A SCENARIO ELICITATION METHOD BASED ON OPERATION SEQUENCE PATTERNS

Hwansu Park [1] and Wan C Yoon [2]
Department of Industrial Engineering, KAIST,
Gusong-dong, Yuseong-gu, Daejeon 305-701, South Korea
{ lespark[1], wcyoon[2] }@kaist.ac.kr

ABSTRACT
Scenario uses have become popular in various fields such as interface design, requirement engineering, and usability evaluation. Although scenarios can help design and evaluation by describing possible ways to accomplish relevant tasks using given interfaces, there are few formal systematic methods to compose the most desirable set of scenarios for design or evaluation. For efficient and effective usability evaluation, it is necessary to select and organize scenarios so that the included tasks contain all important procedures and interface controls without too much redundancy. This article proposes a scenario elicitation method based on operation sequence patterns. A worked example is provided for illustration.
KEY WORDS Scenario elicitation method; Usability evaluation; Operation sequence patterns

1. Introduction

The logical complexity and functional versatility of products increase the importance of usability evaluation. Many usability evaluation methods exist for finding usability problems and predicting usability of products including heuristic evaluation and user testing. These methods frequently use scenarios as a means of selecting tasks to be evaluated (Nielsen, 1995).

Scenario generation has attracted much attention of researchers, because scenarios played diverse but important roles in design-related activities such as usability evaluation, scenario-based design, object-oriented design, and requirement engineering (Hertzum, 2003). Leite et al (2000) classified scenario generation heuristics into top-down and bottom-up approaches. In the top-down approach, Booch (1994) and Sutchliffe (1997) first determined primary scenarios and then detailed them adding more specific scenarios. Booch represented fundamental system functions in primary scenarios and describe variations on the theme of the primary scenarios in secondary scenarios (Leite et al, 2000). Sutchliffe used scenarios in requirements engineering. First, grounding scenarios were developed based on the preliminary domain analysis and then an early design prototype was validated against the scenarios in order to elicit requirements. In the mean time, the scenarios were detailed taking the available interface into account. Hsia et al (1994) also suggested a top-down approach that used a scenario tree, which included all possible states, to generate possible flows of scenarios. This scenario generation method emphasized on ensuring internal completeness of the generated scenarios.

In the bottom-up approach, Dano et al (1997) and Potts et al (1994) first described elementary use-cases, and then generalized them. Dano et al collected use cases and then created a Petri Net for each use case in order to set up inter-use-cases links. Potts et al first represented scenarios at the detailed level in terms of episodes or phases that were sequences of fine-grained actions and then composed families of scenarios using uses relationships.

Despite the extended studies concerning scenario generation, it remains a challenging task to compose an economical set of scenarios that includes all important task elements while avoiding heavy redundancy. Because software programs and electronic devices are getting more sophisticated with increased numbers of functions, the number of required scenarios to insure comprehensiveness tends to grow. It thus becomes important to maintain a parsimonious number of scenarios compared to the size of functionality. This research aims to develop a scenario-eliciting method that helps include all important elements of interface design without undue omission and too high redundancy. The method is devised with an emphasis on examining the more likely causes of user errors.

The proposed method is based on the concept of *operation sequence patterns*. As a general rule of user interface design, procedural consistency for similar tasks and consistent association between operations and interface elements are well expected and tend to be realized to a certain degree. These operation sequence patterns strongly influence the overall cognitive usability since users exploit them to avoid mental complexity in learning or to guess the correct procedures during uses. User face difficulties when this expectation is violated.

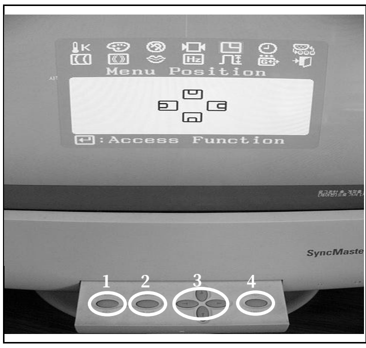
The proposed approach ensures to include all the frequent operation sequence patterns while not missing peculiar interface elements that may cause user difficulties. The scenarios are composed by systematically selecting tasks based on the operations they contain. The method entertains more prototypical tasks pursuing representativeness and completeness of the scenario set and at the same time tries to avoid tasks that are too similar with the already chosen ones.

2. Method

The proposed scenario elicitation process is explained with the parameter-setting tasks of a computer monitor. Users carry out fifteen different tasks including ‘brightness adjustment’, and ‘color density adjustment’, ‘bent balance adjustment’, ‘frequency adjustment’, and ‘removing display spot’ among others. Figure 1 shows the interface for monitor setting with interface controls.

Operation sequences and the interface controls to be used for each operation are somewhat different depending on the tasks. Table 2 shows examples of different operation sequences and interface controls used in the sequences. It is important to select tasks so that all the elementary operations are included in the set of scenarios.

For the association between the operations and controls, ‘change setting up’ operation is executed by the button 3 in ‘brightness adjustment’ task, but by the button 4 in ‘removing spot figure’ task. Such inconsistency is a primary cause of potential user errors. Not to miss the tasks that are prone to potential user errors, it is important to include such peculiar associations between operations and controls.



Button number	Button name	Button Function
1	Location button	To display or hide the menu for location adjustment.
2	Shape button	To display or hide the menu for shape adjustment.
3	Direction button	To scroll the menus or change setting up.
4	Menu button	To display or hide the main menu list.

Figure 1.The interface for a monitor setting

Tasks	Operation sequences				
Brightness adjustment	Display menu (Button 4) →	Navigate menu (Button 3) →	Select menu (Button 4) →	Change setting up (Button 3) →	Hide menu (Button 4)
Frequency adjustment	Display menu (Button 4) →	Navigate menu (Button 3) →	Select menu (Button 4) →	→	Hide menu (Button 4)
Removing spot figure	Display menu (Button 4) →	Navigate menu (Button 3) →	→	Change setting up (Button 4) →	Hide menu (Button 4)

Table 1. Examples of different operation sequences and interface controls

The proposed scenario generation method proceeds in the following five steps.

Step 1 – identifying elementary tasks which a product supports

Elementary tasks usually correspond to functions that a product supports. In this example, the total number of the elementary tasks was fifteen.

Step 2 – identifying the procedure of operations carried out in each elementary task

An *operation* is an elementary action by the user, which produces a system response or a state change. In step 2, the procedure of each task is described as a sequence of operations. Table 2 is a part of the task-operation contingency that appeared in the example.

tasks / operations	Brightness adjustment	color density adjustment	Bent balance adjustment	Frequency adjustment	Removing display spot figure	…
Display Menu	V	V	V	V	V	…
Navigate Menu	V	V	V	V	V	…
Select Menu	V	V	V	V		…
Change setting up	V	V	V		V	…
Save changes						…
Cancel changes						…
Disappear menu	V	V	V	V	V	…

Table 2. Example of task-operation table in monitor setting

Step 3 – finding controls corresponding to operations in each task.

In step 3, controls are found in association with the operations in each task and summarized in an operation-control table. Table 3 is an example of the operation-control table for ‘brightness adjustment’ task.

controls / operations	Location button	Shape button	Direction button	Menu button
Display Menu				V
Navigate Menu			V	
Select Menu				V
Change setting up			V	
Save changes				
Cancel changes				
Disappear menu				V

Table 3. The operation-control table for ‘brightness adjustment’ task

Step 4 – grouping the tasks based on their operation sequence patterns

In step 4, the tasks are grouped to have similar operation procedures and controls. In this example, there are 8 operation sequence patterns for total 15 tasks. Table 4 shows the task groups based on operation sequence patterns and Table 5 shows the operation sequences and interface controls for each task group. Parentheses are used to indicate that only some of the tasks in a task group employ the particular operations.

Task groups	Tasks
1	location adjustment, OSD menu location adjustment
2	size adjustment
3	quadrangle shape adjustment, bent balance adjustment

4	inclination adjustment
5	frequency adjustment, input signal selection, OSD language selection
6	removing display spot figure, removing wave figure
7	reset settings
8	brightness adjustment, color density adjustment, OSD menu display time adjustment,.

Table 4. Task groups based on operation sequence patterns

controls / operations	Location button	Shape button	Direction button	Menu button
Display Menu	1, 2	3, 4		5, 6, 7, 8
Navigate Menu	1, 2	3, 4	5, 6, 7, 8	
Select Menu				2, 4, 5, 7, 8
Change setting up			1, 2, 3, 4, 7, 8	6
Save changes				7
Cancel changes				7
Hide menu	1, 2	3, 4		5, 6, 7, 8

Table 5. Operation sequences and interface controls for each task group

Step 5 – composing scenarios while selecting tasks

First, compose a typical scenario that includes tasks that are representative in their groups. The number of tasks should be controlled considering the purpose of the scenarios. From Table 5, the operation-control associations (i.e., the cells in the table) involved in this scenario are easily identified. Then, the task groups that are frequent in the remaining cells are identified. Compose another scenario that is centered on these tasks. <ore scenarios are composed repeating this composition-identification-elimination cycle until most cells are included. This can be thought as a greedy algorithm. However, whenever inevitable or desirable, parsimonious reuse of the already included cells is allowed. When only a few cells have remained, it should be tried to include the tasks in those cells into the existing scenarios since it becomes harder to compose a reasonable scenario including the remaining, usually peculiar associations; especially suppressing high redundancy

3. Conclusion

In this research, proposed is a new method to generate a set of scenarios based on operation sequence patterns. In the method, systematical analysis is performed to achieve completeness and reduce redundancy of the operation and interface elements especially to find potential user difficulties. Although the approach is systematic, it is still possible for the evaluator or designer to put different weights on tasks, operations, and interface controls as necessary. Expected frequency of those elements or available prior knowledge of usability problems may have to be reflected in such weighting for more effective and efficient evaluation. For example, it is at times recommendable that the experimenter focuses on specific tasks that are suspected to have usability problems (Cordes, 2001). For large size problems with numerous tasks and controls, the proposed approach can easily be elaborated with a quantitative similarity index to define task groups and also with more sophisticated assignment algorithms.

References

Booch, G.., (1994), Rep Object Analysis Design, Vol. 1(3), 3-6.

Dano, B., Briand, H., Barbier, F., (1997) "An approach based on the concept of use case to produce dynamic object-oriented specification", Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, 54-64.

Hertzum, M., (2003), "Making use of scenarios: a field study of conceptual design", International Journal of Human-Computer Studies, Vol. 58, 215-239

Hsia, P., Samuel, J., Gao, J., Kung, D., (1994), "Formal Approach to Scenario Analysis", IEEE Software, Vol. 11(2), 33-41.

Leite, J.C.S., Hadad, G.D.S., Doorn, J.H., Kaplan, G.N., (2000), "A Scenario Construction Process", Requirements Engineering, Vol. 5, 38-61.

Nielsen, J., (1995), Scenarios in discount usability engineering. In: Carroll, J.M. (Ed.), Scenario-Based Design: Envisioning Work and Technology in System Development. Wiley, New York, 59-84.

Park, J., Yoon, W.C., Ryu, H., (2000), "Users' recognition of semantic affinity among tasks and the effects of consistency", International journal of human-computer interaction, Vol. 12(1), 89-105.

Potts, C., Takahashi, K., Anton, AI., (1994), "Inquiry-based requirements analysis", IEEE Software, Vol. 11(2), 21-32.

Reiser, B. J., (1986). The encoding and retrieval of memories of real-world experiences. In J. A. Galambos, R. P. Abelson, and J. B. Black (Ed.), Knowledge Structure (71-99). New Jersey: Lawrence Erlbaum Associates.

Sutchliffe, A.G., Square, N., (1997), "A Technique Combination Approach to Requirements Engineering", Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, 65-74.

Wirfs-brock, R., (1995), Designing objects and their interaction: A brief look at responsibility-driven design. In Carroll, J.M. (Ed.), Scenario-Based Design: Envisioning Work and Technology in System Development. Wiley, New York, 337-360.