BEHAVIOUR & INFORMATION TECHNOLOGY, 1997, VOL. 16, NO. 4/5, 267–278

A proposed index of usability: a method for comparing the relative usability of different software systems

HAN X. LIN, YEE-YIN CHOONG and GAVRIEL SALVENDY

School of Industrial Engineering, Purdue University, West Lafayette, Indiana 47907, USA; email salvendy@ecn.purdue.edu

Abstract. Usability is becoming a more and more important software criterion, but the present usability measurement methods are either difficult to apply, or overly dependent upon evaluators' expertise. Based on human information processing theory, this study identified eight human factors considerations which are relevant to software usability. These considerations as well as the three stages of human information processing theory formed the framework from which our Purdue Usability Testing Questionnaire (PUTQ) is derived. An experiment was conducted to test the validity of PUTQ. The experiment result showed high correlation between PUTQ and the Questionnaire for User Interaction Satisfaction (QUIS version 5.5). In addition, PUTQ detected the differences in user performance between two experimental interface systems, but QUIS failed to do so.

1. Introduction

Loosely and intuitively defined, usability is the ease with which a software product can be used to perform its designated task by its users at a specific criterion. It becomes a major factor of interactive software products for millions of ordinary computer users who have never received any formal training in computer technology. Unfortunately, no tools exist for these end users to compare the usability of different software systems. This paper proposes a method for comparing the relative usability of different software systems. It can be used by users to help them make purchase decisions; it can be used by some third party organizations to issue usability evaluation reports for various kinds of software products in the market; it can also be used by software companies to evaluate their own products in their usability engineering processes.

The article is organized as follows. Section 2 reviews existing tools or methods for measuring software system usability. In section 3, we propose the theorectical framework from which our Purdue Usability Testing Questionnaire is derived. Section 4 introduces PUTQ in detail. Section 5 describes the experiment we have conducted to test the validity of PUTQ. In section 6 we report the experimental results and section 7 discusses the limitations of PUTQ. Section 8 concludes this article.

2. Literature review

The term of usability seems to be defined elusively by different researchers (Ravden and Johnson 1989, Eason 1984, Chapanis 1991, Shackel 1991, Preece 1993). But the definition given by Shackel seems to be accepted by most researchers in the area of Human Computer Interaction (HCI). Shackel (1991) defined usability as '. . . *the capability in human functional terms to be used easily and effectively by the specified range of users, given specified training and user support, to fulfil the specified range of tasks, within the specified range of environmental scenarios*'. In its simple form of usability definition given by Preece (1993), it was stated as '*The goals of HCI are to develop and improve systems that include computers so that users can carry out their tasks: safely, effectively, efficiently and enjoyably. These aspects are collectively known as usability*'.

Compared to the definition introduced above, usability is defined in a more operational sense in ISO (1993): '. . . *the quality of use: the effectiveness, efficiency and satisfaction with which specified users achieve specified goals in particular environments*'. Measures of effectiveness relate the goals or sub-goals of using the system to the accuracy and completeness with which these goals can be achieved (Bevan and Macleod 1994). Effectiveness is highly related to the completeness of the functionality of the software and the effectiveness of each functionality. For example, if a word processor can

not be used to edit a file, its effectiveness will be very low. Measures of efficiency relate the level of effectiveness achieved to the expenditure of resources. If using one word processor to edit a file is faster than using another word processor to edit the same file, the former will have higher efficiency than the latter. Measures of satisfaction describe the perceived usability of the overall system by its users and the acceptability of the system by the people who use it and by other people who are affected by its use.

There exist several approaches which can be used to evaluate interface usability. Perhaps the approach adopted most often is usability testing. Usability testing is based upon the principle of trailing prototypes and capturing data about the system and user performance which can then be analysed. Many user tests take place in specially equipped usability laboratories (Nielsen 1994). Usually such a laboratory has two rooms, one for experiments, the other for observation (Nielsen 1993). Typically, there will be sound-proof, one-way mirrors separating the observation room from the test room to allow the experimenters to discuss user actions without disturbing the subjects. Usability testing can be conducted to get information about user execution time, accuracy, users' satisfaction, as well as video tapes and system logs. Some tools have been developed for user-based measures of usability in the laboratory and the field: user performance, user satisfaction, and cognitive workload (Bevan and Macleod 1994). In order for usability testing to have high validity, subjects must be selected to represent the intended user group, usability testing must be conducted properly, and usability data must be carefully analysed. The advantage of this method is that the results are directly from the users. It can identify serious and recurring problems and avoid low-priority problems. It also has some degree of objectivity. On the other hand, it usually requires user-interface expertise and large samples of users to obtain reliable data, it will be difficult to find individual usability problems and, most of all, it is costly for the laboratory settings.

Another usability testing approach which also collects data directly from users in the thinking-aloud approach or protocol analysis. Basically, this method asks subjects to think aloud while they use the system to be evaluated. By verbalizing their thoughts, the test users enable the experimenters to know how users view the computer system, and how they interpret each individual interface item. Protocol analysis makes it easy for the experimenters to identify the users' major misconceptions and the components of the system which cause these misconceptions. However, thinking aloud seems very unnatural to most people, and some test users have great difficulties in keeping up a steady stream of utterances as they use a system and, for experienced users or experts, it is difficult for them to verbilize the detail of their operational processes.

Still another approach that has some degree of objectivity is formal modelilng. Research in formal modelling aims at developing theories on interaction of human with interfaces in order to structure more objective techniques for the evaluation process. Many models have been proposed in this vein. The GOMS model of Card *et al.* (1983), is a method for classifying the skills necessary to use an interface in terms of Goals, Operators, Methods and Selection rules. One of its variations NGOMSL, which stands for the Natural GOMS Language, was developed by Kieras (1988) to enable the task analysis using a GOMS-like model to be more specific. The purpose of these models is to describe idealized, error-free behaviour in terms of those four concepts (goals, operators, methods, and selection rules) and to provide predictions for the time taken to execute given tasks. The NGOMSL can also be used to estimate minimal learning time, and to quantify different aspects of mental workload. TAG, which stands for Task-Action-Grammars, was developed by Payne and Green (1986) and is a method of describing interfaces in terms of the linguistic structure of the commands. Particularly, TAG requires an existing interface specification for evaluation and has little to say about non-expert performance. It is mainly focused on revealing internal inconsistency existing in the interface mechanisms rather than user performance. One recent formal method was proposed by Thimbelby (1994). In that, it suggested treating user interfaces as finite state machines, and using graph theory as an analytic framework for a given system definition. It can be used to predict users' learning time, expert users' execution time, etc. In general, this approach is objective, it can provide quantitative analysis of interface properties, and it is based on the design specifications rather than the product or prototype. On the other hand, it is complex and requires expertise to apply. It tends to focus on only one dimension, usually high-level, of the interface to be evaluated.

Other less formal usability evaluation methods include guidelines/checklists and heuristic methods. Many computer companies like IBM (1991), Apple Computer, Inc. (1992), Open Software Foundation, Inc. (1993) have their own interface guidelines provided to the interface designers to follow. The most widely cited one is by Smith and Mosier (1986). More guidelines were proposed by Marshall *et al.* (1987), Brown (1988), and Carlow International, Inc. (1992). Guidelines can be very useful in guiding user interface design. For example, a software system which follows Macintosh guidelines can be consistent with other Macintosh

programs. Although guidelines are very popular, they also have been criticized for being hard to follow. Especially for general guidelines such as Smith and Mosier (1986), they can mean different things to different people. Another problem with guidelines is that they have so many detailed items. This makes it even harder for software designers to follow them. Even worse, they might be misapplied to defend bad designs or even create new or more usability problems.

Some usability specialists therefore proposed another method that is easier to follow: heuristic evaluation. Heuristic evaluation is the application, by evaluators of varying degrees of expertise, of a set of heuristics to judge the adequacy of a design prototype. Nielsen and Molich (1990) proposed ten usability heuristics: (1) simple and natural dialogue; (2) speak the user's language; (3) minimize user memory load; (4) consistency; (5) provide feedback; (6) clearly marked exits; (7) shortcuts; (8) good error messages; (9) prevent errors, and (10) help and documentation. It was reported that the heuristic evaluation approach is the most effective among several evaluation methods in identifying large number of usability problems and serious problems (Jeffries *et al*. 1991, Virzi *et al*. 1993). There will be no extra cost involving the recruitment of subjects other than the cost for in-house expert evaluators. However, since it does not involve real users, it has some degree of subjectivity and, it requires several evaluators with some degree of expertise to perform the evaluation which limits its application power.

Although there are many usability evaluation methods, they all have some disadvantages as summarized in Table 1 (adapted from Nielsen 1993, Lansdale and Ormerod 1994). They are either difficult to apply, or dependent upon the evaluators' expertise. To compensate for the disadvantages of these existing evaluation methods, this study aims to develop a quantitative usability measurement tool which will be effective, easy to apply, low-cost, and involve fewer subjects. The instrument which we have developed is different from the Questionnaire for User Interaction Satisfaction (QUIS) in that the QUIS is mainly for measuring users' satisfaction toward a user interface while our instrument tries to measure the usability of software systems, not only users' satisfaction.

3. Theoretical framework

In order to develop a usability measurement tool, it is very important to have a theory-based framework to ensure the validity and reliability of the proposed instrument. One relevant study was done by Reiterer and Oppermann (1993). In this study, two dimensions were identified as important issues in evaluating user interfaces: technical system components and software-ergonomic criteria. In the technical system components dimension, four main categories were included: input/output interface, dialogue interface, functional interface, and organizational interface. Several criteria have been addressed in the second dimension, namely; availability, suitability for the task, clearness, self-descriptiveness, conformity with user expectation, error tolerance, suitability for learning, suitability for individualization, controllability, cooperation and communication, and finally, data protection.

Table 1. Summary of advantages and disadvantages of different usability evaluation methods.

Method name	Advantages	Disadvantages
Laboratory testing	• Identify serious problems; • Identify recurring problems; • Avoid low-priority problems; • Some degree of objectivity.	◆ Require expertise; ◆ High cost; ◆ Need large number of users; ◆ Miss consistency problems.
Thinking aloud	• Pinpoints user misconceptions; • Low cost.	◆ Unnatural to users; ◆ Hard for expert users.
Formal modelling	• Quantitative analysis; • Give unexpected insight; • Some degree of objectivity.	◆ Extremely complex; ◆ Require expertise; ◆ Tend to focus on one dimension.
Guidelines/checklists	• Identify general problems; • Identify recurring problems; • Can be used by non-specialists; • Applicable at all design stages.	◆ Miss some severe problems; ◆ Might be misapplied; ◆ Difficult to follow.
Heuristic evaluation	• Identify many problems; • Identify more serious problems; • Low cost; • Predict further evaluation needs.	◆ Require expertise; ◆ Require several evaluators; ◆ Some degree of subjectivity.

Source: Modified from Nielsen (1993) and Lansdale and Ormerod (1994).

3.1. *Assumptions*

The framework of this study is based on two assumptions: (1) user satisfaction is correlated with other usability measures, namely; effectiveness, efficiency, and learnability; (2) design features of a software affect is effectiveness, efficiency and learnability. Some studies were devoted to measuring users' satisfaction with computer interfaces through carefully designed questionnaires. In QUIS, developed by Norman and Shneiderman (1989), among the total 27 items in the questionnaire, 21 items are directly related to interface features. Another questionnaire SUMI (Software Usability Measurement Inventory) was developed (Kirakowski and Corbett 1993) to measure user satisfaction, and hence to assess user perceived software quality. SUMI provides three types of measures: an overall assessment, a usability profile, and item consensus analysis. In the usability profile, five sub-scales are included: affect, efficiency, helpfulness, control, and learnability. Again, it is illustrated clearly that user satisfaction is related to other usability measures such as efficiency and learnability. For the second assumption, many published researches have shown measurable and significant performance advantages of some design alternatives over the other (Mayhew and Mantei 1994). For example, McDonald *et al*. (1983) did research on comparing random ordering of menu item vs. categorical ordering in a simple search/select task. Their results showed 4 seconds difference in performance time. Another study (Kacmar and Carey 1991) compared user interfaces constructed of text, pictorial, or text-and-pictorial formats. Results showed that the interface constructed of a mixed format (text and pictorial) resulted in the fewest number of incorrect user selections. These experimental results provide support for feature-based software evaluation.

3.2. *Human information processing*

The process of human interaction with computer interfaces is mainly information processing. We need to investigate how humans process information in order to understand the interaction between humans and computers, and hence to develop usable computer interfaces for humans. The three-stage model is probably the most common one used to describe human information processing (Proctor and Van Zandt 1994).

The perceptual stage includes processes involved in the detection, discrimination, and identification of displayed information. If the display is not clear or is not distinguishable, much of the information will be lost. After information has been extracted from a display to allow the identification or classification of the stimuli, processes begin to operate with the goal of determining the appropriate action or response. These processes include the retrieval of information from long-term memory, comparisons among displayed items, comparison between displayed items and information stored in memory, and decision making. In this cognitive stage, such constraints as the limited abilities to attend to multiple sources of information, to retrieve information from memory, to perform complicated mental calculations, and the limited capacity of the short-term memory greatly affect human performance. Errors in performance may arise due to these and other cognitive limitations. Following the perceptual and cognitive stages, an overt response (if required) needs to be selected, prepared, and executed.

3.3. *Human-computer interaction (HCI) considerations*

Based on the theory of human information processing as well as the studies reviewed previously concerning interface usability, we have identified eight human factors criteria that are relevant to HCI. These criteria are compatibility, consistency, flexibility, learnability, minimal action, minimal memory load, perceptual limitation, and user guidance.

3.3.1. *Compatibility (CP):* Stimulus-response (S-R) compatibility refers to the phenomenon that the subjects' responses were faster and more accurate for the pairings of stimulus sets and response sets that corresponded naturally than for those that did not (Fitts and Seeger 1953). However, the compatibility effects depend not only on physical correspondence, but also on conceptual correspondence. Compatibility effects will occur whenever implicit or explicit spatial relationships exist among stimuli and among responses (Proctor and Van Zandt 1994). By attributing compatibility effects to the cognitive codes used to represent the stimulus and response sets, central cognitive processes must be responsible for the effects. Wickens *et al*. (1983) used the term S-C-R compatibility to emphasize the central processes (C). The mediating processes (cognitive processes) can be conceived more generally as reflecting the operator's mental model of the task. The implication of compatibility effects on HCI is that the user performance will be hindered if the displayed information and response required is not compatible.

3.3.2. *Consistency (CS):* In human–computer interaction, consistency is recognized to be able to improve user performance and user satisfaction. According to Grudin (1989), consistency is twofold: internal consis-

tency and external consistency. Internal consistency is the consistency within a system, whereas external consistency means the consistency among different systems. A good interface design should consider both internal consistency and external consistency.

3.3.3. *Flexibility (FX):* Here by flexibility we mean that an interface should be able to adapt to users' needs. The adaptation can be made by users' customization or the interface itself, according to Gong and Salvendy (1994). It is important to ensure good flexibility of an interface since different users may have different needs due to different levels of skill and users' needs may change according to the improvement of their skills over time and through experience.

3.3.4. *Learnability (LN):* A well-designed computer software should be easy to learn. Humans can learn through several formats such as rote learning, learning through understanding, or learning by exploration. The learning process will be enhanced and the result will be retained if users are presented with a well-designed, well-organized interface.

3.3.5. *Minimal action (MA):* This criterion is quite straightforward. Users should be required a minimal number of actions to perform a certain task in order to increase users' efficiency and satisfaction.

3.3.6. *Minimal memory load (MML):* Working memory load is one important aspect of mental workload consideration. Many studies have focused on how to analyse human memory load, and hence tried to decrease memory load to improve performance and satisfaction. One of those is done by Bi and Salvendy (1994) which has supported the inverted U shaped relationship between human mental workload and performance. To ensure minimal working memory load will increase human performance. Minimal long-term memory load requirement will help users learn interface more easily. The less that users need to learn, the faster users can learn it.

3.3.7. *Perceptual limitation (PL):* A good human computer interface should embed the considerations of human perceptual organization limitations. The perceptual organization is the process by which humans apprehend particular relationships among potentially separate stimulus elements (Boff and Lincoln 1988). The most basic is figure-ground organization. The principles of *Gestalt grouping* are especially important in display design in which they describe how elements might be perceived as a group. The principle of proximity is that elements close together in space tend to be perceived as a group. The principle of similarity is that similar elements (in terms of form, colour, or orientation) tend to be grouped together perceptually. The principle of continuity refers to the phenomenon that points connected in straight or smoothly curving lines tend to be seen as belonging together. The principle of closure refers to a tendency for open curves to be perceived as complete forms. Finally, the principle of common rate is that elements moving in a common direction at a common speed are grouped together. Whenever colours will be used in design, the limitations of human colour perception need to be considered. Spatial resolution is another example of the HCI considerations for perceptual limitation. Spatial resolution is the minimal visual angle for a detail that can be resolved by a human being.

3.3.8. *User guidance (UG):* In general, a computer system with a good user guidance scheme will improve the learnability of the system as well as decrease the mental workload of the users since no extra effort will be needed for the users to perform designated tasks.

In considering the usability of computer interfaces, one needs to investigate the application of HCI considerations at each stage of human information processing. It would be appropriate to focus on consistency, compatibility and perceptual limitation considerations at the perceptual stage of human information processing. At the cognitive state, it would be necessary to take into account consistency, compatibility, flexibility, minimal memory load, learnability and user guidance. Finally, at the action stage, it would be proper to consider the consistency, compatibility and minimal action. The discussion is summarized in Table 2.

4. Development of the intelligence index of interface usability

Based on the theory of human information processing, we identified eight human factors principles that are relevant to HCI. Most of these principles are supported by previous empirical studies. Having considered human-computer interaction task requirements, we then collected many user interface guidelines, questionnaire items from the literature (Apple Computer Inc. 1992, Brown 1988, Carlow International Inc. 1992, ISO 1993, Kirakowski and Corbett 1993, Marshall *et al.* 1987, Norman and Shneiderman 1989, Open Software Foundation Inc. 1993, Smith and Mosier 1986). From these items, we developed a preliminary usability questionnaire containing 100 items (Appendix A). Those items are classified according to the HCI considerations with the considerations of human limita-

Table 2. Stages of human information processing in HCI.

Human information processing	Perceptual stage	Cognitive stage	Action stage
Compatibility	✓	✓	✓
Consistency	✓	✓	✓
Flexibility		✓	
Learnability		✓	
Minimal action			✓
Minimal memory load		✓	
Perceptual limitation	✓		
User guidance		✓	

tions embedded. Sample items for each HCI considerations are given as follows:

1. Compatibility (CP)
   Are the results of control entry compatible with user expectations?
   Note: Ensure that the results of any control entry are compatible with user expectations, so that a change in the state or value of a controlled element is displayed in an expected or natural form.
2. Consistency (CS)
   Internal Consistency:
   Is the wording consistent across displays?
   Note: Consistent wording for:
   Displayed data.
   Labels.
   Tables.
   Text.
   External consistency:
   Is colour coding consistent with conventional colour meaning?
   Note: Conventional colour meaning such as:
   - Red: alarm.
   - Yellow: warning.
   - Green: normal.
3. Flexibility (FX)
   Are users allowed to customize windows?
   Note: Users should be able to change window attributes such as: font, size of the window, foreground/background colour, etc.
4. Learnability (LN)
   Is the ordering of menu options logical?
   Note: Organize menu options according to their: functionality, expected frequency of use, alphabetical ordering.
5. Minimal action (MA)
   Is is easy to fill out data entry form?
   Note:
   - Place the cursor at the first data entry field.
   - If possible, provide default data.
   - Advance the cursor automatically to the next data entry field.
6. Minimal memory load (MML)
   How are abbreviations and acronyms used?
   Note: If possible, do not use abbreviations and acronyms at all.
   - If it is necessary to use abbreviations and acronyms, use standard ones.
   - If there are no standard ones, use a consistent rule to generate them throughout the system.
7. Perceptual limitation (PL)
   Does it provide eaisiy distinguished colours?
   Note: Keep the number of colours used simultaneously no more than 4.
   - avoid using the following colour-pairs on the same screen: red and blue, red and green, blue and green.
   - avoid using the following colour-pairs as fore - ground/background colours: yellow on white; yellow on purple; yellow on green; green on white; blue on black; red on black; magenta on black; magenta on green.
8. User guidance (UG)
   System Feedback: How helpful is the error message?
   Note: Error messages should be specific to the point, explain what is wrong, and how it can be corrected. System should be able to give more detailed error message upon users' request.

A sample of the answer sheet of our questionnaire is shown in Appendix B.

The intelligence index will be calculated according to the equation shown as follows:

$$\text{Index} = \frac{\sum w_i \times (\text{Score}_i - \text{Penalty}_i)}{7 \times \sum w_i \times \text{Item}_i} \times 100 \qquad (1)$$

where: i = the ith item.

Score_i = the rating score of item i.

Penalty_i = 1, if the item i is applicable but not available (N/A).

Item_i = 0, if the item i is not applicable.
1, if the item i is applicable.
0, if the item i is not applicable.

w_i = weighting of the importance of item i.

For each item (Item_i), first of all, the evaluator needs to decide whether this item is applicable to the computer interface to be evaluated. If it is not

applicable, then $Item_i$ will be assigned as '0', and then proceed to the next item. If on the other hand, the item is applicable, a '1' will be assigned to $Item_i$. Next, the evaluator will examine if the applicable item is available or not in the system. If 'not available' (N/A) has been determined by the evaluator, then a penalty will be associated with this item which means there will be a possible usability problem according to the lack of this item. If it is applicable and available, then the evaluator will rate the interface on that item on a scale from 1 to 7 with 1 as the worst case and 7 as the best case. Meanwhile, the evaluator will give a weight to the item concerning the degree of importance of the item on a scale from 1 to 3 with higher number associated with higher importance.

The intelligence index of the usability for a specific interface will be the ratio of the total score obtained by the evaluation to the possible perfect score and the ratio will be transformed using the scale of 100.

An assessment of objective usability can be made through PUTQ, a subjective method as long as PUTQ has high reliability and validity. 'Reliability refers to the consistency of scores obtained by the same persons when reexamined with the same test on different occasions, or with different sets of equivalent items, or under other variable examining conditions.' (Anastasi 1976: 103). 'The validity of a test concerns what the test measures and how well it does so.' (Anastasi 1976: 134). There are three kinds of validity: content validity, construct validity, and criterion-based validity (Anastasi 1976). PUTQ has good construct validity and content validity since it is derived from the experimental and theoretical base and its items were selected from lots of user interface guidelines and questionnaire items. An experiment which is reported below was conducted to test its reliability and criterion-based validity. Criterion-based validity was tested by testing the following hypothesis:

PUTQ score for high usability software is greater than that for low usability software.

5. Method

5.1. *Subjects*

Eight graduate students from an accredited American university majoring in engineering and sciences participated as subjects. All the subjects came from mainland China and have resided in USA for less than one year. The age of these subjects ranges from 23 to 27 ($\bar{X}$ = 24.6, SD = 1.51). Four subjects were female and four male.

5.2. *Apparatus*

An IBM-based notebook computer (Toshiba T4700CT) was used for this study. It has 8M memory; its CPU is 80486; it has a 9 inch colour monitor and a standard two-button mouse.

5.3. *Task*

Two experimental systems which were designed for the study of impacts of cultural differences on human computer interface design (Choong 1996) were used for this experiment. One system has abstract knowledge representation and functional strcuture (AF). It is a plain text description of the hierarchical organization of the information such as categories, sub-categories and items in the target system (on-line department store). The other system has concrete knowledge representation and thematic structure (CT). It is a metaphor analogous to a department store with several floors representing the categories. On each floor, there are sections standing for sub-categories under each category. The items displayed in each section are items in each sub-category. Given a shopping list, subjects were asked to find out the items on the list using both AF and CT.

In Choong's study (1996), for the Chinese subjects, CT was significantly better than AF in terms of performance time ($F_{1,36}$ = 4.45; p = 0.04; 18.5% better) and number of errors ($F_{1,36}$ = 18.38; p = 0.0001; 66% better).

5.4. *Experimental design*

A single factor (interface type), within-subject experiment design was used for this study. The two levels of interface type are AF and CT.

To reduce the effect of sex differences, the same number of female and male subjects were used in this experiment. Task order (AF first or CT first) and questionnaire order (QUIS first or PUTQ first) were also balanced among subjects in this experiment.

Dependent variables of this experiment are: performance time of the on-line shopping task, number of errors made during task performance, PUTQ score, and QUIS score. The detailed description of the first two dependent variables can be found in Choong (1996). PUTQ score is calculated from subjects' response to PUTQ using our scoring formula (Equation 1). A raw QUIS score is obtained from the subjects' response to QUIS 5.5 using its scoring scheme. This raw QUIS score, after a simple transformation (Equation 2), becomes the final QUIS score used in this study.

$$\text{QUIS score} = \frac{\text{Raw\_QUIS\_score}}{(\text{Number\_of\_applicable\_items}) \times 9} \times 100 \quad (2)$$

where: Number_of_applicable_items is the number of items which were regarded as applicable to the system being evaluated, and 9 is the maximum score a system can get for each applicable item.

5.5. *Procedure*

The experiment was conducted in two consecutive days. In day one, subjects were asked to fill out a consent form. They were given on-line instructions of the shopping system. Then a brief practice session was conducted to help the subjects understand the operation of the system and the task to be performed. After subjects felt comfortable with the system, they were given a shopping list, and were asked to search out the items from the system. Depending on their treatment, subjects used either AF or CT interface in day one. After they performed the task, they were asked to fill out two questionnaires: QUIS and PUTQ. Again half of the subjects filled out QUIS first and the other half filled out PUTQ first.

In day two, subjects used another interface to do the same task. They were again asked to fill out QUIS and PUTQ after the search task. Finally subjects were paid for their participation.

Totally the experiment took about two hours.

6. Results and discussion

In order to assess the internal reliability of the usability tests, Cronbach Alpha was utilized for both tests. For QUIS, Cronbach Alpha varied from 0.61 to 0.92 with an average reliability level around 0.77. For PUTQ, the Cronbach Alpha values ranged from 0.59 to 0.81 with an average reliability of 0.70.

The hypothesis proposed at the end of Section 4 was tested by the first paired *t*-test in Table 3. It tests whether the PUTQ score between AF and CT are significantly different from each other. The test result showed that this difference is significant at 0.10 level. Since a previous study (Choong 1996) indicated that CT was significantly better than AF in terms of users' performance time and number of errors, which means that CT has higher usability than AF, our hypothesis was supported. PUTQ score for high usability software CT is significantly ($\alpha = 0.07$) higher than PUTQ score for low usability software AF.

The third and fourth paired *t*-tests reported in Table 3 indicated that the effect of interface type on subjects' performance time and number of errors is significant at the 0.05 level. This is consistent with the result of Choong's (1996) study. We did these two tests just for reconfirming that CT is indeed better than AF in terms of fewer user errors and less performance time.

The second paired *t*-test in Table 3 indicated that QUIS is not sensitive enough to differentiate between AF and CT because QUIS scores for AF and CT are not significantly different, but it correlated well ($r = 0.868$) with PUTQ.

Both QUIS and PUTQ were administered by the same researcher. The order of their administration (QUIS first or PUTQ first) was also balanced among subjects. It seems unlikely that their difference can be attributed to the difference in their administration. Because question items in PUTQ are organized around human factors principles and in QUIS they are organized around system components, detailed comparison between PUTQ and QUIS is not possible at this time. To do that, reorganization of PUTQ question items needs to be done first.

7. Limitations

The framework from which PUTQ is derived is based on human information processing theory. Theoretically it can be applied to any man-machine systems. But questionnaire items in PUTQ are mainly focused on conventional graphical user interface software which

Table 3. Paired *t*-test results.

	AF†		CT‡		AF-CT (Difference)		Paired	
	$\bar{X}$	SD	$\bar{X}$	SD	$\bar{X}$	SD	*t* test	*p* value
PUTQ	75.·85	17·12	80·89	14·09	—5·04	6·73	—2·116	0·07211
QUIS	83·44	8·75	85·22	7·12	—1·78	5·75	—0·874	0·41087
Time	248·14	68·36	193·37	65·01	54·78	54·28	2·854	0·02454
Errors	49·74	23·01	41·25	22·50	8·50	10·03	2·397	0·04765

†The system with abstract knowledge representation and functional structure.
‡The system with concrete knowledge representation and thematic structure.

requires visual display, keyboard, and mouse. Cautions should be exercised when using PUTQ to evaluate those interactive software systems which consist of sound or voice, animation, video, and pen-based computing components.

The usability measures considered in PUTQ were limited to traditional dimensions of usability. It is clear that less tangible issues such as pleasure or enjoyment are becoming more of a factor in the use of software systems. Further work needs to be done to incorporate such measures in PUTQ.

Our interest was to develop an instrument to be used by end-users to evaluate the usability of different interactive software systems. For resource reasons, our study focused on a special user population. In the future, we need to do more empirical tests involving many different users.

8. Conclusion

The proposed PUTQ questionnaire for usability testing has good construct validity and content validity since it is derived from the experimental and theoretical base outlined in the published literature. Its criterion-based validity and reliability in our experiment is also good. The full questionnaire is presented here so that it could be freely utilized by diversified groups of people on a variety of different computer software products. By so doing the questionnaire would be further evaluated and validated. The ultimate use of the usability questionnaire would be to derive usability index measures for each software and publish these in a similar way that consumer testing of consumer products is done in the USA. That would further increase the emphasis on user friendly design of computer software systems.

Acknowledgements

We gratefully acknowledge and appreciate the generous support of the NEC corporation in supporting the NEC professorship which made this study possible. We acknowledge the help of the anonymous referees, whose comments greatly improved this paper.

References

Anastasi, A. 1976, *Psychological Testing*. (4th ed.) (Macmillan, New York).

Apple Computer Iinc. 1992, *Human Interface Guidelines: The Apple DesktopInterface*. 2nd ed., (Addison-Wesley, Reading, MA).

Bevan, N. and Macleod, M. 1994, Usability measurement in context. In J. Nielsen (ed.) Usability Laboratories, special issue, *Behaviour & Information Technology*, **13**, 132 – 145.

Bi, S. and Salvendy, G. 1994, Analytical modeling and experimental study of human workload in scheduling of advanced manufacturing systems, *International Journal of Human Factors in Manufacturing*, **4,** 205 – 234.

Boff, K. R. and Lincoln, J. E. 1988, *Engineering Data Compendium: Human Perception and Performance*. (Harry G. Armstrong Medical Research Laboratory, Wright – Patterson AFB, OH).

Brown, C. M. 1988, *Human-Computer Interface Design Guidelines*. (Ablex, Norwood, NJ).

Card, S. K., Moran, J. P. and Newell, A. 1983, *The Psychology of Human Computer Interaction*. (Erlbaum, Hillsdale, NJ).

Carlow International Inc. 1992, *Human-Computer Interface Guidelines*, prepared for Data Systems Technology Division, Software and Automation Systems Branch, Goddard Space Flight Center, Falls Church, VA.

Chapanis, A. 1991, Evaluating usability. In B. Shackel and S. J. Richardson (eds), *Human Factors for Informatics Usability*, (Cambridge University Press, Cambridge), 359 – 396.

Choong, Y. 1996, *Design of computer interfaces for the Chinese population*. Unpublished Ph.D. dissertation, Purdue University, West Lafayette, IN.

Eason, K. D. 1984, Towards the experimental study of usability, *Behaviour & Information Technology*, **3,** 133 – 143.

Fitts, P. M. and Seeger, C. M. 1953, S – R compatibility: spatial characteristics of stimulus and response codes, *Journal of Experimental Psychology*, **46,** 199 – 210.

Gong, Q. and Salvendy, G. 1994, Design of skill-based adaptive interface: the effect of a gentle push. *Proceedings of the Human Factors and Ergonomics Society, 38th Annual Meeting* (HFES, Santa Monica, CA), 295 – 299.

Grudin, J. 1989, The case against user interface consistency. *Communications of the ACM*, **32,** 1164 – 1173.

International Business Machines Corporation, 1991, *Common User Access Guide to User Interface Design* (IBM, Cary, NC).

ISO 1993, ISO CDS 9241 – 11: *Guidelines for Specifying and Measuring Usability*.

Jeffries, R., Miller, J. R., Wharton, C. and Uyeda, K. M. 1991, User interface evaluation in the real world: a comparison of four techniques, *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems*, 119 – 124 (Association for Computing Machinery, New York).

Kacmar, C. J. and Carey, J. M. 1991, Assessing the usability of icons in user interfaces, *Behaviour & Information Technology*, **10,** 443 – 457.

Kieras, D. E. 1988, Towards a practical GOMS model methodology for user interface design. In M. Helander (ed.), *Handbook of Human-Computer Interaction*. (Elsevier, North Holland), 135 – 158.

Kirakowski, J. and Corbett, M. 1993, SUMI: the software measurement inventory, *British Journal of Educational Technology*, **24,** 210 – 212.

Lansdale, M. W. and Ormerod, T. C. 1994, *Understanding Interfaces: a Handbook of Human-Computer Dialogue* (Academic Press, San Diego, CA).

Marshall, C., Nelson, C. and Gardiner, M. M. 1987, Design guidelines. In M. M. Gardiner and B. Christie (eds), *Applying Cognitive Psychology to User-Interface Design*. (J Wiley & Sons, Chichester, U.K.), 221 – 276.

MAYHEW, D. J. and MANTEI, M. 1994, A basic framework for cost-justifying usability engineering. In R. G. Bias and D. J. Mayhew. (eds) *Cost-justifying Usability*, (Academic Press, Boston) 9–43.

MCDONALD, J. E., STONE, J. D. and LIEBELT, L. S. 1983, Searching for items in menus: the effects of organization and type of target. *Proceedings of the Human Factors Society 25th Annual Meeting*, (HFES, Santa Monica, CA) 12–22.

NIELSEN, J. 1993, *Usability Engineering*. (Academic, San Diego, CA).

NIELSEN, J. 1994, Usability laboratories, *Behaviour and Information Technology*, **13,** 8.

NIELSEN, J. and MOLICH, R. 1990, Heuristic evaluation of user interfaces. *Proceedings of ACM CHI '90 on Human Factors in Computing Systems*, Seattle, WA, 249–256 (Association for Computing Machinery, New York).

NORMAN, K. and SHNEIDERMAN, B. 1989, *Questionnaire for User Interaction Satisfaction (QUIS 5.0)*, (HCI Lab, College Park, University of Maryland).

OPEN SOFTWARE FOUNDATION INC. 1993, *OSF/Motif Style Guide*, (Prentice Hall, Englewood Cliffs, NJ).

PAYNE, S. J. and GREEN, T. R. G. 1986, Task-action grammars: a model of mental representation of task languages. *Human-Computer Interaction*, **2,** 93–133.

PREECE, J. (ed.) 1993, *A Guide to Usability: Human Factors in Computing*. (Addison-Wesley, Wokingham).

PROCTOR, R. W. and VAN ZANDT, T. 1994, *Human Factors: In Simple and Complex Systems*. (Allyn and Bacon, Needham Heights, MA).

RAVDEN, S. and JOHNSON, G. 1989, *Evaluating Usability of Human-Computer Interfaces*. (Ellis-Horwood, Chichester).

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

SMITH, S. L. and MOSIER, J. N. 1986, *Design Guidelines for Designing User Interface Software*. Technical Report MTR-10090 (The MITRE Corporation, Bedford, MA).

THIMBLEBY, H. 1994, Formulating usability, *SIGCHI Bulletin*, **26,** 59–64.

VIRZI, R. A., SORCE, J. F. and HERBERT, L. B. 1993, A comparison of three usability evaluation methods: heuristic, think-aloud, and performance testing. *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting*, (HFES, Santa Monica, CA) 309–313.

WICKENS, C. D., SANDRY, D. L. and VIDULICH, M. 1983, Compatibility and resource competition between modalities of input, central processing, and output. *Human Factors*, **25,** 227–248.

Appendix A: Purdue Usability Testing Questionnaire (PUTQ)*

Instruction: This questionnaire contains 100 questions about computer interfaces. They are grouped into eight parts. Please answer each of these questions regarding the system to be evaluated in the order they are given, using the answer sheet provided.

1. *Compatibility*
    1. Is the control of cursor compatible with movement?
    2. Are the results of control entry compatible with user expectations?
    3. Is the control matched to user skill?
    4. Are the coding compatible with familiar conventions?
    5. Is the wording familiar?

2. *Consistency*
    6. Is the assignment of colour codes conventional?
    7. Is the coding consistent across displays, menu options?
    8. Is the cursor placement consistent?
    9. Is the display format consistent?
    10. Is the feedback consistent?
    11. Is the format within data fields consistent?
    12. Is the label format consistent?
    13. Is the label location consistent?
    14. Is the labelling itself consistent?
    15. Is the display orientation consistent? — panning vs. scrolling.
    16. Are the user actions required consistent?
    17. Is the wording consistent across displays?
    18. Is the data display consistent with entry requirements?
    19. Is the data display consistent with user conventions?
    20. Are symbols for graphic data standard?
    21. Is the option wording consitent with command language?
    22. Is the wording consistent with user guidance?

3. *Flexibility*
    23. Does it have by-passing menu selection with command entry?
    24. Does it have direct manipulation capability?
    25. Is the design for data entry flexible?
    26. Can the display be controlled by user flexibly?
    27. Does it provide flexible sequence control?
    28. Does it provide flexible user guidance?
    29. Are the menu options dependent on context?
    30. Can user name displays and elements according to their needs?
    31. Does it provide good training for different users?
    32. Are users allowed to customize windows?

* Both the questionnaire and answer sheets are reproducible without permission provided this footnote is included in all copies used. Reproduced by permission from Han X. Lin, Yee-Yin Choong, and Gavriel Salvendy, A proposed index of usability: a method for comparing the relative usability of different software systems, *Behaviour & Information Technology*, 1997, Oct., pp.

33. Can users assign command names?
34. Does it provide user selection of data for display?
35. Does it handle user-specified windows?
36. Does it provide zooming for display expansion?

4. *Learnability*
37. Does it provide clarity of wording?
38. Is the data grouping reasonable for easy learning?
39. Is the command language layered?
40. Is the grouping of menu options logical?
41. Is the ordering of menu options logical?
42. Are the command names meaningful?
43. Does it provide no-penalty learning?

5. *Minimal action*
44. Does it provide combined entry of related data?
45. Will the required data be entered only once?
46. Does it provide default values?
47. Is the shifting among windows easy?
48. Does it provide function keys for frequent control entries?
49. Does it provide global search and replace capability?
50. Is the menu selection by pointing? — primary means of sequence control.
51. Is the menu selection by keyed entry? — secondary means of control entry.
52. Does it require minimal cursor positioning?
53. Does it require minimal steps in sequential menu selection?
54. Does it require minimal user control actions?
55. Is the return to higher-level menus required only one simple key action?
56. Is the return to general menu required only one simple key action?

6. *Minimal memory load*
57. How are abbreviations and acronyms used?
58. Does it provide aids for entering heirachic data?
59. Is the guidance information always available?
60. Does it provide hierarchic menus for sequential selection?
61. Are selected data highlighted?
62. Does it provide index of commands?
63. Does it provide index of data?
64. Does it indicate current position in menu structure?
65. Are data items kept short?
66. Are the letter codes for menu selection designed carefully?
67. Are long data items partitioned?
68. Are prior answers recapitulated?
69. Are upper and lower case equivalent?
70. Does it use short codes rather than long ones?
71. Does it provide supplementary verbal labels for icons?

7. *Perceptual limitation*
72. Does it provide coding by data category?
73. Is the abbreviation distinctive?
74. Is the cursor distinctive?
75. Are display elements distinctive?
76. Is the format for user guidance distinctive?
77. Do the commands have distinctive meanings?
78. Is the spelling distinctive for commands?
79. Does it provide easily distinguished colours?
80. Is the active window indicated?
81. Are items paired for direct comparison?
82. Is the number of spoken messages limited?
83. Does it provide lists for related items?
84. Are menus distinct from other displayed information?
85. Is the colour coding redundant?
86. Does it provide visually distinctive data fields?
87. Are groups of information demarcated?
88. Is the screen density reasonable?

8. *User guidance*
89. System feedback: How helpful is the error message?
90. Does it provide CANCEL option?
91. Are erroneous entries displayed?
92. Does it provide explicit entry of corrections?
93. Does it provide feedback for control entries?
94. Is HELP provided?
95. Is completion of processing indicated?
96. Are repeated errors indicated?
97. Are error messages non-disruptive/informative?
98. Does it provide RESTART option?
99. Does it provide UNDO to reverse control actions?
100. Is the sequence control user initiated?