Proceedings of ICAD2004
The Third International Conference on Axiomatic Design
Seoul – June 21-24, 2004
ICAD-2004-26

DEVELOPING A FORMAL USABILITY ANALYSIS METHOD FOR CONSUMER PRODUCTS

Shuan Lo
shuanlo@pmail.ntu.edu.sg
School of Mechanical and Production Engineering,
Nanyang Technological University, Singapore

Martin G. Helander
mahel@ntu.edu.sg
School of Mechanical and Production Engineering,
Nanyang Technological University, Singapore

ABSTRACT

This paper proposes a formal method for usability analysis based on the axiomatic design theory. It characterizes the degree of coupling between user goals and user actions that are defined by the design of a product. Couplings between user goals and action reduce usability. As an analytical tool, this method can save time and resources by enabling an early analysis of product usability.

Keywords: complexity, consumer product, ergonomics, formal method, human factors, usability

1 INTRODUCTION

1.1 BACKGROUND

As technology advances, the products people use in their homes and at work are becoming increasingly complex in terms of features and functionality [Weir, 1991]. In fact, some companies even use the number of features in their products as a sales argument. However, researchers have reported that providing more features results in a more complex user interface [Han, Yun, Kim, Kwahk, and Hong 2001]. Users are also increasingly unwilling to tolerate difficult-to-use products. Many customers now consider usability, along with functionality, price, and after sales service quality, as one of the most important factors in making purchasing decisions [Dumas and Redish, 1994]. Usability is also considered as an important criteria by Consumer Reports, which is an influential US guide to purchase of new products. To remain competitive, industry now looks at usability.

Usability may be perceived as a property of the interaction between a product, a user, and a set of tasks that he is trying to complete [Jordan, 1998]. Formally, it is defined as the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use [International Organization for Standardization, 1998]. Effectiveness refers to the accuracy and the completeness with which users achieve the specified goals. Efficiency refers to the resources expended in relation to the accuracy and the completeness with which users achieve the specified goals.

Information about usability can be obtained through a usability evaluation. Jacko and Sears [2003] classify usability evaluation methods using the following categories:

- user-based evaluation,
- inspection-based evaluation, and
- model-based evaluation.

User-based evaluation refers to methods that require the involvement of end-users of the product. Two examples are questionnaires [e.g. Chin, Diehl, and Norman, 1988] and empirical usability testing [Rubin, 1994]. Inspection-based evaluation refers to methods, such as heuristic evaluations [Nielsen and Molich, 1990] and cognitive walkthroughs [Lewis and Wharton, 1997], which rely on the judgment of experts rather than input from potential users. Model-based evaluation refers to methods that predict usability measures by calculation or simulation of how a person would use the proposed system. An example is GOMS [Card, Moran, and Newell, 1983].

What makes a method formal? In computing, the phrase “formal methods” has a narrow meaning. Saiedian [1996] defines a formal requirement specification tool as one that has precise mathematical semantics, and a formal method is one that uses a formal tool or notation. However, In this paper we take a broader view of formalism; we define formal as orderly and methodical. As noted by Luqi and Goguen [1997], the degree formalization of a method can range from very formal to very informal, and formalization is useful only to the extent that it helps to meet concrete goals.

The concepts of usability and complexity are intimately linked. Generally, a system is described as complex when one has problems understanding or dealing with it. Previous research largely agrees there are at least two main sources of complexity: a large number of parts and a large number relations [Steward, 1981; Weng, Bhalla, and Lybengar, 1999; Woods; 1988; Yates, 1978]. Miller [2000] named these types of complexity as “component complexity” and “relational complexity”, respectively.

Both component and relational complexities create user problems due to the limits of cognitive capacity. People can attend to and mentally manipulate only four independent pieces of information at the same time [Rode, 2000]. As the number of elements and relations in a system increases, people have great difficulty in predicting the effects of an action, or tracing the

Copyright © 2004 by ICAD2004

origin of a disturbance in a system [Döner, 1996]. An experiment shows that the time a user needs to complete an uncoupled parameter design task increases linearly with the problem size; however, the time taken to complete coupled parameter design tasks increases geometrically with the problem size [Hirschi and Frey, 2002].

An observer can reduce the perceived complexity of a large system by focusing his attention to a few critical parts of the system. However, when relational complexity is high, component complexity also tends to be high because of the need to pay attention to interrelated elements, thereby expanding the span of attention beyond one's capabilities. When a system is highly coupled, as the size of the system increases linearly, the number of relations increases geometrically, see Figure 1. Hence, coupling is a major source of system complexity.

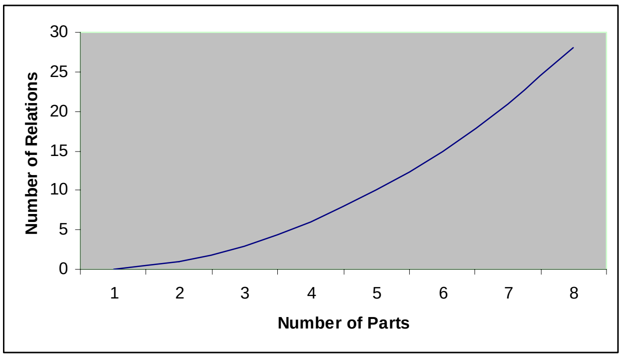

Figure 1. The number of possible relations increases geometrically with the number of parts in a system.

1.2 COMPLEXITY IN HUMAN-MACHINE INTERACTION

Coupling and complexity can exist at different abstraction levels of an interactive system. Weir [1991] separated a system into the following three levels:

- the domain level,
- the control requirements level, and
- the required user interaction level.

Domain refers to a set of system goals and environmental conditions; control refers to the hardware and software control mechanisms of the system; interaction refers to the operators' physical and cognitive responses to the system. The complexities in the different levels interact with each other, and the overall complexity is revealed to the operator thru the human-machine interface.

Weir's [1991] concept of system complexity provides a starting point for studying the subject. By separating a complex system into different levels, an analyst can be more specific in describing the locus of complexity. In addition, identifying the level in which complexity occurs helps to suggest possible redesigns. However, more research is required to develop Weir's abstract concept into an applicable and formalized method. Such a method should facilitate engineers and designers to study the relationships between the various levels and to predict the resulting complexity that will be perceived by an operator.

2 CONCEPTUAL FRAMEWORK

2.1 TWO DOMAINS USABILITY ANALYSIS FRAMEWORK

In this paper, we propose a method for analyzing the relationships between the goals that a user wants to achieve and the actions that are allowed by a design. This method, which is called Formal Top-down Analysis (FTDA), characterizes the complexity of a human-machine interaction by identifying couplings between the user goals and the user actions.

FTDA is inspired by Suh's [1990] axiomatic design theory, which has been successfully applied in various fields such as machine design, product design, and software design [e.g. Schreyer and Tseng, 2000; Suh, 2001]. In recent years, there is also literature in human factors engineering and human-computer interaction (HCI) that refer to axiomatic design theory [Helander and Jiao, 2002; Helander and Lin, 2000; Helander and Lin, 2002; Karwowski, 2003; Quill, Kancler, Revels, and Batchelor, 2001].

A fundamental concept in axiomatic design is the use of design domains. Between each pair of adjacent domains, there is a means-ends relationship; one domain represents the ends, while another represents the means. Two design domains can be used to represent a usability design problem: a user goal domain and a user action domain. A designer's aim should be to select a set of user actions that enables the user to achieve his goals in a direct and straightforward manner. User goals (UGs) refer to the user desired states of a system [Norman, 1988] – what the user wants to achieve. User actions (UAs) refer to the user's physical interaction with the product or system in order to achieve the UGs [Norman, 1988]. The mapping between user goals (UGs) and user actions (UAs) can be represented by a directed graph, see Figure 2.

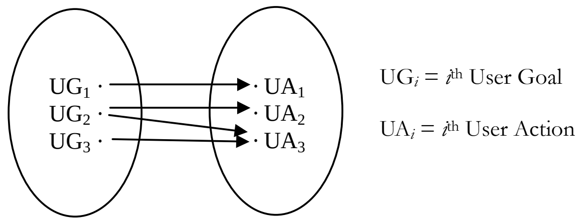

Figure 2. Representing a human-machine interaction design problem using two design domains.

In order to minimize the gulfs of execution and to simplify the user's mental model, couplings between UGs and UAs should be minimized (Norman, 1988). While a one-to-one mapping between UGs and UAs is desirable, a one-to-many mapping is sometimes acceptable [Suh, 1990]. This is because the perceived complexity of a one-to-many system can be reduced if a proper operating sequence is followed. Such a system is referred to as a semi-coupled or decoupled design, and it is discussed in further details below.

Another technique of representing the mapping between UGs and UAs by using a design equation [cf. Suh, 1990]:

$$\{UG\} = [U]\{UA\} \tag{1}$$

Copyright © 2004 by ICAD2004

where {UG} is a vector of user goals, {UA} is a vector of user actions, and [U] is known as a design matrix.

The design matrix of a design that has three UGs and three UAs is of the following form:

$$[U] = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{bmatrix} \qquad (2)$$

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. Each 'X' in a design matrix is referred to as a dependency.

In FTDA, design equations are used instead of directed graphs. This is because, as the number of UGs, UAs, and their interrelationships increases, a design matrix becomes easier to work with than a graph, and it can used for computational analysis. Furthermore, the shape of a design matrix can be used to characterize the degree of coupling, and hence complexity, in a design.

Based on the shape of a design matrix, three levels of coupling can be identified [Suh, 1990]:

- coupled,
- uncoupled, and
- semi-coupled or decoupled.

A coupled design is one in which there are dependencies on both sides of the diagonal of [U]; there are many-to-many mappings between user goals and user actions. Such a design is complex to analyze and to use. An uncoupled design is one in which all the non-diagonal elements of [U] are zeros; there is a one-to-one mapping between UGs and UAs. Such a design enables the user to construct a straightforward mental model.

The design of a water faucet can be used to exemplify a coupled design and an uncoupled design [Sohlenius, personal communication, 1998]. Consider a water faucet that has two knobs; one for controlling hot water, another for controlling cold water. The UGs for such a water faucet can be stated as follow:
UG_1 = Desired water flow rate
UG_2 = Desired water temperature
The corresponding UAs can be stated as follow:
UA_1 = Turn hot water knob
UA_2 = Turn cold water knob
The design equation is

$$\begin{Bmatrix} UG_1 \\ UG_2 \end{Bmatrix} = \begin{bmatrix} X & X \\ X & X \end{bmatrix} \begin{Bmatrix} UA_1 \\ UA_2 \end{Bmatrix} \qquad (3)$$

Equation 3 shows that the design matrix is full; hence it is a coupled design. The user needs to perform several adjustments of both controls to achieve the desire combination of temperature and flow rate.

Consider another water faucet design that has a single handle for controlling the flow rate and the water temperature. While the UGs for this design is same as above, the UAs for this design are as follow:
UA_1 = Pivot handle up/down
UA_2 = Pivot handle left/right
The design equation is

$$\begin{Bmatrix} UG_1 \\ UG_2 \end{Bmatrix} = \begin{bmatrix} X & \\ & X \end{bmatrix} \begin{Bmatrix} UA_1 \\ UA_2 \end{Bmatrix} \qquad (4)$$

Equation 4 shows that the design matrix is diagonal; hence this is an uncoupled design. The user is able to achieve his goals in a straightforward manner.

A semi-coupled design is one in which [U] is a lower or upper triangular matrix: there is a one-to-many mapping between user goals and user actions. Such a design is acceptable, provided that the proper operating sequence is known to the user. An example of a decoupled design is an adjustable microscope workstation [Helander and Lin, 2002]. The basic UGs for the workstation are as follow:
UG_1 = Support for feet
UG_2 = Table at elbow height
UG_3 = Eyes at microscope height
The corresponding UAs are as follow:
UA_1 = Adjust chair height
UA_2 = Adjust table height
UA_3 = Adjust microscope height
The design equation is

$$\begin{Bmatrix} UG_1 \\ UG_2 \\ UG_3 \end{Bmatrix} = \begin{bmatrix} X & & \\ X & X & \\ X & X & X \end{bmatrix} \begin{Bmatrix} UA_1 \\ UA_2 \\ UA_3 \end{Bmatrix} \qquad (5)$$

The triangular design matrix in Equation 5 indicates that this is a decoupled design. Hence if the correct sequence for determining the UAs is not followed, the design behaves like a coupled design and repeated adjustments are necessary.

2.2 FOUR DOMAINS USABILITY ANALYSIS FRAMEWORK

The framework presented above can be used to analyze the degree of coupling in a human-machine interaction. However, in order to identify the root of usability problems, often the structure of the product has to be included in an analysis.

In axiomatic design theory, the structure of a product is represented by a functional domain, a physical domain, and their interrelationships. The functional domain contains a minimum set of functional requirements (FRs) that completely characterizes the functional needs of a product. The physical domain contains a set of key physical variables or design parameters (DPs) that characterize the design that satisfies the specified FRs.

Functional and physical attributes may belong to different aspects of design; for the purpose of our analysis, we consider only the FRs that are related to the UGs, which are referred to as FR^us, and the DPs that are related to the FR^us, which are referred to as DP^us. An exemplary set of UG, FR^u, DP^u, and UA for a variable-illumination ceiling lamp is as follows:
UG: Desired amount of light
FR^u: Illumination adjustability
DP^u: Electrical impedance
UA: Rotate light switch

A usability analysis framework that consists of four design domains is illustrated in Figure 3. The goal domain and the action domain characterize the tasks that a user has to perform, while the functional and physical domains characterize the

Copyright © 2004 by ICAD2004

structure of the product. The mapping across the domains represents the design decision-making process.

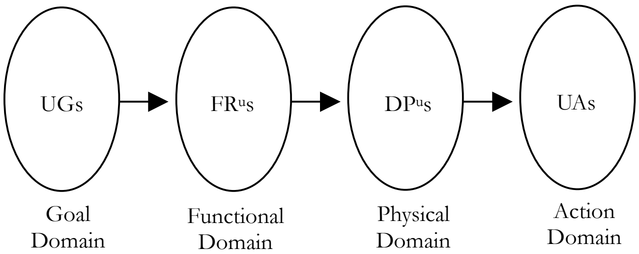

Figure 3. Representing a human-machine interaction design problem using four design domains.

Design equations are also used to represent the relationships between UGs and FR^us, between FR^us and DP^us, and between DP^us and UAs:

$$\{UG\} = [A]\{FR^U\} \tag{6}$$

$$\{FR^U\} = [B]\{DP^U\} \tag{7}$$

$$\{DP^U\} = [C]\{UA\} \tag{8}$$

Hence,

$$\{UG\} = [A][B][C]\{UA\} \tag{9}$$

and

$$[U] = [A][B][C] \tag{10}$$

Therefore, the overall complexity that a user experiences is determined by the decisions that are made in different facets of design. The relationships between the UGs and the UAs determine the directness of the interaction, but the UAs are determined by the interpretation of the customer needs, the specification of the design problem, the selection of the design solution, and the conceptualization of the user interface.

3 CASE STUDIES

This section describes three case studies of usability analysis by FTDA, which involves existing designs of a refrigerator design, a manual film camera, and a digital camera.

3.1 NORMAN'S REFRIGERATOR

Norman [1988] criticized the usability of a refrigerator design with a freezer compartment and a fresh food compartment. We encountered a similar design. There are only two controls on the user interface; one controls the temperature of the air by adjusting a thermostat in the refrigerator, the other adjusts a valve that determines the percentage of cold air entering either of the two compartments, see Figure 4.

One can understand the difficulty by considering the following scenario: suppose the settings of the controls are at 'A' and '3', and the user wishes to lower the temperature of the fresh food department without changing the temperature of the freezer department; how should he do it?

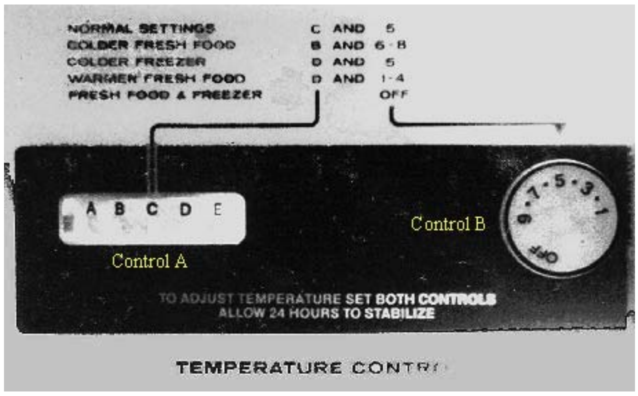

Figure 4. User interface of a refrigerator for controlling temperature.

FTDA can be used to identify the underlying cause of the usability problem. The UGs are formulated as follow:
UG_1 = Desired temperature of the freezer compartment
UG_2 = Desired temperature of the fresh food compartment
These UGs are mapped to the following FR^us:
FR^U_1 = User control for freezer compartment temperature
FR^U_2 = User control for fresh food compartment temperature
The DP^us that were selected by the designer of the refrigerator are identified as
DP^U_1 = Air temperature thermostat
DP^U_2 = Angle of the air volume valve
Based on the user interface design, the corresponding UAs are
UA_1 = Set control A
UA_2 = Set control B
The design equations are as follow:

$$\begin{Bmatrix} UG_1 \\ UG_2 \end{Bmatrix} = \begin{bmatrix} X & \\ & X \end{bmatrix} \begin{Bmatrix} FR^U{}_1 \\ FR^U{}_2 \end{Bmatrix} \tag{11}$$

$$\begin{Bmatrix} FR^U{}_1 \\ FR^U{}_2 \end{Bmatrix} = \begin{bmatrix} X & X \\ X & X \end{bmatrix} \begin{Bmatrix} DP^U{}_1 \\ DP^U{}_2 \end{Bmatrix} \tag{12}$$

$$\begin{Bmatrix} DP^U{}_1 \\ DP^U{}_2 \end{Bmatrix} = \begin{bmatrix} X & \\ & X \end{bmatrix} \begin{Bmatrix} UA_1 \\ UA_2 \end{Bmatrix} \tag{13}$$

Therefore,

$$\begin{Bmatrix} UG_1 \\ UG_2 \end{Bmatrix} = \begin{bmatrix} X & X \\ X & X \end{bmatrix} \begin{Bmatrix} UA_1 \\ UA_2 \end{Bmatrix} \tag{14}$$

Equation 14 shows that this refrigerator has a coupled interaction design. Equation 12 shows that the locus of the coupling lies between the functional and the physical domains. According to Norman [1988], presenting a wrong conceptual model to the user and the lack of feedback are the causes of the usability problem of this refrigerator. However, the results of

Copyright © 2004 by ICAD2004

FTDA suggest more specifically that the root of the usability problem is the coupling between the UGs and the UAs.

3.2 MANUAL FILM CAMERA

The second product that is used to demonstrate the use of FTDA for usability analysis is a manual film camera. The model that was studied is a Nikon FM2, but the design of this camera is typical of manual single-lens-reflex (SLR) film cameras. In taking photograph, these are the key UGs:

UG_1 = Desired image sharpness
UG_2 = Desired level of depth-of-field
UG_3 = Desired amount of motion blur
UG_4 = Desired image brightness
UG_5 = Desired image captured

Based on our analysis of the camera functions, the following UAs are involved:

UA_1 = Rotate lens focusing ring
UA_2 = Rotate aperture ring
UA_3 = Rotate shutter speed ring
UA_4 = Press shutter release

Subsequently, the mapping between the UGs and the UAs is

$$\begin{Bmatrix} UG_1 \\ UG_2 \\ UG_3 \\ UG_4 \\ UG_5 \end{Bmatrix} = \begin{bmatrix} X & & & \\ X & X & & \\ & & X & \\ & X & X & \\ X & X & X & X \end{bmatrix} \begin{Bmatrix} UA_1 \\ UA_2 \\ UA_3 \\ UA_4 \end{Bmatrix} \quad (15)$$

Since the number of UAs is less than the number of UGs, it is a coupled design [Suh, 1990]. Equation 15 shows couplings lie between UG_2, UG_3, UG_4, UA_2, and UA_3. In order to understand the underlying cause of this coupling, there is a need to study the structure of the product, and hence, the mapping between the other design domains.

The UGs are translated into the following FR^us:

$FR^U{}_1$ = Provide user control for focusing picture
$FR^U{}_2$ = Provide user control for adjusting depth-of-field
$FR^U{}_3$ = Provide user control for adjusting degree of motion blur
$FR^U{}_4$ = Provide user control for adjusting degree of exposure
$FR^U{}_5$ = Provide user control for exposing frame

The following DP^us are identified:

$DP^U{}_1$ = Distance between lens and film
$DP^U{}_2$ = Aperture size
$DP^U{}_3$ = Shutter speed
$DP^U{}_4$ = Shutter release mechanism

The design equations are formulated as follow:

$$\begin{Bmatrix} UG_1 \\ UG_2 \\ UG_3 \\ UG_4 \\ UG_5 \end{Bmatrix} = \begin{bmatrix} X & & & & \\ & X & & & \\ & & X & & \\ & & & X & \\ X & X & X & X & X \end{bmatrix} \begin{Bmatrix} FR^U{}_1 \\ FR^U{}_2 \\ FR^U{}_3 \\ FR^U{}_4 \\ FR^U{}_5 \end{Bmatrix} \quad (16)$$

$$\begin{Bmatrix} FR^U{}_1 \\ FR^U{}_2 \\ FR^U{}_3 \\ FR^U{}_4 \\ FR^U{}_5 \end{Bmatrix} = \begin{bmatrix} X & & & \\ X & X & & \\ & & X & \\ & X & X & \\ & & & X \end{bmatrix} \begin{Bmatrix} DP^U{}_1 \\ DP^U{}_2 \\ DP^U{}_3 \\ DP^U{}_4 \end{Bmatrix} \quad (17)$$

$$\begin{Bmatrix} DP^U{}_1 \\ DP^U{}_2 \\ DP^U{}_3 \\ DP^U{}_4 \end{Bmatrix} = \begin{bmatrix} X & & & \\ & X & & \\ & & X & \\ & & & X \end{bmatrix} \begin{Bmatrix} UA_1 \\ UA_2 \\ UA_3 \\ UA_4 \end{Bmatrix} \quad (18)$$

Equation 16 shows that the mapping between the UGs and the FR^us is semi-coupled. This implies that there is a required sequence for satisfying the UGs; the satisfaction of UG_1, UG_2, and UG_3, should precede the satisfaction of UG_4. This sequential requirement is reflected in the resulting mapping between the UGs and the UAs, see Equation 15.

Equation 17 shows that the locus of coupling lies between the functional and the physical domains. Two design parameters: $DP^U{}_2$ and $DP^U{}_3$, are used to satisfy three functional requirements: $FR^U{}_2$, $FR^U{}_3$, and $FR^U{}_4$. As a result of the coupling, the product may not be able to satisfy all the FR^us concurrently. We note that this design is common for most manual SLR film cameras.

Since the number of FR^us is one more than the number of DP^us, adding an appropriate DP^u, and hence a corresponding UA, may help to decouple the design [Suh, 1990]. One example of a suitable DP^u is a variable power electronic flash. Another way of solving the problem would be to use a film with variable ISO film speed, see Equation 19. In fact, there is a design feature for some digital cameras, which allows a user to manipulate the sensitivity of the receptor surface.

$$\begin{Bmatrix} FR^U{}_1 \\ FR^U{}_2 \\ FR^U{}_3 \\ FR^U{}_4 \\ FR^U{}_5 \end{Bmatrix} = \begin{bmatrix} X & & & & \\ X & X & & & \\ & & X & & \\ & X & X & X & \\ & & & & X \end{bmatrix} \begin{Bmatrix} DP^U{}_1 \\ DP^U{}_2 \\ DP^U{}_3 \\ DP^U{}_* \\ DP^U{}_4 \end{Bmatrix} \quad (19)$$

where $DP^U{}_*$ = Electronic flash power or ISO setting.

3.3 DIGITAL CAMERA

The third product that was used to demonstrate the use of FTDA is a digital camera Fujifilm MX-2700. The attributes in the goal and functional domains remain the same as above, hence only the mapping between the physical and the action domains is presented below.

In the manual mode, there are two user interface features that control the electronic flash: flash mode control and flash brightness control. There are also two features that control the digital shutter speed and the aperture size: "slow synchro" mode and exposure compensation control. We think that the DP^us that are related to these features are as follow:

$DP^U{}_1$ = Flash power
$DP^U{}_2$ = Digital shutter speed
$DP^U{}_3$ = Aperture size

The corresponding UAs are as follow:

UA_1 = Set flash mode
UA_2 = Set flash brightness
UA_3 = Switch slow synchro mode on/off
UA_4 = Set exposure compensation

Copyright © 2004 by ICAD2004

If our understanding of the working mechanisms in the camera is correct, the mapping between the DP^us and UAs, as shown by Equation 20, is coupled. This example shows that, due to the design of the user interface, coupling can also lie between the physical and the action domains

$$\begin{Bmatrix} DP^U{}_1 \\ DP^U{}_2 \\ DP^U{}_3 \end{Bmatrix} = \begin{bmatrix} X & X & & \\ & & X & X \\ & & X & X \end{bmatrix} \begin{Bmatrix} UA_1 \\ UA_2 \\ UA_3 \\ UA_4 \end{Bmatrix} \quad (20)$$

4 DISCUSSION

4.1 FORMAL APPROACH TO USABILITY EVALUATION

This paper presents a model for consumer product interaction design based on axiomatic design theory. It integrates decision-making in engineering design with decision-making in interaction design. At this stage, the model may not be complete or final, but it helps to conceptualize the relationships between various facets of consumer product design.

Based on the model, a formal and analytical method for usability analysis may be carried out. It is analytical in the sense that test subjects and user testing are not necessary. This method adopts an "anti-reductionist" or systems view [Sanderson, 2003], and it characterizes the degree of coupling between potential user goals and user actions that are designed into a product. This type of coupling reduces the usability of a product because it increases the level of complexity that a user perceives.

Test subjects are not likely to discover that the source of a problematic design is that it is coupled; the problem is difficult to analyze unless matrices are used. Even Norman [1988] did not consider coupling as the main source of usability problem for the refrigerator design.

FTDA measures the complexity of the interaction between a product and its targeted users. This type of complexity, unlike mass and volume, is not an independent property of a product. In order to begin the analysis, the analyst needs to understand the user goals. This is important because a design that is uncoupled for one set of user goals may be coupled for another. For example, the manual SLR camera described above will appear to be less complex if the user is willing to sacrifice some of the goals that were assumed in the analysis, such as the depth of view of a picture. The ability to specify a suitable set of user goals for design and analysis depends on the expertise of the designer and his understanding of the targeted users.

FTDA is not formal in the sense of Z [Spivey, 1989] or other formal specification methods in computing [see Wing, 1990]. However, it has some elements of formality, such as a well-prescribed process. In addition, the design matrix allows a designer or analyst to perform some analysis at the surface level of the representation, which is the hallmark of formalism in mathematics [Dix, personal communication, 2003].

As mentioned above, the use of matrices to represent mapping in design provides certain advantages. First, matrix-clustering algorithms can be used to identify any locus of coupling [e.g. King, 1979; Steward, 1981; Suh, 1990]. Consequently, the design matrix can suggest what couplings should be removed, and thereby indirectly suggest what could be done to improve the design. However, detailed design solutions will still have to be stated by the designer, and design synthesis remains difficult. Second, the symmetrical or asymmetrical properties of a matrix give a powerful summation and overview of the design characteristics and potential couplings. In addition, as compared to diagrammatical notations and common language textual descriptions, a matrix potentially requires less space for documentation.

To characterize a usability problem, four domains were presented above; the goal domain, the functional domain, the physical domain, and the action domain. There is, however, some flexibility regarding how many domains should be used for the analysis. The water faucets example demonstrates that one can evaluate usability by using the goal and the action domains. This is similar to the approach used in some task analysis methods. [Stanton and Diaper, 2003]. Doing so can help to simplify the analysis and enable a "rapid usability evaluation"; an analyst is able to compare the complexity in design and thereby the usability of the water faucets even without understanding the internal working mechanisms of the faucets, or even before the mechanisms have been worked out in detail by the engineering designers. However, as shown in the manual camera case study, including the product structure in the analysis increases the ability to suggest design improvements. This is because the underlying cause of coupling in a user interaction design may involve engineering design decisions.

4.2 USABILITY METHOD FOR NON-EXPERTS

A re-occurring theme in applied ergonomics is the idea of "giving the methods away" to those with little formal education in the subject [Stanton and Young, 2003]. In our opinion, FTDA is rather straightforward. The criteria for evaluation are easy to understand; the concept of decoupling a problem is not new to engineering, and it does not require deep understanding of human factors engineering and cognitive psychology. [Sadun, 2001] Therefore, designers and engineers may be more likely to accept, learn, and practice this method.

An early analysis of complexity would be a significant advantage and would save time and resources [Hamacher, Marrenbach, and Kraiss, 2001]. Stanton and Young [2003] noted that companies have difficulties in making decisions on human performance and usability early in the design process. The method presented in this paper provides a rational and general criterion for predicting usability and can help to improve the quality of design iteration by aiding designers in identifying design flaws at the early stage of the design process.

However, this analysis method is not proposed as a full substitute for other usability evaluation methods. Analytical methods can only predict the usability of a design; user reports reveal the actual situation. Moreover, user-based methods offer a broader scope of understanding, such as personal preferences, aesthetics, and user satisfaction [see Helander, Khalid, and Tham, 2001]. Rather, FTDA is conceived as a tool for the early prototyping stages of a design process..

Copyright © 2004 by ICAD2004

4.3 FUTURE WORK

A substantial amount of work needs to be done to further formalize this method. In addition, case studies that involve a variety of consumer products are required for understanding and improving the generalizability of FTDA.

There is also a need to investigate how FTDA for product design can be further developed to make use of concepts such as design hierarchies and zigzagging. This is important because user goals and tasks are often described in terms of abstraction levels, and a mix of abstraction levels also leads to coupling [Helander and Lo, 2003].

As with any design and analysis methods, there is a need to evaluate how this method can be used by engineers and designers.

5 ACKNOWLEDGMENTS

We wish to express our thanks to Roger Jiao Jianxin for his many valuable comments during the process of writing this paper.

6 REFERENCES

Card, S., Moran, T., and Newell, A., *The Psychology of Human-Computer Interaction*, Hillsdale, NJ: Erlbaum, 1983.

Chin, J. P., Diehl, V. A., and Norman, K. L., "Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface", *Proceedings of Human Factors in Computing Systems '88*, pp. 213-218, 1988.

Dix, A., "Upside down As and Algorithms - Computational Formalisms and Theory", in *HCI Models, Theories, and Frameworks: Toward a Multidisciplinary Science*, J. Carroll (ed.), San Francisco, USA: Morgan Kaufmann, pp. 381-429, 2003.

Döner, D., *The Logic of Failure: Why Things Go Wrong And What We Can Do To Make Them Right*, New York: Henry Holt, 1996.

Dumas, J., Redish, J., *A Practical Guide to Usability Testing,* Norwood, NJ: Ablex, 1994.

Hamacher, N., Marrenbach, J., and Kraiss, K., "Formal Usability Evaluation of Interactive Systems", in *8th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design, and Evaluation of Human Machine Systems*, G. Johannsen (ed.), pp. 577-581, 2001.

Han, S. H., Yun, M. H., Kwahk, J., and Hong, S. W., "Usability of Consumer Electronics Products", *International Journal of Industrial Ergonomics*, Vol. 28, pp. 143-151, 2001.

Helander, M. G., and Jiao, J., "Coupling in Design of Human Computer Interaction. Implications for Usability", in *Proceedings of the Second International Conference on Axiomatic Design - ICAD2002*, B. El-Haik, and D. Tate (eds.), Cambridge, MA: Institute for Axiomatic Design, 2002.

Helander, M. G., and Lin, L., "Anthropometric Design of workstations", in *Proceedings of the First International Conference on Axiomatic Design – ICAD2000*, D. Tate (ed.), Cambridge, MA: Institute for Axiomatic Design, pp. 130-138, 2000.

Helander, M. G., and Lin, L., "Axiomatic Design in Ergonomics and an Extension of the Information Axiom", *Journal of Engineering Design*, Vol. 13(4), pp. 321-339, 2002.

Helander, M. G., and Lo, S., "Developing a Method for Formal Top-Down Analysis of Usability", in *Proceedings of SEAMEC 2003*, H. M. Khalid, T. W. Lim, and N. K. Lee (eds.), Kuching, Malaysia: UNIMAS, 2003.

Helander, M., Khalid, H. M., and Tham, M. P. (eds.), *Proceedings of the International Conference on Affective Human Factors Design.* London: Asean Academic Press, 2001.

Hirschi, N. W., and Frey, D. D., "Cognition and Complexity: An Experiment on the Effect of Coupling in Parameter Design" *Research in Engineering Design*, Vol. 13, pp. 123-131, 2002.

International Organization for Standardization, *ISO 9241: Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs)*, Geneva, Switzerland: International Organization for Standardization, 1998.

Jacko, J. A., and Sears, A. (eds.), *The Human-Computer Interaction Handbook*, Mahwah, NJ: Lawrence Erlbaum Associates, 2003.

Jordan, P. W., *An Introduction to Usability*, London: Taylor & Francis, 1998.

Karwowski, W. (2003), "Achieving Compatibility in Human-Computer Interface Design and Evaluation", in, *The Human-Computer Interaction Handbook*, pp. 1226-1238, J. A. Jacko, and A. Sears (eds.), Mahwah, NJ: Lawrence Erlbaum Associates, 2003.

King, J. R., "Machine-Component Grouping in Production Flow Analysis: An Approach Using a Rank Order Clustering Algorithm," *International Journal of Production Research*, Vol. 18(2), pp. 213-232, 1980.

Lewis, C., and Wharton, C., "Cognitive Walkthroughs", in *Handbook of Human-Computer Interaction*, 2nd ed., M. Helander, T. K. Landauer, and, P. Prabhu (eds.), Amsterdam: Elsevier Science, pp. 717-732, 1997.

Luqi, J. A. G., "Formal Methods: Promises and Problems", *IEEE Software*, Vol. 14(1), pp. 73-85, 1997.

Miller. C., "The Human Factor in Complexity", in *Automation, Control and Complexity: An Integrated Approach*, J. Weyrauch, and T. Samad (eds.), Singapore: John Wiley & Sons, pp. 35-572000.

Nielsen, J., and Molich, R., "Heuristic Evaluation of User Interfaces", in *Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems*, J. Carrasco, and J. Whiteside (eds.), New York: ACM Press, pp. 249-256, 1990.

Norman, D. A., *The Psychology of Everyday Things*, New York: BasicBooks, 1988.

Quill, L., Kancler, D. E., Revels, A. R, and Batchelor, C., "Application of Information Visualization at Various Stages of System Development, in *Proceedings of the Human Factors and Ergonomics Society 45th Annual Meeting*, Santa Monica, CA: Human Factors Society, pp. 1713-1717, 2001.

Rode, C., *In the Eye of the Beholder - Visual and Verbal Cognitive Capacities in Complex Problem Solving* (Research Report No. 1), Zurich: Think Tools AG, 2001.

Copyright © 2004 by ICAD2004

Rubin, J., *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests,* New York, NY: John Wiley & Sons, 1994.

Sadun, L., *Applied Linear Algebra: The Decoupling Principle*, New Jersey: Prentice Hall, 2001.

Saiedian, H., "An Invitation to Formal Methods", *IEEE Computer*, Vol. 29(4), pp. 16-30, 1996.

Sanderson, P. M., "Cognitive Work Analysis", in *HCI Models, Theories, and Frameworks: Toward a multidisciplinary Science*, J. M. Carroll (ed.), San Francisco, CA: Morgan Kaufmann, 2003.

Schreyer, M., and Tseng, M., "Hierarchical state Decomposition for the Design of PLC Software by Applying Axiomatic Design", in *Proceedings of the First International Conference on Axiomatic Design – ICAD2000*, D. Tate (ed.), Cambridge, MA: Institute for Axiomatic Design, pp. 264-271, 2000.

Spivey, J. M., *The Z Notation: A Reference Manual*, Englewood Cliffs, NJ: Prentice Hall, 1989.

Stanton, N. A., and Diaper, D. (ed.), *Handbook of task analysis for human-computer interaction*, Mahwah, NJ: Lawrence Erlbaum Associates, 2003.

Stanton, N. A., and Young, M. S., "Giving Ergonomics away? The Application of Ergonomics Methods by Novices", *Applied Ergonomics*, Vol. 34, pp. 470-490, 2003.

Steward, D. V., *Systems Analysis and Management: Structure, Strategy, and Design*, New York: Petrocelli Books, 1981.

Suh, N. P., *The Principles of Design*, New York: Oxford University Press, 1990.

Suh, N. P., *Axiomatic Design: Advances and Applications*, New York: Oxford University Press, 2001.

Weir, G. R. S., "Living with Complex Interactive Systems", in *Human-Computer Interaction and Complex Systems*, Weir, G. R. S. & Alty, J. L. (eds.), New York: Academic Press, pp. 1-21, 1991.

Weng, G., Bhalla, U. S., and Lyengar, R., "Complexity in Biological Signaling Systems", *Science*, Vol. 284, pp. 92 – 96, 1999.

Wing, J. M., "A Specifier's Introduction to Formal Methods", *IEEE Computer*, Vol. 23(9), pp. 8-24, 1990.

Woods, D. D., "Coping with Complexity: The Psychology of Human Behaviour in Complex Systems", in *Tasks, Errors and Mental Models: A festschrift to celebrate the 60th birthday of professor Jens Rasmussen*, L. P. Goodstein, H. B. Andersen, and S. E. Olsen (eds.), New York: Taylor & Francis, pp. 129-147, 1988.

Yates, F. E., "Complexity and the Limits to Knowledge", *American Journal of Physiology*, Vol. 4, pp. R201-R204, 1978.

Copyright © 2004 by ICAD2004