_o	P_c	(%)	κ	Z
1	15	0.987	0.408	99.3	0.978	20.25***
2	15	0.979	0.274	98.7	0.972	22.17***
3	15	0.800	0.081	86.7	0.783	60.77***
4	14	0.781	0.082	81.4	0.762	53.28***
5	9	0.752	0.118	80.0	0.719	32.37***
6	3	0.565	0.378	63.3	0.299	2.57**

** $p < 0.01$.
*** $p < 0.001$.

P_c column indicate the proportion of agreement expected by chance. Kappa accounts for the fact that the proportion of chance agreement decreases as the number of choices increase. As shown in Table 4, the proportion of chance agreement is higher at the top levels in the User Action Framework than the lower levels because there are fewer choices at the top of the framework. Thus, observed agreement requires substantially higher values to overcome chance agreement at the top levels of the User Action Framework. Average agreement is shown in column 5 to indicate the percentage of participants, on average, who agreed at each level in the User Action Framework. The kappa values shown in Table 4 (κ column) indicated strong agreement through level 5 within the User Action Framework. The Z column contains the observed values for the standard normal variate obtained by dividing kappa by its standard error. The high z values through level 5 of the User Action Framework indicated that kappa scores were significantly greater than chance agreement ($p < 0.001$). Level 6 classification only applied to three cases producing $\kappa = 0.299$, indicating agreement was comparatively low at the lowest level in the User Action Framework. The corresponding z value still indicated the agreement at level 6 was significantly greater than chance ($p < .01$).

4.7.2. Agreement for the Interaction Cycle parts of the User Action Framework. Table 5 shows the reliability analysis for the Interaction Cycle parts of the User Action Framework. We included the sub-part, translation, because of its relevance to a number of usability problem case descriptions.

The number of categories for the expert to select from at each part in the User Action Framework is shown in column 2 of Table 5. The Interaction Cycle parts presented classifiers with two or three choices, except for translation, which presented five choices. Column 3 shows how the case descriptions were distributed among the Interaction Cycle parts. The planning part, which includes high-level planning and translation sub-parts, had most of the cases (of which translation had 7 of the 8). Assessment had five and physical actions had two relevant cases. Agreement was very strong for the Interaction Cycle parts with kappa ranging from 0.673 to 0.943. Agreement scores were significantly greater than chance as indicated by high z values ($p < 0.001$). The results also revealed

TABLE 5

Results of reliability analysis for the Interaction Cycle parts of the User Action Framework

Interaction Cycle Parts	Categories	Relevant cases	P_o	P_c	κ	Z
Planning	2	8	0.987	0.779	0.943	3.65***
Translation	5	7	0.760	0.265	0.673	17.92***
Physical actions	2	2	1.000	1.000	—	—
Assessment	3	5	0.960	0.361	0.937	15.33***

Note. Dashes indicate too few cases to calculate classifier agreement.
*** $p < 0.001$.

that the translation sub-part was more difficult in terms of obtaining consistent agreement among the classifiers. Planning and assessment parts showed very high agreement, indicating that classifiers were able to easily differentiate problem attributes based on these two parts of the Interaction Cycle. Reliability calculations for classification within the physical actions part of the User Action Framework were not possible because of the limited data points generated from only two relevant cases with a binomial choice at this level. Even though agreement for these two cases was perfect, kappa calculates to zero since P_o and P_c are equal at 1.000.

4.7.3. Overall agreement. We also calculated overall agreement of classifiers by examining the final end-node descriptions across all usability cases. Thus, the overall agreement provides reliability information for the various paths taken by each classifier. Kappa results for overall agreement (Table 6) showed strong reliability ($\kappa = 0.583$, $p < 0.001$), indicating agreement is greater than what would be expected by chance. In calculating kappa across all cases, we essentially transformed the six hierarchical levels of the User Action Framework into a flat structure with more than 150 end-node descriptions. Therefore, the probability of chance agreement was extremely small ($P_c = 0.048$) considering the number of possible end-node descriptions available to the classifiers.

4.7.4. Heuristic evaluation results. Data from reliability calculations for the heuristic reliability study are shown in Table 7. Kappa results showed moderate agreement ($\kappa = 0.325$, $p < 0.001$), indicating agreement is greater than what would be expected by chance. Results from hypothesis testing for independent samples revealed that the reliability of the heuristic classifiers was not as strong as the reliability obtained from the User Action Framework classifiers ($p < 0.001$). Only the agreement levels at level 6 in the User Action Framework were comparable to the result obtained by the heuristic classifiers. Table 8 summarizes these results from the statistical hypothesis testing using the standard normal distribution (z) as the test statistic.

5. Discussion

Built as a structured knowledge base of usability concepts and issues, the User Action Framework is intended to provide a framework underlying usability engineering support

TABLE 6

Overall reliability for the User Action Framework

No. of cases	P_o	P_c	κ	Z
15	0.603	0.048	0.583	61.86***

*** $p < 0.001$.

TABLE 7

Results of reliability analysis for heuristic reliability study

No. of cases	P_o	P_c	κ	Z
15	0.404	0.116	0.325	19.13***

*** $p < 0.001$.

TABLE 8

Summary of reliability comparison between heuristic and User Action Framework Classifiers

κ (User Action Framework)	κ (Heuristic evaluation)	Z (κ_{UAF}–κ_H)	Conclusion
0.583 (overall)	0.325	12.90***	User Action Framework > heuristic
0.978 (level 1)	0.325	12.75***	User Action Framework > heuristic
0.972 (level 2)	0.325	13.79***	User Action Framework > heuristic
0.783 (level 3)	0.325	21.12***	User Action Framework > heuristic
0.762 (level 4)	0.325	19.54***	User Action Framework > heuristic
0.719 (level 5)	0.325	14.02***	User Action Framework > heuristic
0.299 (level 6)	0.325	− 0.77	No significant difference ($p > 0.10$)

*** $p < 0.001$.

tools to aid practitioners with a standardized method for developing usability problem descriptions that distinguish different problem types and help form a shared understanding of the specific attributes of the problem. We conducted a reliability study to determine the degree of consistent use by usability practitioners classifying a given set of usability problems. We believe consistent classification of usability problems is necessary to produce high-quality problem reports that lead to more direct solutions and more efficient use of resources in the documentation process.

Results from the User Action Framework reliability study showed higher overall agreement ($\kappa = 0.583$) than was found in our previous work with the Usability Problem Taxonomy ($\kappa = 0.403$). More importantly, agreement scores through level 5 of the User Action Framework ($\kappa = 0.719$–0.978) were higher than the top level of the Usability

Problem Taxonomy ($\kappa = 0.403$). Finally, agreement using the User Action Framework (levels 1–5) was significantly stronger than the results obtained from the same experts using the heuristic evaluation ($\kappa = 0.325$). Agreement at level 6 in the User Action Framework was not particularly strong, but this was not surprising since the types of end-node descriptions at this level represent very fine differentiations of a single characteristic. Evaluators using the User Action Framework were especially consistent at using the parts of the Interaction Cycle to begin their classification of each usability problem. Only one evaluator on one usability problem diverged to a different part of the Interaction Cycle during the classification process. Such a result supports the notion that a model-based framework is important to providing a reliable classification system.

As a hierarchically structured knowledge base, the User Action Framework provided much more description and discrimination power than the heuristic evaluation technique. Heuristic categories are generally not distinct and often result in evaluator confusion when selecting appropriate labels for a problem (Jeffries *et al.*, 1991; Doubleday *et al.*, 1997). In terms of measuring reliability, a hierarchically structured framework is more problematic than a flat structure such as Nielsen's (1994*a*) heuristics. In addition to overall reliability, we were also able to report classifier agreement at each level according to the number of usability cases applying at that level. Reporting classifier agreement at each level allowed us to examine how users were able to consistently understand and select choices as they traversed the hierarchical structure. In fact, reporting classifier agreement by level provides more information regarding the structure of the User Action Framework than we can obtain by just looking at overall reliability. Reporting reliability for only an end-node description hides valuable information about classifier agreement at previous levels reaching the end-node. As an example, consider the case shown in Figure 9 where evaluators start to disagree at level 3. The circles represent the number of evaluators choosing a particular node in the classification hierarchy. Disagreement is apparent if reporting results at level 4. However, to report on the agreement (or disagreement in this case) at level 4 without consideration for the extent of agreement at higher levels essentially masks an important aspect of classifier agreement reaching the end-node. That is, the description at level 4 is only complete in the context of the path taken to a particular end-node description. In the example shown in Figure 9, the entire context of the path for 4 of the 10 classifiers is something like: *The usability problem is a high-level planning issue involving the user's model of the system in order to understand the overall concept*.

Although our results show improved reliability due to our iterative refinements, it is difficult to make literature-based comparisons. Reliability, as defined in this study, is not documented in the current usability literature. Practitioners would not deny the importance of providing consistent results from usability engineering support tools, but the matter of operationally defining consistent performance is a different issue. Some practitioners may be interested in knowing that one evaluator can use a tool consistently across projects. Others may be more interested in knowing that different evaluators are relatively consistent in their use of the usability engineering support tool. In either case, the consistent use of a tool does not guarantee that the output of usability evaluation will produce quality problem reports that communicate problems and causes precisely and suggest solutions for down-stream redesign activities. Providing better quality problem reports depends on both the structure of the usability framework that guides the

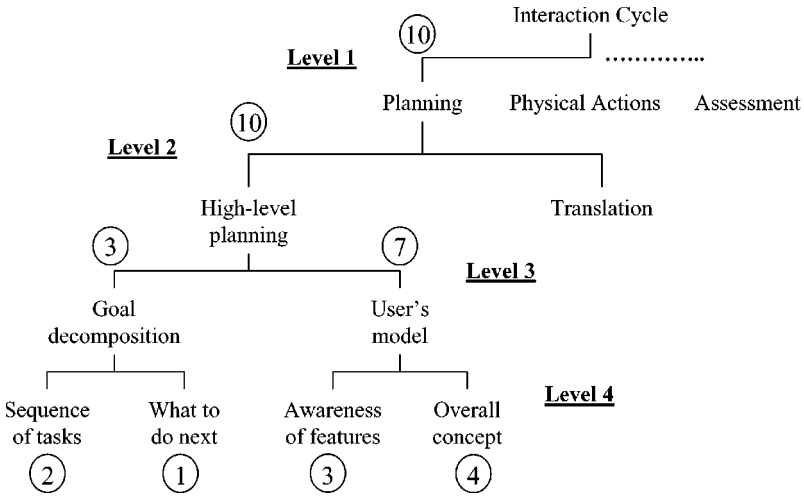

FIGURE 9. Example of classification path for a usability problem.

description process and the content of the framework that helps to provide a complete understanding of the usability problem.

In testing the reliability of the User Action Framework, our focus was primarily on the iterative refinement of the content and structure to produce the most effective tool for the practitioner. Iterative refinement allowed us to have easy transition to a summative study in order to document the level of reliability; measuring our own progress and comparing to another usability evaluation method. In this process, two limitations are particularly noteworthy. First, we used a relatively small sample of usability practitioners. Although these users represented five different organizations, they may not be representative of the larger population of usability practitioners. Second, we pre-selected the 15 usability case descriptions from a larger database containing over 100 usability problem cases. Such a distillation process may have restricted the scope of fully testing the reliability of the entire framework.

6. Future work

The strong reliability results from the User Action Framework have allowed us to proceed in developing other usability engineering support tools with the knowledge that these new tools share a reliable framework in terms of how experts generally think about usability problems.

As implemented in this reliability study, the User Action Framework was built as a local Website, with all relationships represented by hypertext links. Future plans call for implementing the User Action Framework as a database with all relationships represented as relations within the database. By implementing as a database, we can easily map to various support tools simply by adding new fields to each database node without changing the inherent nature of the User Action Framework structure. In each mapping the structure and the content of each User Action Framework node is retained.

In addition, each tool is represented by an associated database field containing an expression of the node content tailored to the purpose of that tool. Moving from the User Action Framework to each tool is merely a matter of looking at multiple fields in the same relational database record.

One of the first tools under development is the Usability Problem Inspector, the fields of which contain expressions in terms of the types of problems to look for in a usability inspection. The database approach to node descriptions easily facilitates filtering of the tool content to greatly improve cost effectiveness by focusing each inspection instance. For example, a user of the Usability Problem Inspector may apply a filter for a particular user class, thus tailoring the inspection to include only User Action Framework nodes tagged with attributes relating to expert users, thereby using only inspection questions that may be relevant for a system that was primarily for expert users instead of novices. The Usability Problem Inspector can also allow for tailoring to various levels of interface development maturity. For example, an evaluator may inspect an early paper prototype where detailed object design questions are not relevant, and such detailed questions can be filtered out and not included in the inspection. Future studies will investigate the effectiveness of the Usability Problem Inspector as compared to other methods such as the cognitive walkthrough and the heuristic evaluation technique.

When using these mappings to tools in combination with mappings to various interaction styles, the database can yield expressions to accommodate both the purpose of a tool and the interaction style to which it is being applied. For example, mappings to a usability inspection tool for application to virtual reality interfaces will translate User Action Framework content into a structured set of questions about potential usability problems with emphasis on navigation and other features appropriate to virtual environments. By combining purpose and interaction style mappings, the user is able to apply a tool that is specifically focused on the nature of the target system and the specific objectives of the evaluation.

In terms of the potential limitations of the study, we also propose expanding the experimental base to include more practitioners and more types of usability problems. Future studies could even investigate the various modes of presenting usability case descriptions. In our implementation of the summative study, we used written descriptions of usability problems. A study to compare written with video-taped usability problems would further expand the scope of the reliability claim.

7. Conclusion

Because it is a knowledge base of usability concepts and issues and built upon a theory-based model adapted from Norman (1986), the User Action Framework provides a reliable framework for usability engineering support tools. Results from our reliability analysis of the User Action Framework provide the breakthrough we have been looking for, removing the barrier to continuing work on other usability engineering support tools. From the User Action Framework, we plan to map to various usability tools in support of activities such as inspection, classification, usability problem data maintenance and interaction design guidelines. We expect that the User Action Framework, and associated mappings, will provide usability professionals with comprehensive tools to conduct a more efficient and effective usability evaluation, analysis, description, and

reporting process through an easily understood framework, a complete way to understand problems, and built-in links to possible design solutions.

References

BODKER, S. (1991). *Through the Interface: A Human Activity Approach to User Interface Design.* Hillsdale, NJ: Lawrence Erlbaum Associates.

BROOKS, P. (1994). Adding value to usability testing. In J. NIELSEN & R. L. MACK, Eds., *Usability Inspection Methods*, pp. 255–271. New York: John Wiley & Sons.

CARROLL, J. M., KELLOGG, W. A. & ROSSON, M. B. (1991). The task-artifact cycle. In J. M. CARROll, Ed., *Designing Interaction: Psychology at the Human-Computer Interface*, pp. 74–102. Cambridge, UK: Cambridge University Press.

COCKTON, G. & LAVERY, D. (1999). A framework for usability problem extraction. In *Proceedings of the IFIP 7th International Conference on Human–Computer Interaction—INTERACT '99,* pp. 344–352. London: IOS Press.

COHEN, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, **20,** 37–46.

CUOMO, D. L. (1994). A method for assessing the usability of graphical, direct-manipulation style interfaces. *International Journal of Human-Computer Interaction*, **6,** 275–297.

CUOMO, D. L. & BOWEN, C. D. (1992). Stages of user activity model as a basis for user-system interface evaluations. In *Proceedings of the Human Factors Society 36th Annual Meeting*, pp. 1254–1258. Santa Monica, CA: Human Factors and Ergonomics Society.

DESURVIRE, H. W. (1994). Faster, cheaper! Are usability inspection methods as effective as empirical testing? In J. NIELSEN & R. L. MACK, Eds., *Usability Inspection Methods*, pp. 173–202. New York: John Wiley & Sons.

DOUBLEDAY, A., RYAN, M., SPRINGETT, M. & SUTCLIFFE, A. (1997). A comparison of usability techniques for evaluating design. In *Designing Interactive Systems* (*DIS '97*) *Conference Proceedings*, pp. 101–110. New York: ACM Press.

DUTT, A., JOHNSON, H. & JOHNSON, P. (1994). Evaluating evaluation methods. In G. COCKTON, S. W. DRAPER & G. R. S. WEIR, Eds., *People and Computers IX*, pp. 109–121. Cambridge, UK: Cambridge University Press.

FLEISS, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, **76,** 378–382.

GARZOTTO, F., MATERA, M. & PAOLINI, P. (1998). Model-based heuristic evaluation of hypermedia usability. In *Proceedings of the Working Conference on Advanced Visual Interfaces—AVI '98,* pp. 135–145. New York: ACM Press.

GOOD, M., WHITESIDE, J., WIXON, D. & JONES, S. (1984). Building a user-derived interface. *Communications of the ACM*, **27,** 1032–1043.

GRAY, W. D. & SALZMAN, M. C. (1998). Damaged merchandise? A review of experiments that compare usability evaluation methods. *Human–Computer Interaction*, **13,** 203–261.

HARTSON, H. R., ANDRE, T. S., WILLIGES, R. W. & VAN RENS, L. (1999). The user action framework: a theory-based foundation for inspection and classification of usability problems. In H. BULLINGER & J. ZIEGLER, Eds., *Human–Computer Interaction: Ergonomics and User Interfaces*, Vol. 1, pp. 1058–1062. Mahway, NJ: Lawrence Erlbaum.

HIX, D. & HARTSON, H. R. (1993). *Developing User Interfaces: Ensuring Usability through Product and Process.* New York: John Wiley & Sons.

JEFFRIES, R. (1994). Usability problem reports: helping evaluators communicate effectively with developers. In J. NIELSEN & R. L. MACK, Eds., *Usability Inspection Methods*, pp. 273–294. New York: John Wiley and Sons.

JEFFRIES, R., MILLER, J. R., WHARTON, C. & UYEDA, K. M. (1991). User interface evaluation in the real world: a comparison of four techniques. In *CHI* '91 *Conference Proceedings*, pp. 119–124. New York: ACM Press.

JOHN, B. E. & MARKS, S. J. (1997). Tracking the effectiveness of usability evaluation methods. *Behaviour and Information Technology*, **16,** 188–202.

KARAT, C. (1994). A comparison of user interface evaluation methods. In J. NIELSEN & R. L. MACK, Eds. *Usability Inspection Methods*, pp. 203–233. New York: John Wiley & Sons.

KAUR, K., MAIDEN, N. K. & SUTCLIFFE, A. (1999). Interacting with virtual environments: an evaluation of a model of interaction. *Interacting with Computers*, **11,** 403–426.

KEENAN, S. L. (1996). *Product usability and process improvement based on usability problem classification*. Unpublished doctoral dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA.

KEENAN, S. L., HARTSON, H. R., KAFURA, D. G. & SCHULMAN, R. S. (1999). The usability problem taxonomy: a framework for classification and analysis. *Empirical Software Engineering*, **4,** 71–104.

LAVERY, D., COCKTON, G. & ATKINSON, M. P. (1997). Comparison of evaluation methods using structured usability problem reports. *Behaviour and Information Technology*, **16,** 246–266.

LIM, K. H., BENBASAT, I. & TODD, P. (1996). An experimental investigation of the interactive effects of interface style, instructions, and task familiarity on user performance. *ACM Transactions on Computer–Human Interaction*, **3,** 1–37.

MAYHEW, D. J. (1992). *Principles and Guidelines in Software User Interface Design*. Englewood Cliffs, NJ: Prentice-Hall.

MEISTER, D. (1985). *Behavioral Analysis and Measurement Methods*. New York: John Wiley & Sons.

MULLER, M. J., DAYTON, T. & ROOT, R. (1993). Comparing studies that compare usability assessment methods: an unsuccessful search for stable criteria. In *INTERCHI '93 Conference Proceedings* (*Adjunct*), pp. 185–186. New York: ACM Press.

NIELSEN, J. (1993). *Usability Engineering*. Boston: Academic Press.

NIELSEN, J. (1994*a*). Enhancing the explanatory power of usability heuristics. In *CHI* '94 *Conference Proceedings*, pp. 152–158. New York: ACM Press.

NIELSEN, J. (1994*b*). Heuristic evaluation. In J. NIELSEN & R. L. MACK, Eds. *Usability Inspection Methods*, pp. 25–62. New York: John Wiley and Sons.

NIELSEN, J. (1994*c*). UPA 93 trip report. *ACM SIGCHI Bulletin*, **26,** 29–32.

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

NIELSEN, J. & MOLICH, R. (1990). Heuristic evaluation of user interfaces. In *CHI '90 Conference Proceedings*, pp. 249–256. New York: ACM Press.

NORMAN, D. A. (1986). Cognitive engineering. In D. A. NORMAN & S. W. DRAPER, Eds. *User Centered System Design: New Perspectives on Human-Computer Interaction*, pp. 31–61. Hillsdale, NJ: Lawrence Erlbaum Associates.

PAYNE, S. J. (1991). Display-based action at the user interface. *International Journal of Man–Machine Studies*, **35,** 275–289.

RIZZO, A., MARCHIGIANI, E. & ANDREADIS, A. (1997). The AVANTI project: prototyping and evaluation with a cognitive walkthrough based on the Norman's model of action. In *Designing Interactive Systems* (*DIS '97*) *Conference Proceedings*, pp. 305–309. New York: ACM Press.

RUBIN, J. (1994). *Handbook of Usability Testing*. New York: John Wiley & Sons.

SEARS, A. (1997). Heuristic walkthroughs: finding the problems without the noise. *International Journal of Human–Computer Interaction*, **9,** 213–234.

SHNEIDERMAN, B. (1998). *Designing the User Interface: Strategies for Effective Human–Computer Interaction* (3rd edn.). Reading, MA: Addison-Wesley.

SUCHMAN, L. (1987). *Plans and Situated Action: The Problem of Human–Machine Communication*. New York: Cambridge University Press.

SUTCLIFFE, A., RYAN, M., SPRINGETT, M. & DOUBLEDAY, A. (1996). *Model Mismatch Analysis: Towards a Deeper Evaluation of Users' Usability Problems*. School of Informatics Report, City University, London.

VAN RENS, L. S. (1997). *Usability problem classifier*. Unpublished master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, VA.

VORA, P. (1995). Classifying user errors in human–computer interactive tasks. *Usability Professional's Association Common Ground*, **5,** 15.

WELLER, H. G. & HARTSON, H. R. (1992). Metaphors for the nature of human–computer interaction in an empowering environment: interaction style influences the manner of human accomplishment. *Computers in Human Behavior*, **8,** 313–333.

Paper accepted for publication by Associate Editor, Dr Saul Greenberg.