Aspect Model-Based Methods for Scenarios and Prototype Development

Youn-kyung Lim

Illinois Institute of Technology
350 N. LaSalle St.
Chicago, IL 60610 USA
yklim@id.iit.edu

Keiichi Sato

Illinois Institute of Technology
350 N. LaSalle St.
Chicago, IL 60610 USA
sato@id.iit.edu

Abstract

Prototyping is a critical method for exploring, communicating and testing new concepts in an interactive system development process. However, the criteria used in prototype model evaluation often do not address the original project objectives and issues. To improve this situation, a prototyping specification mechanism with aspect models and scenarios created through the aspect models are introduced, using Design Information Framework (DIF) as a foundation of organizing information necessary for prototyping. Scenarios created through the aspect models are introduced for prototype testing context definition. This paper presents the prototyping specification mechanism with its application.

1 Introduction

In an interactive system development process, prototyping is a critical method for exploring, communicating and testing new concepts. However, there are no clear mechanisms for using prototyping to evaluate design concepts in relation to the issues and problems identified in the earlier phases of the process (Alavi, 1984). Not only the evaluation of the test outputs, the testing itself tends to be executed without a structured way.

This research developed a mechanism for specifying, constructing and evaluating prototype models with consistent and systematic views. This prototyping specification method selects aspects to be prototyped and transforms the represented aspects into real prototype models. If a specific aspect of a product is clearly defined and represented, more effective evaluation of the solution would be possible (Ribeiro & Bunker, 1988) (Sato 1991). A scenario addressing original user experience specifies the context and tasks for testing the prototype models. This paper explains the prototyping specification mechanism based on these concepts through a design case of a digital video camera.

2 Aspect Models and Scenarios with the Design Information Framework (DIF)

The Design Information Framework (DIF) has been introduced as a unified design information platform for supporting activities in the design process. The DIF provides a structure for organizing design parameters relating to those design activities. The prototyping specification mechanism proposed in this research uses the DIF to specify the variables of aspect models which are embodied in prototype models and to translate the aspect models into scenarios which identify context of a test and necessary tasks users should proceed in the test. The aspects for the prototype specification are derived from requirements for the system development that reflect the originally addressed issues and problems.

The DIF has two different levels for representing information. The lower level of representation consists of Design Information Primitives (DIPs) that cannot be further decomposed into smaller conceptual units. The higher level of representation consists of Design Information Elements (DILs) composed of those DIPs (Lim & Sato, 2001) (Figure 1). An aspect of the system can be represented by several design parameters which correspond to DILs or DIPs in the DIF. The DIF also provides an interpretive mechanism for integration among different aspect models of a product. It helps designers to create solutions with a holistic view by identifying clear relationships among different aspects. A set of the aspect models of a use situation can also be used to create scenarios. The DIF enables the translation of the aspect models into a narrative format for the scenarios through its sentential compatible structure.

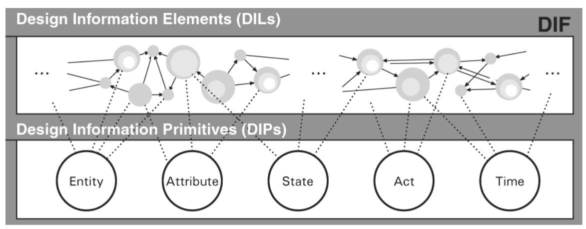


Figure 1: Structure of the Design Information Framework (DIF)

3 Prototyping Specification Mechanism with Scenarios and Aspect Models

Improving the usability of the digital video camera was the primary focus for this case. The process of this study is as follows: 1) set requirements for design, 2) identify problems through user observation, 3) generate a scenario for the representation of a use situation imposing the problems, 4) propose solutions to the identified problems, 5) develop prototype models to evaluate the solutions by using the prototyping specification mechanism, 6) set up the context and user tasks for testing with the created scenario, and 7) evaluate the solutions with test results.

3.1 Scenarios for Representing the Use Situation

For this case study, two requirements were given: 1) make relevant interface elements easier to find, and 2) help users to understand which mode they are in while accomplishing tasks. Aspect models to evaluate these requirements through prototype models were an operation sequence model, an interface layout model, and a state-transition model.

In the observation, the following problems were identified for the situation of taking still images with the digital video camera: 1) the user could not find the button for taking still images, identified from the operation sequence aspect, 2) it took time to find the relevant button to display the selected image, identified from the interface layout aspect, and 3) the user could not recognize whether the image was taken or not, identified from the state-transition aspect.

These requirements and problems were used as a foundation of creating solutions, and the requirements provide the rationalization of the solutions (McKerlie & MacLean, 1994).

The aspect models that revealed these problems were used to create a part of a scenario. The operation sequence model provided user's actions for the story of an event, the interface layout model provided the information of the system, and the state-transition model provided the reactions of the system to the user's actions.
These models formed the main story of the scenario. For the background information such as scene setting and actor description, the real situation observed through the user research was referred. Box 1 shows a part of the scenario created for this case. This scenario was used for preparing the test set up, which will be explained later.

Box 1: A part of the scenario created

David, who is a college student, needs to take a digital picture of a classroom for his project. He is in the classroom now, and he has a digital video camera to take it. The room is bright enough to take a picture without a night shot function. He first finds the scene to take by zooming in and out with the zoom lever. After finding the right scene, he tries to take the photo, but he can not locate the photo button among many other buttons. He presses several buttons near the "photo" label, and finally figures out the button that seems like taking a picture. However, what he thought he took is not stored in the memory. (…)

3.2 Aspect Models for Prototype Models and Scenarios for Test Set-up

We identified important information elements that should be included in the prototyping specification: *requirements* for prescribing which aspect models should be selected, *aspect models* that best illustrate the aspects of a system that the model represents, *variables of a prototype model* for evaluation determined by the selected aspect models, *functions for simulation* that define the external elements on the prototype models such as interface elements, *functions for performing prototyping testing* for capturing data through the monitoring and recording of user inputs, *media of a prototype model* determined according to the nature of variables and functions, *benchmark tasks* that represent the tasks which can be identified by a scenario, and *assumed context* that should be considered in a test which can also be identified by the scenario.
In this case study for constructing prototype models, the following aspect models were concerned: an operational sequence model mapped onto the interface layout and a state-transition model within the task sequence were used. The first aspect generated three alternative solutions toward how to re-design the interface layout to make the photo button easy to find. Regarding the second aspect, the main concern was how to provide feedbacks/outputs for users to distinguish the focusing mode from the image taking mode. Two ideas were generated from this aspect.
The functions that should be embedded to the prototype model 1 (Figure 2(a)) were zooming in and out, and still photo taking. These functions determined the input elements for the directly related user tasks such as a zoom lever, a photo button, some other input elements that would affect the time required for users to find the photo button such as a "focus set" button, and a "night shot" switch, and the output elements such as a beeping sound for monitoring user's actions in the test. Table 1 shows a partial specification for prototype model 1 by adapting the DIF structure.
For prototype model 2 (Figure 2(b)), the important factors were the relationships between inputs and outputs of the product, as well as between the outputs and states of the product. To effectively capture time information, automatic time stamping and product state recording functions were implemented for testing purposes.
For the test set-up, the created scenario was used to define the benchmark tasks that the subjects need to proceed in the test. The following tasks are extracted from the scenario as the benchmark tasks: zooming in and out, focusing, and taking a still image. Besides the benchmark tasks, the scenario guided to define an assumed context for the test. The background information such as

scene setting of the scenario was adapted to define it. In this case, the subjects assumed to be in a classroom to take a still picture, and the room was assumed to be bright enough to take the picture. In order to make consistency of subjects' task procedure, every subject followed the same operation sequence provided to them. After the test, several questions were asked for debriefing. During the test, subjects were asked to think aloud.

Table 1: A partial specification of prototype model 1 for concept A

Entities for a prototype model	**Attributes** for the entities	**Functions** for simulation	**Inputs**	**Outputs**	**Functions** for performing the test	**Inputs**	**Outputs**
Photo button	- Shape: circle - Size: R5.5(mm) - Location: (x, y) =(45(mm), 10(mm)) - Pushable				Notify if the photo button is pushed	Push the photo button	Make a beep sound
Zoom lever	- Shape: rounded rectangle - Size: 23.3(mm) x 13(mm), R1(mm) - Location: (x, y) =(65(mm), 10(mm)) - Slidable	Simulate zooming operation	Slide a zoom lever	Not needed			
Body	- Shape: rectangular parallelepiped - Front plane size: 80(mm)x110(mm)	N/A					

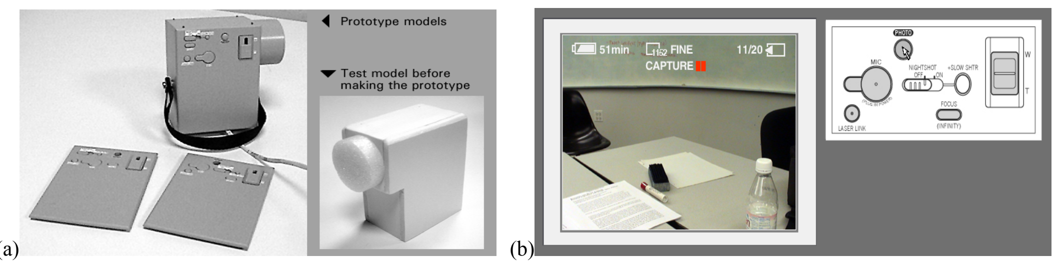
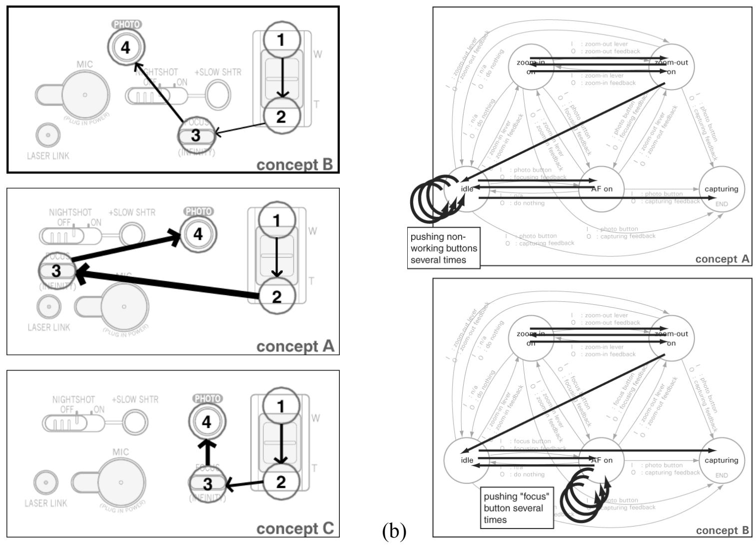
(a) (b)


Figure 2: (a) Prototype model 1 and (b) prototype model 2

3.3 Evaluation of Test Results

For the evaluation, the test results can be represented through the selected aspect models that were related to the requirements. They were an operation sequence model, an interface layout model, and a state-transition model in this case. The interface layout model corresponded to the first requirement, “make relevant interface elements easier to find,” and the state-transition model corresponded to the second requirement, “help users to understand which mode they are in while accomplishing tasks.” The benchmark tasks provided to the subjects formed the operation sequence model, and the operation aspect was mapped onto the other aspects imposed by the prototype models to produce output models for the evaluation (Figure 3).

The outputs for prototype model 1 illustrates that concept B is best in terms of reducing the time taking to find the relevant interface elements (Figure 3(a)). The outputs for prototype model 2 showed that, for both concepts, repetition of inputting on one state happened (Figure 3(b)). However, with concept A, the subjects could not go to the auto-focus (AF) state easily unlike concept B with which the subjects could go to the AF state. The several trials of focusing operations was caused by the unrealistic simulation by the prototype model. The closure look to the real output data with debriefing revealed that the subjects could understand that a still picture was not taken on the AF state which the original product system failed to make users recognize.



(a) (b)

Figure 3: (a) The output of the aspect models for prototype model 1 and (b) the output of the aspect models for prototype model 2

4 Discussion and Future Studies

Aspect models supported the effective evaluation of the prototype models as well as their construction, and scenarios provided relevant context to address the selected aspects in the test. However, some problems might occur because of the representation of only specific aspects in prototype models. This case suggests that the prototyping specification mechanism can be improved by examining how related aspects could be implemented in the prototyping mechanism to enhance the evaluation. The structural relationships among the information elements defined in DIF could effectively support this mechanism. This issue will be investigated with more case studies of developing prototype models with this specification mechanism.

5 References

Alavi, Maryam. (1984). An assessment of the prototyping approach to information systems development. *Communications of the ACM*, 27 (6), 556-563.

Lim, Youn-kyung and Keiichi Sato. (2001). Development of Design Information Framework for interactive systems design. In *Proceedings of the 5th Asian Design Conference*. Seoul, Korea: International Symposium on Design Science.

McKerlie, Diane and Allan MacLean. (1994). Reasoning with design rationale: practical experience with design space analysis. *Design Studies*, 15 (2), 214-226.

Ribeiro, Rita A. and Ralph E. Bunker. (1988). Prototyping analysis, structured analysis, prolog and prototypes. In *Proceedings of the ACM SIGCPR Conference on Management of Information Systems Personnel*. College park, Maryland: ACM.

Sato, Keiichi. (1991). Temporal aspects of user interface design. In *Proceedings of the '91 International Symposium on Next Generation Human Interface*. Tokyo: Institute for Personalized Information Environment.