ORIGINAL ARTICLES

SMALL MULTIPLE ORDERED EVALUATION AND INTERFACE DESIGN FOR INTERACTIVE SURVEYS

Eric COOPER and Katsuari KAMEI

College of Science and Engineering, Ritsumeikan University, Kusatsu, Shiga 525-8577, Japan

Abstract: This paper provides a fresh look at the central data collection technique in *Kansei* engineering, the questionnaire or survey. The focus is the construction of the survey interface. We detail the construction of an original method of evaluation for computer-based surveys that is less fatiguing, has fewer acclimation problems, and requires less time than other methods, such as paired comparison. This method, Small Multiple Ordered Evaluation (SMOE) has been used in several *Kansei*-related projects. We compare SMOE with a common relative, paired comparison, from the viewpoint of human-computer interface evaluation models. Time trials and subject interviews show that SMOE, which is more complex by these methods of measurement, is quicker and less fatiguing than paired comparison. This analysis leads to further discussion of the survey interface in general, including interactive color balance experiments and common *Kansei* methods such as semantic differential. The main object of the paper is to discuss human-computer interaction concepts as they relate to questionnaires the field of *Kansei*, which depends so heavily on these types of surveys.

Keywords: *Interactive surveys, Interactive questionnaires, Usability engineering, Survey engineering*

1. INTRODUCTION

Surveys have played essential roles in *Kansei* engineering for the collection of data. Nearly every *Kansei* system relies on some type of survey to collect data about the sensibilities of target subjects. This is because surveys are more direct, easier to implement, and provide data that is easier to analyze and interpret than physical measurements such as eye movements and electrocardiograms. So questionnaires and surveys, which have formed the central method of data collection in *Kansei* engineering will continue to play a large role in *Kansei* engineering, as well as contributing to a number of related fields, including psychology, sociology, and market research. These days, surveys on interactive computer systems are becoming more and more common. However, the common types of survey may not be optimal interfaces for human-computer interaction. Usability is a serious problem for *Kansei* engineers because the survey is a costly and time-consuming part of the *Kansei* engineering process. Bad interfaces may cause large numbers of non-response data or misinterpreted data, which means wasted time and/or money.

As more and more surveys go online, the interface design of the interactive survey has become a critical, and often overlooked, part of the *Kansei* engineering process. Furthermore, the same issues are present on

both ends of *Kansei* projects that put together whole software packages devoted to *Kansei* engineering. On one end, data collection is performed by questionnaire and, after the system is complete the user enters *Kansei* information by a similar interface. These interfaces share similar usability engineering problems. This paper discusses quantitative and qualitative experience with several *Kansei*-related surveys, primarily with the interactive survey Small Multiple Ordered Evaluation (SMOE), but also with the Cooper-Kamei color conspicuity experiments and other *Kansei* experiments that have been constructed in our laboratory.

2. BACKGROUND

2.1. Types of Survey

Questionnaires and surveys as experimental tools were developed primarily in the field of experimental psychology. Some of common types of scaling are Likert scale, Guttman scale, Thurstone scale, semantic differential, and paired comparison. In the Likert scale, the subject responds to statements on a scale from agreement to disagreement. The Guttman scale only requires subjects to indicate agreement or disagreement in a strategy to develop a hierarchy of mutual dependence between these statements. The Thurstone scale is developed in a similar way, with judges

separating statements into categories and final scoring being based on subjects' approval or disapproval to the given statements. Paired comparison is simply a method of sorting stimuli asking the subject to select one of a pair and doing this for all possible pairs of the given group of stimuli [1]. Alternatives to paired comparison include sorting of a whole group at once, usually done with a group of cards representing visual stimuli to be sorted. This type of comparison is the ancestor of the Small Multiple Ordered Evaluation survey described in section 3.

The most common type of survey used in *Kansei* engineering has been the semantic differential, or SD survey. This survey was based primarily on the work of Osgood et al [1]. In the survey, two opposing adjectives are placed on each end of a scale and the subject of the survey chooses a response that reflects the degree to which each of the adjectives applies, which is a set of responses, usually an odd number. A common variation allows the subject to mark the scale to indicate a more precise response or even to make an indication as to the quality of response. Engineering this type of survey in HTML or a multimedia programming language seems to be a relatively straightforward procedure but there are some common pitfalls, discussed in Section 4.

All of these types of questionnaire can be implemented on a graphical user interface (GUI) but the success of the implementation, the burden on subjects and researchers, and the validity of the data obtained depend heavily on the usability of the survey interface.

2.2. Paper vs. GUI Surveys

With the coming of the digital age, it was once thought that paper would be in museums only. However, paper documents are easier than ever to create with computers and the use of paper documents has increased over the last decade rather than decreased [2]. One exception may be the case in which the document is a tool for collecting data, such as a questionnaire or survey. Paper questionnaires simply require too much input time to be practical. Electronic surveys offer immediate storage and pre-analysis preparation of the data. After storage, the electronic database can retrieve, manipulate, and analyze the data instantly. The question of permanence has been rendered moot, as data storage becomes more reliable and electronic storage offers unlimited replication.

Another advantage of GUI-based surveys is the ability to collect data that is not available on the pencil and paper interface. This is particularly relevant to the present discussion of the questionnaire or survey as interface because supplementary data about the subjects' interaction with the survey can be useful both for evaluating the interface data and for evaluating subject data. An excellent example of data collected that is not given directly by the user is the time between events, an aspect of both interface and data that is not available in the paper interface. If the subject took longer than others to respond, was there difficulty understanding the survey or interface? If the subject took milliseconds for each response and chose the same response twenty times in a row can that data be thrown away as "no response"? Without recording the time, these important facts cannot even be considered. The relationship of time to interface development is discussed in more detail in Section 2. Other useful evaluation data is the keystrokes and mouse motions, in addition to responses that are later changed. For example, in the semantic differential survey, every response made is recorded, instead of only the final response. If the researchers are concerned about the volume of data collected, the modern computer is capable of storing both a full-data file and an abbreviated data file at negligible difference in development time and speed, another benefit of electronic surveys. Over networks, the abbreviated data, containing only the necessary response and subject data, is sent as a separate transaction first and the supplementary data, all of the data collected, is sent afterwards so the important data are not lost due to network interruption.

Although paper is a very portable medium, surveys can be conducted over the networks, for a much wider and more accurate sample. Conducting interactive surveys at long distances sounds simple enough except to anyone who has actually conducted a survey over a network. Most surveys require some monitoring of the survey process that cannot be done over the network. Monitoring is another reason to record the supplemental data discussed above when working with subjects that cannot be seen.

There are two fallacies that lead to bad design of electronic surveys and questionnaires. The first fallacy is that such surveys must be based on paper designs. Actual testing cannot support the argument that check boxes and

other features inherited from paper interfaces are more comfortable to the user. Users have adapted to new and different interfaces such as the desktop computer's graphical user interface. The other fallacy, related to the first, is that following a pattern consistent with previous designs is effective. When a better alternative may exist, does the questionnaire designer have to use radio buttons simply because these buttons are usually used? This fallacy is common in interface design and has been refuted by Raskin [3] and Grudin [4]. Because electronic surveys are designed after previous paper surveys, paper designs may seem more intuitive, without testing or considering the alternatives. However, abandoning the analogy of recording each response with a pencil leads to electronic surveys that are more efficient.

2.3 Interface Usability Evaluation

Although *Kansei* engineering is often considered a division of human interface engineering, little attention has been paid to the actual interfaces used in *Kansei*. In this section, we discuss some of the major issues that interface designers look at, by describing some usability engineering tools from an engineering perspective.

Although Fitts' law and Hick's law both originated in the field of experimental psychology, they have contributed a lot to theories of human-computer interaction as well. They both deal with the common scenario of a user selecting an item on an interface. Both compute time required to perform an action, often used to evaluate the efficiency of an interface. Hick's law describes the time required to make a selection from a group of possible choices, for example to choose between several responses on a survey question. Fitts' law describes the time required to move to a target, for example to move a mouse cursor to an on-screen button.

Hick's law, also known as Hick-Hyman Law [5], states that response time increases as a loglinear function of the number of choices [1, 3, 5]. In other words, from an interface point of view, there is a big difference between offering five choices and nine choices for each questionnaire response. The difference may seem irrelevant for the usual survey design but a good question is: Does the number of choices in the interface have to equal the number of possible responses? In section 4, we discuss an example of an interface that uses successive binary selection for a range of nine possible responses.

Fitts' law deals with the size and location of choices on an interface. Not surprisingly, larger and more strategically placed targets are easier (faster) to reach [1, 3, 6]. Yet some interface designers pay little attention to the size and placement of objects to be selected. The equation for Fitts' law is also loglinear so there is a big difference between a tiny slider with only a few pixels and one a little larger, as discussed in section 4. This difference can influence the data entered. For example, the subject may feel that she wants to go back and alter a previous entry but quickly decides not to because getting the mouse on the tiny indicator and dragging correctly requires just a little too much effort. Below, we will use Raskin's term Fitts' penalty for the extra time required to target small objects on the screen.

Besides Fitts' and Hick's, there are a wide range of tools available for evaluation of interfaces. One common tool, that incorporates Hick's and Fitts' laws, is the GOMS keystroke level model, part of a larger strategy of usability engineering called GOMS (Goals, Operators, Methods, and Selection) in which the cognitive process of a task is broken down into essential elements. In the GOMS keystroke level model, the elements are individual keystrokes, mouse clicks, etc. required to perform a task [3, 6, 7]. The time required to perform a task can be accurately predicted by summing a set of standard values for each step of a task. Few interface designers make an accurate analysis of every step required to perform a task for an accurate keystroke-level analysis but often such work is not necessary. For example, as discussed in Sections 3 and 4, repeated binary decisions are probably better made on a keyboard than with the mouse, especially if the two items offered are small targets with high Fitts' penalties. On the keyboard, the user, who has already positioned two fingers on the proper keys, simply decides and pushes the correct key. At the keystroke level, this is far more simple than finding the target, moving to the correct spot on the screen and clicking the mouse.

Although we do not strictly calculate times according to these methods in this paper, understanding them provides an idea of the quantitative and qualitative perspective from which we evaluate the survey designs discussed in sections 3 and 4. More importantly, these measures show that there is a wide range of quantitative

tools available to the survey designer to make designs that are better than previous designs.

One of the best ways to evaluate an interface, and one that is almost always overlooked when the designer is simply using previous designs that are assumed to be easy to use, is through actual use. *Kansei* engineers often test questionnaire data to eliminate redundant questions and to test the distribution of possible responses, to make sure that the data collected will be useful. However, testing of the interface is just as important and can be done simultaneously with preliminary data tests.

3. SMOE

3.1 SMOE History and Characteristics

Small Multiple Ordered Evaluation (SMOE) is a survey design for comparative evaluation such as is done with paired comparison [8-11]. The term small multiples is taken from Tufte, who discussed the use of large groups of many objects for the visual display of quantitative information [12]. The advantage of having many small objects on the a visual interface is that the human eye and brain can scan and extract information from large groups quickly. This also leads to improved algorithms for sorting objects.

The SMOE system has been used in several studies, including two studies of visualization design [9, 10] and one study of color conspicuity [11]. The SMOE method is basically the same as previous methods of sorting a group of cards, except that the sorting is done on a computer screen with a mouse. Objects are shown as visual images at full size one at a time, in random order. The subject browses through the whole set which is then shown as a group in reduced form on one screen. The original SMOE experiments were lined up in the shape of a triangle, ten objects with one at the apex and four on the base, to allow the subject the image of a hierarchy. The subject of the experiment uses the mouse to move an image by clicking and dragging the image to the new position. When the mouse button is released, the moved image takes the place of its current position and all of the other images shift in the direction of the original position. The subject views the stimuli at full size by clicking a reduced image twice or by moving an image outside the group. At full size, the subject can also browse the current order using [next] and [previous]

on-screen buttons. Clicking a full-size image returns the subject to the SMOE group sorting interface. This switch between full-size and reduced size is the only part of this interface that subjects did not pick up intuitively.

This SMOE interface was originally developed and employed for color support system evaluation, in which users rated color schemes in order of ease-of-use. [8, 9] Each set of one visual object (a graph, map, or diagram) shown with ten different color schemes was arranged for the SMOE group sort interface, as shown in Figure 1. The details of the color placement experiments that produced this data is not relevant to the discussion of the interface, except to mention that in these experiments no labeling or text was appropriate or necessary. The same was true for Fujihara et al [11], in which nine objects were sorted for color conspicuity. The interface evaluation experiments described in the following subsection were completed with ten sets of ten each for an evaluation of one hundred objects.

Figure 1: SMOE interface for sorting a set of graphs [9].

3.2 Interface Evaluation of SMOE

We compared SMOE to another common method of interdependent comparative evaluation, paired comparison, also known as pairwise comparison. The objects compared in these tests were the graphs, maps, and diagrams from our visualization experiments. Each pair is the same design with a different color scheme. Our paired comparison had a very efficient interface in terms of keystroke level analysis.

The first problem of paired comparison construction in our survey was that the full-sized images would not fit on our seventeen-inch monitor. (The same monitor was used in all experiments). So the user toggled between the

two images using the [space] key on the keyboard. After viewing both images as much as desired, the subject of the experiment pushed the [return] key to indicate the design that was judged to be easier to read and the experiment moved on to the next pair to be evaluated. A basic keystroke level analysis shows that the user has only two choices: [return] or [space] and there is no homing or targeting required.

Two important asides need to be made here. First, paired comparison does not require a mouse. If we happened to have images that were small enough to place each pair on the screen at once, there would still be no need for a mouse, which would introduce a time-consuming select-target-click, routine where it is not needed. The Fitts' and Hick's times would not be prohibitive for two large targets but an even worse design for paired comparison would be to have two radio buttons (or any buttons at all) next to the two stimuli for selection. Assuming the two stimuli are somewhat larger than the buttons, using these buttons only incurs a huge and unnecessary Fitts' penalty. The discussion of the use of radio buttons in surveys is continued in Section 4. To show the method's relationship with one non-comparison type survey, we also did a one-question SD survey.

The paired comparison survey had 450 sets of pairs to compare, 10 sets of 45 each. The main difficulty of this is the drudgery of completing the same set of actions 450 times and the fatigue on the eyes. Paired comparison took an average of 45 min. per evaluator. By contrast, our SMOE survey took an average of 31 minutes per user. Evaluators reported that the SMOE survey was not fatiguing, allowing them time to make a proper evaluation. Subjects in the paired comparison reported eye fatigue and difficulty concentrating from repetitive action [8].

A GOMS keystroke evaluation, as described in [3], shows that the basic actions in SMOE require far more time than in paired comparison. The paired comparison requires only the operation of two keys and no targeting of on-screen objects or careful placement. Calculating by a single operation alone, SMOE should be many times slower than the paired comparison with two key input. Yet testing shows that on the whole, SMOE is faster than paired comparison. One reason that SMOE is faster is that the interface operations are far more efficient, requiring fewer steps. The number of steps for paired

comparison can be reduced with a win-loss tree but the best sort algorithm that paired comparison produces is the bubblesort algorithm, considered to be an inefficient sorting algorithm [13]. The SMOE interface allows the subject to implement any number of more efficient and intuitive algorithms.

To establish the validity and reliability of SMOE, we compared the total scores for each design on SMOE to the total score for paired comparison [8].

Survey interface designers create more efficient interfaces and obtain more accurate data when they consider both keystroke level operations and survey processes for the survey design. The study shows that new methods of interactive survey can outperform previous standards and therefore there is no reason to stick to previous designs simply because they are commonly used.

3.3. Other SMOE Construction Guidelines

There are some differences between the previous uses of SMOE and future uses that must be outlined here. Previous uses of SMOE were for purely visual objects. To adapt SMOE to, for example, car design, some changes are needed. Labels not only increase the size of targets and make scanning easier [3], but they also allow the subject to consider aspects of the stimulus other than appearance. Small information windows that appear on mouse rollover and disappear when the image is clicked would also assist the subject to make a clear comparison of the attribute or sensation in question.

Floating windows may also reduce or eliminate the problematic switch between full-size and reduced-size parts of the survey. A feature to allow zooming in and out could also reduce the modality [3] of the interface. If the browsing interface is necessary, arrows left and right are probably good shortcuts for the [next] and [previous] buttons. Future variations on SMOE will expand upon its uses, while improving and adapting the interface.

4. KANSEI INTERFACE POINTS

4.1. Common SD Survey Pitfalls

Because of the prevalence of the SD study in *Kansei* engineering, this section discusses some of the problems of designing SD surveys, from our own survey design experience, as well as from the point of view of the quantitative tools described in section 2.

The most common way to implement SD in HTML (hypertext markup language) or other interface is to use so-called radio buttons. These are a group of screen objects indicating response selection that are mutually exclusive, in other words they allow only one selection. When the user makes a different selection, the previous one is eliminated and only the new one remains. Response targets are usually indicated by small circle or raised diamond shape. These shapes are almost always much smaller than the space allows, giving them a high Fitts' value. The user takes much more time to home in on and select the small buttons or check boxes than larger targets, such as the large buttons. To make matters worse, the small targets are usually round, or diamond shaped and therefore have even smaller total area than square shapes that fill the available area.

Figure 2: Typical implementation of SD interface has tiny targets for a high Fitts' penalty.

Another problem with radio buttons is that they require the user to perform unnecessary mouse movement and selection, which in a GOMS keystroke level analysis or similar evaluation, is much more costly than a simple key selection.

Raskin says that for mutually exclusive choices "there is rarely reason to use other mechanisms" besides radio buttons [3]. From a Fitts' law point of view, that is only true when large radio buttons are available or when labels are also part of the selection target. Also, in *Kansei* surveys, unlike in the common interface where radio buttons are used, questions are normally marked in succession on the screen rather than in random order so allowing user access to all at once is unnecessary.

The radio button is sometimes replaced by rows of sliders. These GUI input devices require the user to home in on a small target, the slider position indicator, click and hold, and slide the indicator, then release. This action is completely natural on an actual sliding input device, such as those found on a stereo equalizer, because the user receives excellent tactile feedback. The common mouse offers no such feedback, making this choice even worse than the radio buttons. To make

matters worse, the slider indicator is sometimes only a few pixels in width, for a high Fitts' penalty. Presumably, the slider offers the whole range of a line of choices instead of a limited set, but there are probably more efficient ways of implementing this concept, as discussed below.

The following subsection discusses alternatives to the standard radio button.

4.2. Alternatives to Radio Buttons

One alternative to the radio-button approach is to simply allow a binary selection. In other words, the indicator starts at some place, for convenience probably the middle position, and the subject selects left or right to move the indicator. This is the approach taken in the Cooper-Kamei conspicuity experiments, in which the user selects from two colors [9, 10].

Figure 3: The color conspicuity survey interface, binary selection, large targets, nine responses [9, 10].

These experiments used the mouse for a direct selection each time, allowing the user to understand exactly which selection was made. The Fitts' penalty was small because the targets were large compared to buttons or slider indicators. However, in hindsight, the mouse is not necessarily an efficient method of binary selection. The arrow keys may have worked better. As noted in sections 2 and 3, a binary selection allows the user to have two fingers permanently in place for the selection.

We offer this proposal for simple and direct interface. The left and the right arrow keys can move the current indicator toward or away from response extremes. The up and the down buttons are used to move from one

response to the next, or back, as required. This only requires the addition of a clear indicator, such as a highlight, for where the current selection is being made. If mouse input is necessary or desired, the labels themselves become the buttons for a quicker binary selection, as shown in Figure 4. Repetitive binary selection is faster when there are few selections, provided the targets are large. For more choices, direct selection is the most efficient.

For quick construction, large buttons that function as radio buttons, are better than common default radio buttons. Takatera et al give one example of an SD survey with well-designed buttons [14].

Figure 4: Binary SD selection with mouse button. Arrows indicate three items: position in survey, position of response, key shortcuts.

Using buttons to indicate an exclusive choice is an ugly interface design decision, and not recommended in most interface guidelines, but preferable to the standard tiny radio button for the *Kansei* questionnaire, in which the user must repeat the selection process dozens or hundreds of times. A time-consuming compromise is to design and implement large radio buttons that look and operate just as the small ones, as shown in Figure 5. Manual construction is worth the effort for laboratories that produce a lot of this type of survey or for very long surveys.

Figure 5: A better design for radio buttons in surveys has larger targets but requires more programming.

Even for the normal survey subject these changes will improve data collection but if the survey sample may include subjects with vision or movement difficulties, such as some elderly subjects, these changes are mandatory.

Similarly, sliders with very large indicators for easy control are much better than the default sliders offered in

browsers and programming environments. The part that does the position indicating can be a small line or arrow on the larger control. Also, if the designer does not allow movement of the slider with a mouse click anywhere on the axis, repetitive clicking and dragging becomes tedious after only a few survey questions.

5. EVALUATION OF EMPATHY MODEL

This paper described some alternatives to common survey or questionnaire interfaces. Several examples of actual surveys and their evaluation provide the background for the discussion of survey interfaces. We found that alternatives to common designs reduce the burden on the survey subject and the time required for researchers to collect data.

Although this topic is relevant to survey design and implementation in many fields, the concepts are especially critical in *Kansei* engineering, which relies very heavily on questionnaires to collect the data that becomes the *Kansei* system, as well as the interaction with the completed system itself.

REFERENCES

1. A. M. Coleman; A Dictionary of Psychology; Oxford University Press, New York (2001).
2. Z. Lui, D. G. Stork; Is Paperless Really More? Communications of the ACM, 43 (11), 94-97.
3. J. Raskin: The Humane Interface: New Directions for Designing Interactive Systems, ACM Press, Addison-Wesley, Boston (2000).
4. J. Grudin; The Case Against User Interface Consistency, Communications of the ACM; 32 (10), 1164-1173 (1989).
5. K. R. Paap, R. J. Roke-Hofstrand; Design of Menus. In; M. Helander (Ed.), Handbook of Human-Computer Interaction; Elsevier, Amsterdam, 205-235 (1988).
6. A. Dix, J. Findlay, G. Abowd, R. Beale; Human-Computer Interaction, Prentice Hall, New York (1993).
7. B. E. John; Why GOMS?, Interactions, 2 (4), 80-89 (1995).
8. E. Cooper, K. Kamei; An Interactive Survey for Evaluation of Color Placement Support Systems,

Proceedings of Fourth Workshop on Evaluation of Heart and Mind: Heart and Mind '99, Okayama, pp. 71-72 (1999).

9. E. Cooper, K. Kamei; Development of a Color Balance Support System, Journal of Human Interface Society, 1 (4), 73-80 (1999).
10. E. Cooper, K. Kamei; A Study of Color Conspicuity for Ease-Of-Use Inference in Visualization, Color Research and Application, 27 (2), 74-82 (2002).
11. N. Fujihara, K. Katsuari, E. Cooper; A Proposal of Conspicuity Model of Street Sign Based on Color Placement Experiments, Proceedings of the Human Interface Symposium 2001, 599-602 (2001).
12. E. R. Tufte; Envisioning Information; Graphics Press, Connecticut, 81-95 (1990).
13. D. Harel; Algorithmics: The Spirit of Computing, Addison-Wesley, Massachusetts, 2nd Edition, 138-143 (1992).
14. M. Takatera, T. Furukawa, Y. Shimizu, M. Kamijo, S. Hosoya, T. Morisaki, A. Ohtake, Apparel Products Search System Considering Individual *Kansei* Evaluation, KANSEI Engineering International; 1 (2), 1-8, See Figure 6 (2000).