Proceedings of ICAD2006
4th International Conference on Axiomatic Design
Firenze – June 13-16, 2006
ICAD-2006-20

USABILITY EVALUATION METHODS WITH POOR USABILITY: INSPECTION

Yee Soon Lim
limy0064@ntu.edu.sg
Nanyang Technological University,
50 Nanyang Avenue,
Singapore 639798

Martin G. Helander
martin@ntu.edu.sg
Nanyang Technological University,
50 Nanyang Avenue,
Singapore 639798

ABSTRACT

Usability inspection is commonly used in evaluation of computer systems user interfaces. Examples of inspection methods include: heuristic evaluation, guideline review, standards inspection, design rules, and cognitive walkthrough. This paper highlights frequently used inspection methods in the user interface design field. While understanding their context of use, usability issues of these methods are identified – some methods are too profound and some are too superficial, and most of them are coupled. Nielsen's usability heuristics, for example, is a coupled methodology. Besides being coupled, the heuristics are often too profound and poor in diagnosing context specific problems. On the other hand, functional and/or operational inspection methods, such as standards inspection and design rules, identify problems at a level that is too superficial. Hence, this paper suggests the employment of a formal user-centered design methodology to design inspection methods. The authors demonstrate how axiomatic design theory can be used to design inspection methods.

Keywords: axiomatic design, guidelines, heuristics, inspection methods, usability evaluation, and user interface design patterns.

1 INTRODUCTION

Over the past two decades, users have become increasingly intolerant of poor usability. Comparing two functional equivalent computer systems, the one with poorer user interface (UI) usability will become obsolete, and will be abandoned. Due to this trend, there has been an increasing emphasis on UI usability, which has resulted in the development of numerous usability evaluation methods. These methods were classified by Ivory and Hearst (2001) into five categories: testing, inspection, inquiry, analytical modeling, and simulation.

Among these five classes of methods, inspection is the only one that does not involve users. Analytical modeling and simulation may sometimes exclude users. However, to create accurate user models, evaluators must interact extensively with users. Hence, users are indirectly involved in these methods. It is good to have users' involvement during usability evaluation and feedback from them can be valuable. Nevertheless, in situations where users can not be involved, evaluators will have to rely on inspection methods to evaluate the UI.

A large number of inspection methods are available. On the surface, most of them appear quite similar to one another. Upon closer examination they can actually be segregated into several distinctive groups: heuristic evaluation, guideline review, standards inspection, style guides, design rules, cognitive walkthrough, and pluralistic walkthrough (Nielsen, 1994; Vanderdonckt, 1999; Ivory and Hearst, 2001).

Unfortunately, inspection methods may themselves suffer from usability problems. The objective of this study is, therefore, to demonstrate how a highly usable inspection method can be designed.

1.1 REVIEW OF INSPECTION METHODS

Before proceeding to design a highly usable inspection method, it is necessary to understand usability problems with existing inspection methods. For this purpose, the literature is first reviewed.

Several studies have compared usability evaluation methods (Desurvire *et al.* 1991, Jeffries *et al.* 1991, Karat *et al.* 1992 and Virzi *et al.* 1993). These studies suggest that heuristic evaluation and user testing are the most effective methods for identifying usability problems. Hence, they are frequently employed by evaluators. On the other hand, when concluding which of the two is the more effective method, the studies gave contradictory recommendations. This can be explained from the results of a subsequent empirical test conducted by Fu *et al.* (2002), which revealed that heuristic evaluation identifies significantly more low performance level problems (skill-based and rule-based problems), while user testing identifies significantly more high performance level problems (knowledge-based problems). Hence, the previous studies display contradictory results because they utilized tasks of different level of difficulty – studies that utilized skill-based and rule-based task revealed that heuristic evaluation is more effective than user testing, while studies that utilized knowledge-based task revealed otherwise.

Several usability issues concerning inspection methods have been reported. Some methods are, as we mentioned, coupled. For example, Nielsen's usability heuristics is a coupled method; see Figure 1 (Helander 2003). Ideally there should be a one-to-one correspondence between the functional requirements (FRs) and design parameters (DPs) in Figure 1. However, most of the FRs were satisfied by several DPs and vice versa. This makes a coupled process; we may find that one value of a DP will satisfy one FR but not another FR. By reengineering the FRs, we generated a less coupled matrix (Figure 2). The number of couplings was reduced from 22 *X*s in the matrix to 17 *X*s. In addition, only the inner square of the matrix was coupled, which reduced the search for solutions.

experiment which students were asked to use a two page interface standard to design an interface, and a field experiment which studied developers' usage of their company's interface standard. The students achieved only 71% compliance with the two page standard. The developers, while using their company's standards, were able to find only 4 of 12 deviations in a sample system. Three of their real products violated 7 to 12 of the 22 mandatory rules in the standard.

From this review, we understand that user testing and heuristic evaluation are the most usable UI evaluation methods. If evaluators are to choose between several evaluation methods, it is likely that they will select user testing or heuristic evaluation. This also implies that if evaluators decide to employ inspection methods and have to select one, it is likely that they will select heuristic evaluation. We also identified several usability issues for every inspection method – including heuristic evaluation.

$$\left\{\begin{array}{l} FR_1: \text{Keep users informed about system status} \\ FR_2: \text{Match between system and the real world} \\ FR_3: \text{User control and freedom} \\ FR_4: \text{Consistency} \\ FR_6: \text{Facilitate recognition rather than recall} \\ FR_7: \text{Flexibility and efficiency of use} \\ FR_8: \text{Help users recognize, diagnose, and recover from errors} \end{array}\right\} = \begin{bmatrix} X & X & & X & X & & X \\ X & X & & X & X & & \\ & & X & & X & & \\ & X & & X & & & \\ & & & & X & & X \\ & & X & & & X & \\ X & X & X & X & & & X \end{bmatrix} \left\{\begin{array}{l} DP_1: \text{Provide feedback} \\ DP_2: \text{Speak user's language} \\ DP_3: \text{Provide emergency exit} \\ DP_4: \text{Follow platform conventions} \\ DP_6: \text{Visibility of objects, actions, and options} \\ DP_7: \text{Facilitate accelerators} \\ DP_8: \text{Suggest solutions in error messages} \end{array}\right\}$$

Figure 1 – illustrates that Nielsen's usability heuristics is a coupled system.

$$\text{Constraint } C_1 = \text{Minimalist design}$$

$$\left\{\begin{array}{l} FR_1^{new}: \text{Understandability of vocabulary} \\ FR_8^{altered}: \text{Guidance for error recovery} \\ FR_4^{altered}: \text{Consistency of behavior} \\ FR_2^{new}: \text{System transparency} \\ FR_2^{altered}: \text{Match between system behavior and the real world} \\ FR_3: \text{User control and freedom} \\ FR_7: \text{Flexibility and efficiency of use} \end{array}\right\} = \begin{bmatrix} X & & & & & & \\ X & X & & & & & \\ & & X & & X & & \\ X & X & X & X & X & & \\ & & X & X & X & & \\ & & & X & & X & \\ & & & & & X & X \end{bmatrix} \left\{\begin{array}{l} DP_1^{new}: \text{Conventional or user familiar words} \\ DP_8: \text{Suggest solutions in error messages} \\ DP_4^{altered}: \text{Follow platform conventional behavior} \\ DP_2^{new}: \text{Visibility of objects, actions, options, and system status} \\ DP_2^{altered}: \text{User familiar concepts and techniques} \\ DP_3: \text{Provide emergency exits} \\ DP_7: \text{Facilitate accelerators} \end{array}\right\}$$

Figure 2 – Nielsen's usability heuristics was reengineered into a less coupled one.

Mahemoff and Johnston (1998), and Dix et al. (1998) reported several usability problems of guideline review; Vanderdonckt (1999) enumerated 27 problems after going through five development milestones of guidelines. Generally, the problems are: guidelines too profound or too superficial, difficult to select appropriate guidelines, difficult to apply guidelines to specific context, difficult to interpret the intended prescription of guidelines, conflict between guidelines, and validity of guidelines.

Souza and Bevan (1990) conducted an empirical test to evaluate a set of usability standards drafted by ISO (International Standards Organisation). Designers were asked to study the standards, and use them to redesign a menu interface. The designers made errors or had difficulties with 91% of the standards. Thovtrup and Nielsen (1991) conducted two empirical tests: a laboratory

2 METHOD

As stated in the literature review, UI evaluators encounter many usability issues when using inspection methods. To resolve these issues, formal user-centered design methodologies must be employed to redesign inspection methods. During the design process, inspection methods can be treated as products and UI evaluators as end users, with the goal of ensuring that user needs are satisfied.

Below we propose a procedure for designing a usable inspection method. It has two stages. In the first stage, we interviewed UI evaluators to understand the usage context of inspection methods. In the second stage, we employed axiomatic design theory to specify the method's functional requirements, design parameters, and design constraints. The subsequent sections describe the procedure in detail.

2.1 INTERVIEW OF UI EVALUATORS

From the review of literatures, we identified several usability issues of inspection methods, but not their context of use. Hence, the interview of UI evaluators was conducted to document the context of usage of inspection methods. Two UI evaluators were interviewed. One of them was a human factors consultant with much experience in evaluating clients' UIs. The other was a human factors engineer who would frequently evaluate the UIs that he had designed. The interviews were conducted face to face and through email. Examples of questions asked are:

- What are the common procedures employed by UI evaluators when they evaluate UI?
- Are inspection methods usually used and in what context are they used?
- Is heuristic evaluation useful and easy to use, and in what context is it used?
- Are automatic capture and analysis tools, base on inspection methods, frequently used?
- You have mentioned that you mix and match guidelines from different sources. May I know from which sources do you usually get your guidelines?
- After evaluating UIs using guidelines, do you use other evaluation methods to evaluate further?

2.2 EMPLOYMENT OF AXIOMATIC DESIGN THEORY

Axiomatic design theory provides a methodology to compute component and relational complexities in design. This systematizes complexity analysis and hence facilitates complexity reduction. The principal concepts of axiomatic design can be summarized as follow:

- Design domains are used to group together different types of attributes in design.
- In a domain, the attributes form a hierarchical structure.
- Decision making in design is perceived as a mapping process between attributes that belong to two adjacent domains.
- Design equations are used to represent this mapping process between domains.

According to Suh (1990), a design equation is written as:

$$\{FR\} = [A]\{DP\}$$

{FR} is a vector of functional requirements; {DP} is a vector of design parameters; and [A] is a design matrix. Functional requirements (FRs) represent design goals, or what a designer wants to achieve. Design parameters (DPs) represent design solutions, or how the designer plans to achieve the design goals. In other words, there is a means-ends relationship between FRs and DPs.

The design matrix of a design with three FRs and three DPs is of the following form:

$$[A] = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

Conventionally the values of a design matrix will either be 'X' or '0', where 'X' represents a mapping between the corresponding vector components while '0' signifies no mapping.

Based on AD theory, Lo and Helander (2004) proposed a framework that analyzes couplings between the goals that a user wants to achieve and the control actions that are designed into a user interface. This type of coupling increases the gulf of execution (Norman, 1988). Lo and Helander termed the framework Design Equations for Systems Analysis (DESA) and demonstrated how it can be used to analyze consumer products such as film cameras.

There are four design domains in DESA. Each domain contains a unique set of design attributes. A variable-illumination ceiling lamp can be used to illustrate the characteristics of these attributes. The goal domain contains user goals (UGs) that describe a user's desired state of a system, such as "desired amount of light in the room". The functional domain contains functional requirements (FRs), which characterize the function of a system, such as "provide a range of illumination". The physical domain contains design parameters (DPs), which are physical embodiments or variables that are selected by a designer to satisfy the FRs, such as "electrical resistance". The action domain contains user actions (UAs) that are designed into a user interface for controlling the DPs, such as "rotate lamp switch".

Design equations can also be used to represent the mapping between UGs and FRs, between FRs and DPs, and between DPs and UAs:

$$\{UG\} = [A]\{FR\}$$
$$\{FR\} = [B]\{DP\}$$
$$\{DP\} = [C]\{UA\}$$

Hence,

$$\{UG\} = [U]\{UA\}, \text{ where } [U] = [A][B][C]$$

The last equation reflects the notion that the directness of control tasks does not solely depend on user interface design; it also depends on the functional specification and the underlying workings of the system.

With supporting information from the literature review and the interview, we proceed to employ axiomatic design theory in our design process, which can be summarized into the following 8 steps:

1. State product's title and list its users

2. List user goals (UGs)
3. Define functional requirements (FRs), and suggest their corresponding design parameters (DPs)
4. Define non-functional requirements (NFRs), and decompose them into lower levels of their hierarchy
5. Suggest design constraints (DCs) corresponding to the lowest level NFRs
6. Decompose FRs and DPs into lower levels of their hierarchy
7. Specify user actions (UAs) corresponding to the functional UGs
8. Analyze coupling between UGs and FRs, between FRs and DPs, and between DPs and UAs

3 RESULTS

In the following sections, we report results corresponding to our design methodology.

3.1 INTERVIEW RESULTS

The interviews revealed that heuristic evaluation and user testing are the most frequently used evaluation methods. This finding coincides with previous studies. The interview also revealed a common procedure for usability evaluation, which is to employ inspection methods followed by user testing. The preference for this procedure is evident if we examine Fu's findings – inspection methods were more effective in identifying low performance level problems (skill-based and rule-based problems), while user testing was more effective in identifying high performance level problems (knowledge-based problems. Therefore, many evaluators use inspection methods to eliminate low performance level problems, so as to focus on high performance level problems during user testing.

Based on Fu's findings, user testing alone would be inadequate, as it cannot identify many low performance level problems; it should be used together with inspection methods to identify the maximum number of problems. It may, however, be erroneous to judge an evaluation method's effectiveness based on the number of problems identified. The validity of the identified problems should hold more weight than the quantity. Based on our interview results, the latest trend in the UI design field is: if users do not report predicted problems, then they are not usability problems – many low performance level problems identified by inspection methods are, thus, invalid, and user testing alone is the final measure. As a result, some evaluators with time and/or budget constraints proceed straightaway to user testing.

One might question whether inspection methods can also be conducted alone. This procedure is usually employed when circumstances do not permit users' involvement. Inspection methods are often used together with other methods. This is because inspection methods can not identify many high performance level problems that are mostly valid, and can not assess usefulness of UIs. Then again, one might question further whether it is possible to develop an inspection method that can identify high performance level problems, and assess both ease of use and usefulness of UI. To answer this question, further research and investigation are needed.

The interview also revealed that although evaluators have numerous inspection methods to choose from, only heuristic evaluation and guideline review are more frequently used. This implies that heuristic evaluation and guideline review are more useful compared to other inspection methods, as Nielsen (1995) showed that inspection methods' usage frequency has strong positive correlation with its perceived usefulness. Nielsen's findings also showed that heuristic evaluation and user testing are the most frequently used evaluation methods, and thus most useful, which again coincides with the interview results and previous studies. Nielsen's study states 6 requirements for a frequently used inspection method:

1. Provides information that is useful in improving UI
2. Cheap
3. Fast to use
4. Easy to learn
5. Flexible and adapt to specific context
6. Aggressive advocacy

Most of these requirements correspond with inspection methods' usability issues identified in the literature review.

Sometimes, UI evaluators employ several inspection methods together. Some evaluators select a complete set of heuristics or guidelines, while others prefer to compare several sets, and select relevant prescriptions from each set.

From the interview results in the preceding paragraphs, we know that inspection methods, particularly heuristic evaluation and guideline review, are frequently employed for usability evaluation of UI. We also know that unless users' involvement is impractical, inspection methods are usually used together with user-participatory methods, and user testing is the most frequently employed method in such cases.

3.2 AXIOMATIC DESIGN RESULTS

In the last paragraph of method section, we specify 8 steps in our axiomatic design process. The following are results corresponding to each step:

1. Product: Inspection method

   Users:

   - Software design consultants
   - Software designers/engineers
   - Software developers/programmers

2. UGs were thought up by simulating mental processes of the users:

- Evaluate usability of UIs
- Easy to use method
- Effective method
- Inexpensive method

3. Only the first UG is functional. Hence, there is only 1 FR:

    - Usability evaluation

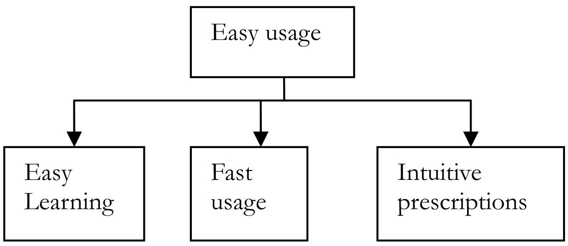
    Corresponding DP suggested:

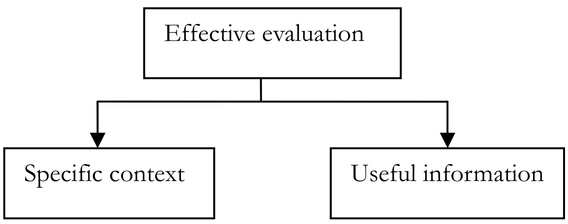
    - UI design patterns – a pattern describes the best common solution to a common problem within a specific context

4. The rest of the UGs are non-functional. Hence, there are 3 NFRs:

    - Easy usage
    - Effective evaluation
    - Low cost

    The first 2 NFRs can be decomposed into lower levels:





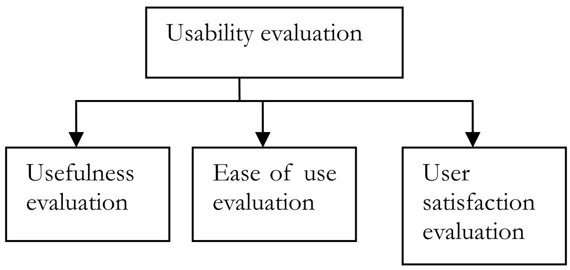
5. The decomposition in step 4 yield 6 lowest level NFRs:

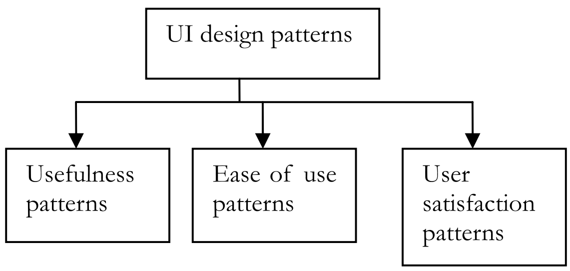
    - Easy learning
    - Fast usage
    - Intuitive prescriptions
    - Specific context
    - Useful information
    - Low cost

    Corresponding DCs suggested:

    - Step-by-step patterns' prescriptions for users to follow
    - Patterns follow standard presentation format. Automate searching of contextually relevant patterns from their database
    - Concise explanation of each pattern, which includes problems encountered, usage contexts, prescribed solution, solution rationale, and a pictorial example
    - Write specific patterns for specific usage context
    - Write patterns related to high performance level problems with the aid of user testing results
    - Online database of patterns for any users to employ, comment, update, and also contribute new ones. Link each type of UI to all its related patterns so that one evaluator is enough to identify most problems

6. Decomposition of the FR and corresponding DP into lower levels of their hierarchy:





7. UA corresponding to the functional UG:

    - Users search for contextually relevant UI design patterns from a web repository, and use them for usability evaluation

8. Since there is only 1 UA, coupling is absent.

4 DISCUSSION

Base on the axiomatic design results, there are nine requirements, both FRs and NFRs, for a highly usable inspection method:

1. Usefulness evaluation
2. Ease of use evaluation

3. User satisfaction evaluation
4. Easy learning
5. Fast usage
6. Intuitive prescriptions
7. Specific context
8. Useful information
9. Low cost

Designers of inspection methods can use this set of requirements in the design of new inspection methods. They can also use the requirements to systematically evaluate existing inspection methods, and use the results to redesign the methods. When existing methods are redesigned, they might have to be renamed because of changes in their inherent characteristics. Although there are many formal user-centered design methodologies for these designers to select from, we recommend the use of axiomatic design theory, as it can effectively analyze coupling. We are unable to demonstrate coupling analysis in the above design, as coupling is absent.

Besides inspection methods, axiomatic design theory can be used to design other types of usability evaluation methods. The procedure will be similar to the one employed in this study, and the evaluation methods' requirements may coincide with many of the above nine requirements.

UI evaluators can use the nine requirements to compare existing evaluation methods, and rationally decide the most appropriate method for the evaluation context. This can be done by inputting contextually relevant requirements and each evaluation methods' DPs into a QFD's (Quality Function Deployment) "house of quality". After rating the importance of each requirement, the evaluators can then make rational comparisons between the different methods. It also clearly displays trade-offs between the DPs of different methods.

The suggested DP, UI design patterns, is a satisficing solution, and thus, may not be the best solution. Patterns are increasingly used to design and evaluate UI. Yahoo! has designed and built a pattern library that contains web UI design patterns: http://developer.yahoo.com/ypatterns. Nevertheless, there may be other solutions that better fulfill the nine requirements. Whatever the DP is, it has to be further decomposed into levels lower than what is shown in the results. The same applies for the FR.

5 CONCLUSIONS

This paper reviews inspection methods' usability issues reported in previous studies, and advocates the use of formal user-centered design methodologies to design inspection methods. It demonstrates how this can be done using axiomatic design theory. The axiomatic design process is supported with information from the literature review and interview of UI designers. Future inspection methods designers should also employ formal methodologies to design inspection methods in order to ensure good usability. They can consider following the procedure proposed in this paper.

6 REFERENCES

Desurvire, H., Lawrence, D., & Atwood, M. (1991). Empiricism versus judgment: Comparing user interface evaluation methods on a new telephone-based interface. *SIGCHI Bulletin, 23*, 58-59.

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

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

Helander, M. G. (2003). Couplings among HCI design parameters cause poor usability. In *Proceedings of Human Factors and Ergonomics Society 47th Annual Meeting*. Santa Monica, CA: Human Factors and Ergonomics Society.

Ivory, M. Y., & Hearst, M. A. (2001). The state of the art in automating usability evaluation of user interfaces. *ACM Computing Surveys, 33*, 470-516.

Jeffries, R., Miller, J. R., Wharton, C., & Uyeda, K. M. (1991). User interface evaluation in the real world: A comparison of four techniques. In *Proceedings of ACM CHI'91 Conference on Human Factors in Computing Systems* (pp. 119-124). New York, NY: ACM Press.

Karat, C. M., Campbell, R., & Fiegel, T. (1992). Comparison of empirical testing and walkthrough methods in user interface evaluation. In *Proceedings of the ACM CHI'92 Conference on Human Factors in Computing Systems* (pp. 397-404). New York, NY: ACM Press.

Lo, S., & Helander, M. G. (2004). Method for analyzing the usability of consumer products. In *Proceedings of the third international conference on axiomatic design – ICAD2004*. Seoul, Korea: Institute for Axiomatic Design.

Mahemoff, M. J., & Johnston, L. J. (1998). Principles for a usability-oriented pattern language. In *Proceedings of Australian Computer Human Interaction Conference OZCHI'98* (pp. 132-139). Los Alamitos, CA: IEEE Computer Society.

Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. In *Proceedings of ACM CHI'94 Conference on Human Factors in Computing System* (pp. 152-158). New York, NY: ACM Press.

Nielsen, J. (1995). Technology transfer of heuristic evaluation and usability inspection. *Presented as a keynote at the IFIP INTERACT'95 International Conference on Human-Computer Interaction*, Norway.

Norman, D. A. (1988). *The psychology of everyday things*. New York, NY: BasicBooks.

Souza, F., & Bevan, N. (1990). The use of guidelines in menu interface design: Evaluation of a draft standard. In *Proceedings of IFIP INTERACT'90: Human-Computer Interaction* (pp. 435-440). Cambridge, UK: North-Holland.

Suh, N. P. (1990). *The principles of design*. New York, NY: Oxford University Press.

Vanderdonckt, J. (1999). Development milestones towards a tool for working with guidelines. *Interacting with computers, 12*, 81-118.

Virzi, R. A., Sorce, J.F., & Herbert, L. B. (1993). A comparison of three usability evaluation methods: Heuristic, think-aloud, and performance testing. In *Proceedings of the Human Factors and Ergonomics Society 37th Annual Meeting* (pp. 309-313). Santa Monica, CA: Human Factors and Ergonomics Society.