

# An Integrated Environment for Testing and Assessing the Usability of Information Appliances Using Digital and Physical Mock-Ups

Satoshi Kanai<sup>1</sup>, Soh Horiuchi<sup>1</sup>, Yukiaki Kikuta<sup>2</sup>, Akihiko Yokoyama<sup>2</sup>,  
and Yoshiyuki Shiroma<sup>3</sup>

<sup>1</sup> Hokkaido University, Japan

kanai@ssi.ist.hokudai.ac.jp

<sup>2</sup> Cs Lab. Co. Ltd., Sapporo, Japan

<sup>3</sup> Sapporo City University, Sapporo, Japan

**Abstract.** Executing user-test of information appliances costs much due to fabrication of physical functional mockups, the securing of human subjects for the test, and preparation of testing facilities. Results of the user test have to be manually analyzed by usability engineers, and the analysis takes a couple of weeks. In this paper, we propose an integrated software and hardware environment for prototyping and assessing usability of information appliances where the users can operate a user-interface either of a 2D digital mockup or 3D digital mockup or a physical-mockup of the appliance under the test. This function enables engineers to test and evaluate the usability of the appliances both from the cognitive aspect and from the physical aspects.

**Keywords:** User Interface, Usability Test, Digital Mockup, Physical Mockup.

## 1 Introduction

Recently, information appliances such as digital consumer electronics have spread to general users. Product design conscious of “usability” of user interfaces (UIs) is needed in the manufacturers. As a method of usability assessments, “user-test” is recognized as the most objective one where the end users are directly involved in the test process. However, executing user-test of information appliances usually costs much due to fabrication of physical functional mockups, securing of human subjects for the test and preparation of testing facilities. Results of the user test have to be manually analyzed by usability engineers, and the analysis usually takes a couple of weeks. Therefore, an advanced environment for the user-test is strongly needed to remove the cost and the time barriers of executing the test.

In this paper, we propose an integrated software and hardware environment for prototyping and assessing usability of information appliances where the users can operate a user-interface either using a 2D digital mockup, 3D digital mockup or a physical mockup of the appliance under the test. This function enables engineers to test and evaluate the usability of the appliances both from the cognitive aspect and from the

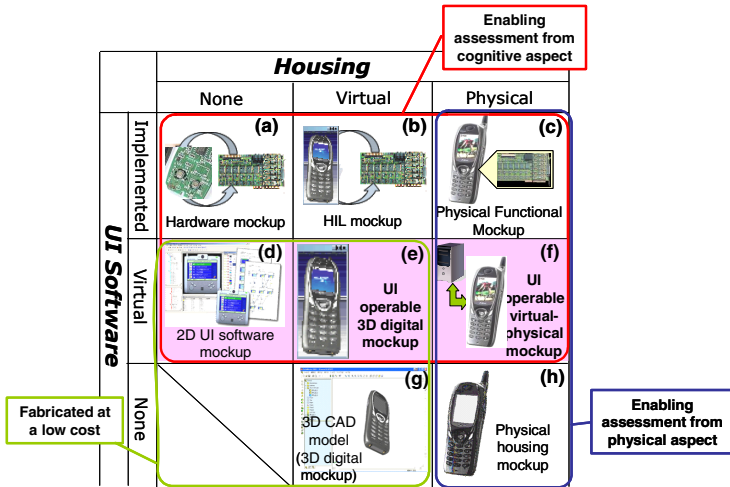


Fig. 1. Different kinds of mockups in developing information appliances

physical aspects, and to choose an optimal combination of the housing models and UI models according to the product type, constraints of cost and time, and required reliability of test results.

In the following sections, related works and features of the proposed environment are described, and detail functions of the environment are explained. Finally, an application of the environment

## 2 Related Works

As for the user test of information appliances, different kinds of mockups could be made corresponding to different embodiment combinations of the housing and UI software as shown in Figure 1. In user-test, developers have to evaluate these mockups from physical aspect and from cognitive aspect. Among those mockups, our environment deals with 2D UI software mockup (Figure-1 (d)), UI operable 3D digital mockup (e), and UI operable virtual-physical mockup (f).

As for 2D UI software mockup, there are some commercial prototyping tools [1,2] which have a UI simulation function. Moreover, there are a few tools which have usability assessments functions [3,4]. However, these tools were designed to develop UI for standard PC or mobile phone, and it is difficult for subjects to operate their UI with sufficient reality when executing the user test for appliances.

UI operable 3D digital mockups have been studied which consist of a virtual housing model (CAD model) and virtual UI software [5,6,7]. The virtual UI software can enhance the reality of interaction. However, these mockups do not necessarily provide functions of defining and executing the state-transition UI behaviors. The mockup

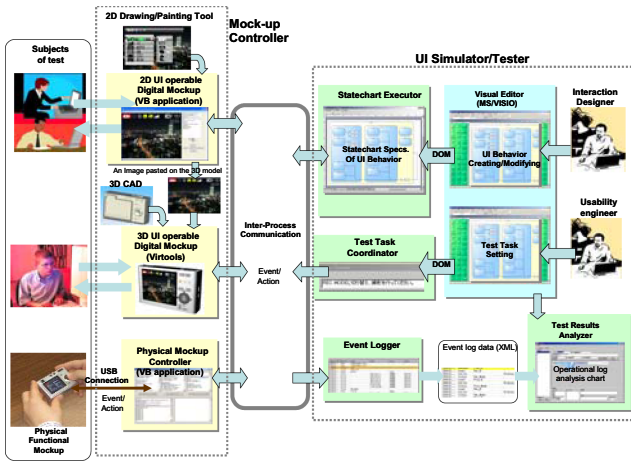


Fig. 2. An Overview of Testing and Assessing Environment

also does not have a physical housing, so that the usability assessment from physical aspect is ignored in this type of mockup.

On the contrary, UI operable virtual-physical mockup has a physical housing mockup and a virtual UI software simulation function [8,9,10,11]. The subjects can directly grasp a physical housing and can manipulate the UI elements on it. However, these mockup need the fabrication of the housing models and hardware connection between the physical mockup and UI software simulator.

The tools described above only have a function of UI simulation and of executing user test only by using one of the 2D digital mockup, 3D digital mockup or physical mockup. There is no tool so far where usability engineers can seamlessly shift the type of mockup for the test from the virtual one to the physical one.

### 3 Overview of the Testing and Assessing Environment

Figure 2 shows the proposed testing and assessing environment. The environment mainly consists of the mockup viewer/controller part and the UI simulator/tester part. These two parts can be communicated by socket-communication via network. The mockup controller part includes the 2D UI operable digital mockups, 3D UI operable digital mockup, and the physical mockup controller. A physical functional mockup can be connected to this environment via the controller. On the other hand, the UI simulator/tester part includes statechart executor, test task coordinator, event logger and test result analyzer.

In the working situation, a tester operates the digital or physical mockup, then an event published from the mockup is sent to the statechart executor. And an action published from the statechart executor is then returned to the mockup to make the indicated status of UI changed. The statechart executor carries out a transition of UI state and generates an action based on the predefined statechart specification.

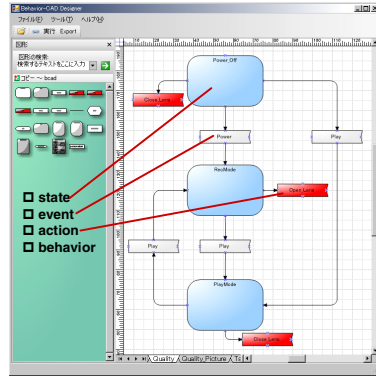
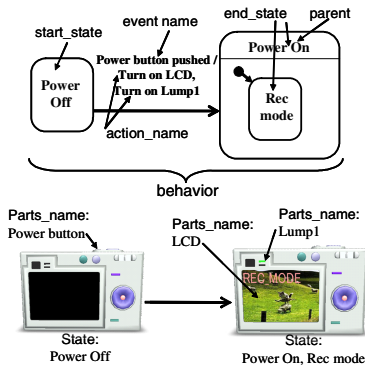


Fig. 3. UI behavior model based on Statechart Fig. 4. MS/Visio-based Statechart editor

## 4 User Interface Simulator / Tester

### 4.1 Statechart Executer

In the statechart executer, UI behavior is processed in response to incoming events such as mouse click issued from a test subject either on the 2D digital, the 3D digital or the physical mock-up. UI behavior data model uniformly represents event-driven behaviors on the appliance's side which describe how status on the output devices (CRT, LCD, lumps etc.) of the appliance changes as a result of the event from the input devices (buttons, dials, etc.). This event-driven behavior is modeled by the Statechart [12].

The UI behavior model mainly consist of “state”, “event”, “behavior” and “action” as shown in Figure 3. A state represents an identical physical status of the UI. An event represents a user's action inputted into the UI such as “a power button is pushed”, and brings on a transition from one state to the other state. An action is expresses dynamic change occurred on the appliance. A change of the LCD property such as “turn on a LCD from dark to blue” is an example of the action. A behavior expresses a transition between two states, and is defined as a combination of two states before and after the transition, an event causing the transition and a sequence of actions of the appliance caused by the transition.

As shown in Figure 4, the statechart specification of the UI behavior can be schematically created and edited by a commercial visual editor (MS/Visio), and the statechart executer can directly read the document-object-model of Visio. Interaction designers can easily input and edit the UI behavior specification in graphical way.

### 4.2 Test Task Coordinator

In a user test, subjects are asked to complete a set of “task” by operating the UI, and his/her actual UI operations for each task are logged to evaluate the usability. In the standard [13], a task is defined as “the activities undertaken to achieve a goal”. As shown in Figure 5, in our test task coordinator, a task is modeled by a start state, goal

state and a list of task routes. And a task route consists of a list of checkpoints. A checkpoint is a certain state where a correct sequence of UI operations must pass. A start state and a goal state both refer to the “state” defined in the Statechart.

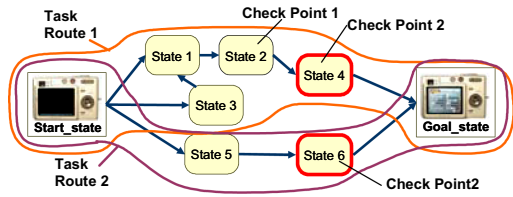


Fig. 5. Task, Checkpoint and Route

Generally there are often multiple correct sequences of UI operations to complete a certain task and to achieve a goal. In our task model, multiple correct task routes are allowed in one task by defining the different set of checkpoints shown in Figure 5. If actual number of input events from the continuous checkpoints stays within a predetermined number of time, or if actual elapsed time of the operation between these checkpoints stays within a specified time limit, the operations are judged to be correct. In this way, examiners of user-test can define alternatives of task routes, and can flexibly adjust a range of correct UI operations for the task when evaluating the number of error operations and task completion in the test results analysis.

After defining tasks, the test task coordinator indicates a predefined goal of each task in the form of an imperative sentence when a new test starts. “Switch shooting mode” is a typical example of the goal. The goal is indicated on another window of the screen just above the 3D digital mock-up as shown in Figure 8.

### 4.3 Test Results Analyzer

The test results analyzer compares the event log data from the subject’s operation with the test task data and UI behavior specification, and outputs measures of the usability assessments. We adopted the following four measures of usability assessment based on three basic notions of usability (effectiveness, efficiency and satisfaction) defined in ISO 9241-11 [13]; 1) number of events inputted in each task and in each subject, 2) elapsed time in each task and in each subject, 3) personal task achievement ratio, and 4) scores for SUS questionnaire.

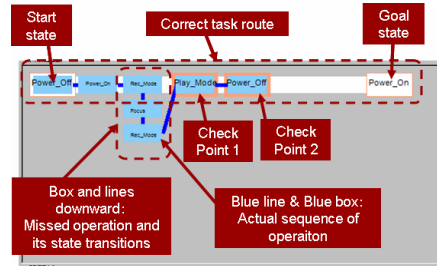


Fig. 6. Operational log analysis chart

An actual sequence of state-transitions operated by a subject is automatically compared with the one of correct task routes defined in the test task data, and the result of the comparison is graphically displayed in the form of “operational log analysis chart” on the lower part of the screen. Figure 12 shows the notation of this operational log analysis chart. Each rectangle shows a state, and a line between two rectangles does a

transition between states. A left-most rectangle in the chart indicates a start state, and a right-most rectangle does a goal state. The upper most horizontal white straight line indicates transitions (behaviors) on a correct task route, and every rectangle with orange edges on this line except both ends corresponds to each checkpoint. While the blue rectangles and blue lines indicate actual operation sequence of the subject. If a subject does UI operations whose elapsed time or number of events between two neighboring check points exceeds the predefined bounds, the tool judges that a subject did a wrong operation on the UI, and draws additional blue rectangles and blue lines in downward direction corresponding to these wrong operations. Therefore, as a depth of the chart becomes larger, the examiner of the user test can easily recognize that the subject did more missed operation in this task.

## 5 Mockup Viewer/Controller

### 5.1 2D UI Operable Digital Mockup

In case of simulating UI behavior of appliances where all of the UI elements such as LCD, buttons and lamps are placed in one surface of the appliances, 2D UI operable digital mockups is sufficient. Therefore, we build a Visual Basic application for 2D UI operable digital mockup as shown in Figure 7. In this mockup, images of menus and icons displayed on LCD and ones of buttons are painted using 2D drawing or painting tool by interaction designers in advance. Then a 2D spatial arrangement of these images and change of the display status of these images are defined in an XML document according to the format of our extended vision of UsiXML[14] which is one of the Model Driven Architecture methodologies of developing the UI application.

When executing UI simulation and user test, the incoming events such as mouse click issued from a test subject on the mockup are sent to the statechart executer via inter-process communication, and an action published from the statechart executer is then returned to the mockup to make the indicated UI status of the mockup.

### 5.2 3D UI Operable Digital Mockup

In case of simulating UI behavior where UI elements are placed in multiple surfaces of the appliances, 2D UI operable digital mockups is insufficient and a digital mockup with 3D housing shape data is needed. To build this 3D UI operable digital mock-up,

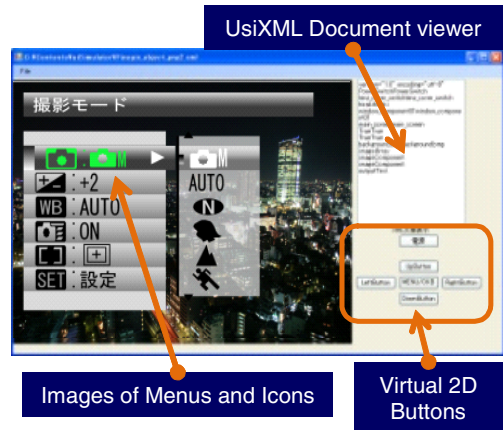


Fig. 7. 2D UI operable digital mockup

we used a commercial Web3D programming tool (Virtools) and added two functions of UI simulations to the tool.

The 3D CAD data is imported to the tool in the format of 3D-XML [15]. The 3D-XML format includes the assembly configuration information which describes relationships among an assembly and parts. 3D-XML format includes the assembly configuration information which describes relationships among an assembly and parts. In the 3D CAD data of the housing, a 3D object which becomes a target of an event or an action has to be modeled as a single part in the assembly. A button, a switch, a LCD screen and a LED are typical examples of such objects. A usability engineer interactively ties each of events or actions in the behavior data to the name of a part selected from the list. Consequently this operation makes logical links between the part names of 3D CAD data and events or actions in UI behavior data.

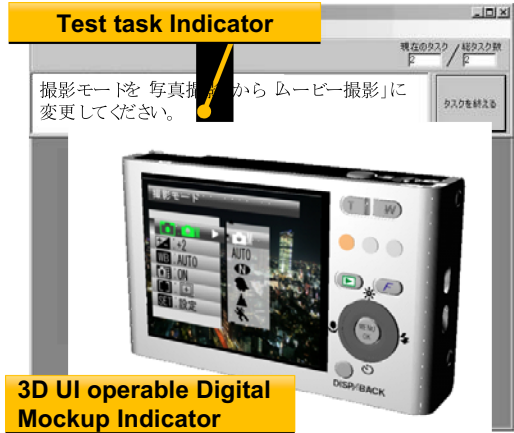


Fig. 8. 3D UI operable digital mockup

The mechanism of executing UI simulation and user test on this 3D UI operable digital mock-up is as same as the one of 2D digital mockup as shown in Figure 8. The inter-process communication function was developed and built in the tool. When executing the 3D mockup, the 2D mockup described in 5.1 is also executed with this 3D mockup simultaneously. An image data displayed on the LCD part on this 3D mockup is generated by the 2D mockup in real time, and is pasted on the 3D mockup via a scratch file by the image sharing function which we developed.

### 5.3 Physical Mockup and Its Controller

In the 2D or 3D digital mockups described above, the user has to complete his/her operations on the UI completely on a computer screen, and this sometimes makes the test results inaccurate. To solve the problem, we also developed the physical mockup controller which enables us communication between a variety of physical input and output devices and the statechart executer. Physical dials, buttons, lumps, LCDs and LEDs placed on the physical housing model are connected to this controller, and can be directly operated by the test subjects.

Moreover, to reduce the time and cost of fabricating the physical mockup, we can connect the RFID embedded physical mockup [16] to the controller. Figure 9 shows an example of this RFID embedded physical mockup of a digital camera. In this RFID embedded physical mockup, 2mm x 2mm small RFID tags are stuck at the position of buttons on the mock-up, and a subject puts a glove-type wearable RFID R/W on his/her

arm where small antennas of the R/W are attached to each fingertip. When the subject is operating the mock-up, physical contact between the RFID tag in the button and his/her fingertip is detected by the wearable RFID R/W. The tag's ID code is sent to the physical mockup controller.

A button with a RFID tag is very small and is easily attached to and detached from the mock-up surface, because of its wireless and powerless communication functions. The position of buttons or switches on the mock-up can easily be repositioned. Therefore redesign-and-test activities can be repeated in a short cycle.

By integrating these physical mockups with the 2D and 3D digital mockups, the usability engineer can seamlessly shift the type of mockup for the test from the virtual one to the physical one, so that more accurate results of operation can be obtained.



Fig. 9. RFID embedded Physical Mockup

## 6 Examples of Usability Testing and Assessment

User tests using the UI operable 3D digital mock-up (“3D-DMU”) and RFID embedded physical mockup (“PMU”) were done to verify an effectiveness of the environment. A 3D-DMU and PMU of a digital camera (Fuji-Fine Pix Z1) on the market shown in Figure 8 and 9 were built for the test. These two types of mock-ups (3D-DMU and PMU) are operated by two subject groups each of which consists of 7 subjects who did not have experiences of operating this camera. The test results of 3D-DMU and PMU were compared and analyzed.

We set two test tasks; turning on the camera (task-1), and setting self-timer's duration to 2 second (task-2). To complete the task-1, the subject has to open the lens cover on the front side of the housing. While in the task-2, the subject has to push several buttons on the back side.

Figure 10 shows two operational log analysis charts for the task-2 when using the 3D-DMU (Figure 10-(a)) and the PMU (Figure 10-(b)). A wider line on the chart indicates more subjects passed over the route of operations. It was found that the sequences of operations including wrong operations were very similar to each other in the 3D-DMU and in the PMU, and that many subjects stepped to a wrong way at a state of “Rec\_Mode”. By observing this state on the 3D-DMU, it was found that many subjects did not notice that the self-time setting was allocated to the downward cursor button, and found that many of them pushed the center button instead of the downward one. This result suggested that many subjects could not recall the



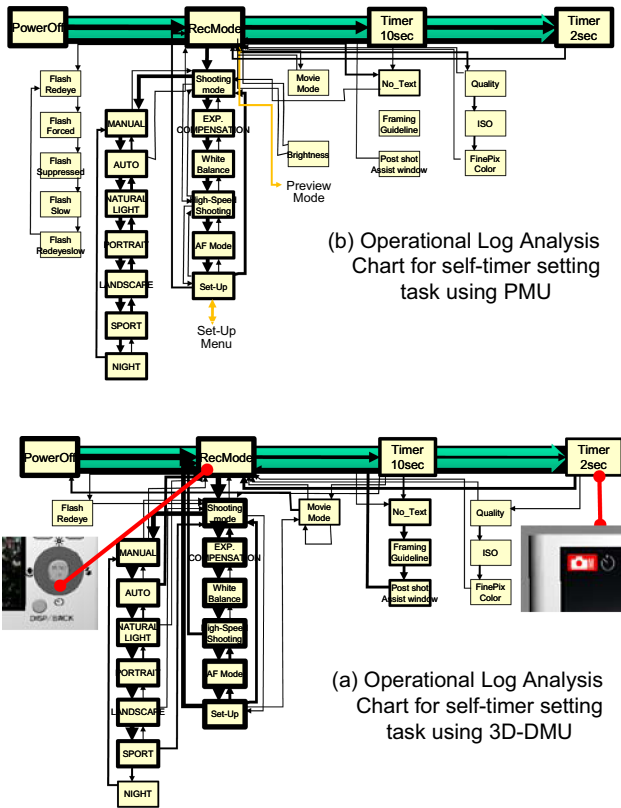


Fig. 10. A difference of operational log analysis chart using 3D-DMU and PMU

“self-timer” from the small icon indicated on the housing surface just under the downward cursor button as shown in Figure 10-(a). The shape and size of this self-timer icon should be redesigned.

## 7 Conclusions

An integrated software and hardware environment was proposed for prototyping and assessing usability of information appliances. The users could operate a user-interface either using a 2D digital mockup, 3D digital mockup or a physical-mockup of the appliance under the test. This function enabled engineers to test and evaluate the usability of the appliances both from the cognitive aspect and from the physical aspects. The results of the user test using a 3D digital mock-up in comparison with those of a physical mockup of the digital camera confirmed an effectiveness of our proposed environment.

**Acknowledgements.** This work was financially supported by a grant-in-aid of Intelligent Cluster Project (Sapporo IT Carrozzeria) funded by MEXT.

## References

1. Protobuilder, [http://www.gaio.co.jp/product/dev\\_tools/pdt\\_protobuilder.html](http://www.gaio.co.jp/product/dev_tools/pdt_protobuilder.html)
2. Rapid Plus, [http://www.e-sim.com/products/rapid\\_doc/index-h.htm](http://www.e-sim.com/products/rapid_doc/index-h.htm)
3. Avrahami, D., et al.: Forming Interactivity: A Tool for Rapid Prototyping of Physical Interactive Products. In: Proceedings of the ACM Symposium on Designing Interactive Systems, pp. 141–146 (2002)
4. Lee, K.P.: User-Participatory Testing Using Web, In: Proceedings of HCII 2003 (2003)
5. Kuutti, K., et al.: Virtual prototypes in usability testing, In: Proceedings of the 34th Hawaii International Conference on System Sciences 2001, vol. 5 (2001)
6. Bruno, F., et al.: A new approach to participatory design: usability tests in virtual environment, In: Proceedings of Virtual Concept 2005 (2005)
7. Kanai, S., et al.: Digital Usability Assessment for Information Appliances using User-Interface operable 3D Digital Mock-up. In: Proceedings of Virtual Concept 2006, HUCEID-P235, vol. 2, Springer, Heidelberg (2006)
8. Verlindern, J., et al.: Qualitative comparison of Virtual and Augmented Prototyping of Handheld Products, In: Proceedings of International Design Conferences, pp. 533–538 (2004)
9. Nam, T.-J.: Sketch-Based Rapid Prototyping Platform for Hardware-Software Integrated Interactive Products, In: Proceedings of CHI'05 2005, pp. 1689–1692 (2005)
10. Aoyama, H.: Development of system using mixed reality technology for evaluating designability and operability of product. In: Proceedings of Virtual Concept 2006, WoIM-P188, vol. 2, Springer, Heidelberg (2006)
11. Hartmann, B., et al.: d.tools: Visually Prototyping Physical UIs through Statecharts, In: Proceedings of UIST 2005 (2005)
12. Horrocks, I.: Constructing the User interface with Statecharts. Addison-Wesley, London, UK (1998)
13. ISO9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability (1998)
14. Vanderdonckt, J., et al.: UsiXML: a User Interface Description Language for Specifying Multimodal User Interfaces, In: Proceedings of W3C WMI'2004 (2004)
15. 3D-XML: <http://www.3ds.com/3dxml>
16. Horiuchi, S., et al.: Low-cost and rapid prototyping of UI-embedded mock-ups using RFID and its application to usability testing, In: Proceedings of HCII 2005 (2005)