

# 3D digital prototyping and usability enhancement of information appliances based on UsiXML

Satoshi Kanai · Taiga Higuchi · Yukiaki Kikuta

Received: 15 February 2009 / Accepted: 23 June 2009  
© Springer-Verlag 2009

**Abstract** Recently, usability assessment of user interfaces (UIs) in the early design stage has become indispensable for the human-centered design of information appliances. Several digital prototyping tools of the UI software, in which the UI behavior can be simulated, have been developed so far. However, some problems can arise when using them for the usability assessment, such as: a lack of simulation fidelity, time-consuming modeling work, and lack of systematic usability assessment functions. To solve these problems, we developed a tool for virtual 3D prototyping and usability assessment of information appliances. First, we extended the specification of UsiXML (which is the XML-compliant UI description language) so that the static structure both of the logical elements displayed on the UI and of the physical elements placed on the housings can be modeled. Secondly, we developed 3D prototyping and simulation functions so that the extended UsiXML is combined with the 3D CAD models of the housings and where the interaction can be simulated based on the dynamic-behavior model of the UsiXML. Finally, we developed the automated user test and usability assessment functions which are connected to the models of the extended UsiXML. An example of the efficient usability assessment and the UI redesign using a 3D digital prototype of a digital camera is shown.

**Keywords** Digital prototyping · Usability · 3D CAD · Information appliance · User interface

---

S. Kanai (✉) · T. Higuchi  
Graduate School of Information Science and Technology,  
Hokkaido University, Kita 14, Nishi 9, Kita-ku, Sapporo, Japan  
e-mail: kanai@ssi.ist.hokudai.ac.jp

Y. Kikuta  
C's Lab Co., Ltd., Maruito-building, Kita 2,  
Nishi 1-1, Chu-o-ku, Sapporo, Japan

## 1 Introduction

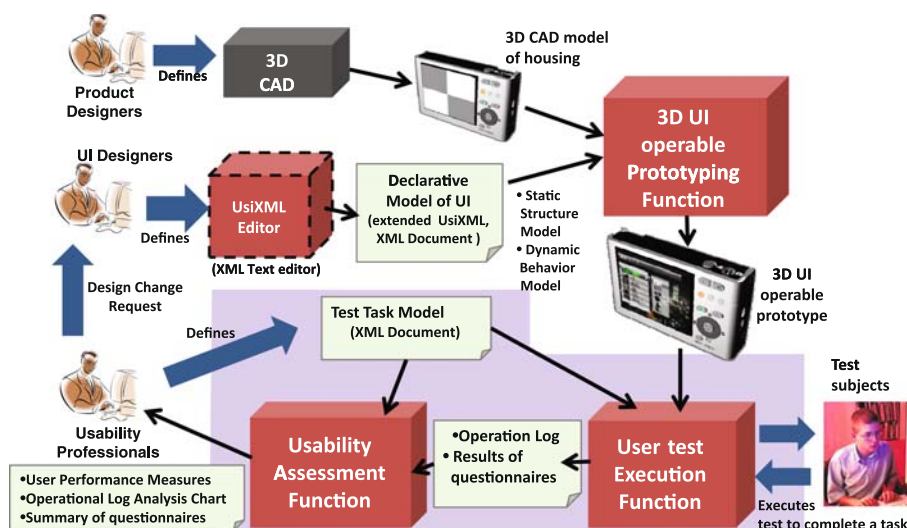
Recently information appliances have become readily available for anyone, and their functions are growing more and more. This brings about a drastic increase in the development process of user interface (UI) software. At the same time, the manufactures of the appliances have to enhance their market competitiveness. As one of the keys for differentiating their products from those of their competitors, the manufactures are aware of human centered design (HCD).

The manufactures are placing a premium on increasing efficiency and consciousness of “usability” in the UI software development of their appliances. In the standard [1], the “usability” 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”.

There are several methods of evaluating usability [1]. Among the methods, the “user test” is recognized as the most objective and effective one where end users are directly involved in the evaluation. In the user-test, designers make end users operate a “prototype” of the appliance, observe the user’s operational situation, and closely investigate ergonomic issues of the UI design.

However in the UI software development for the prototype, its specifications are still described by written documents, and the software is implemented based on the documents. This makes the prototype implementation process inefficient if a redesign of the UI is needed after the user test. Moreover, the “physical” prototypes of the appliances are mostly used in the user-tests. But fabricating the UI-operable physical prototypes is expensive, and they are available only in late stages of the design. If problems of the UI design appear at this time, it is often too late for changes within their development schedule.

**Fig. 1** An overview of the developed prototyping, user-test and usability assessment tools



To solve the problem, in this paper, we propose an advanced tool for UI prototyping where a UI-operable “3D digital” prototype can be fabricated in a less-expensive way based on the model-based UI specification, and where the user test and usability evaluation can be performed in more systematic and efficient way than in the physical prototype.

Figure 1 shows the overview of this research. The objectives of the paper are summarized as follows:

- The model-based specification of UsiXML, which is one of the XML-compliant UI description languages, is extended to enable the UI designer to specify the declarative model not only of the logical UI elements displayed on the screen but of the physical UI elements such as buttons and dials placed on appliance’s housings.
- 3D UI prototyping and simulation functions are developed where the extended UsiXML is combined with the 3D CAD models of the housings and where the UI interaction are simulated based on the declarative model of the UI behavior described by the UsiXML.
- The automated usability assessment functions are developed in such a way that they are tightly connected to the declarative model of the UI and to the simulation functions.
- An example of the usability assessment and the UI redesign using the 3D digital prototype of a digital camera using our tool is shown to indicate the effectiveness and reliability of our proposed tool.

The rest of the paper is organized as follows. Related works and the features of our tool are shown in Sect. 2. The outline of the UsiXML and our extensions are described in Sect. 3. The 3D UI prototyping and simulation functions of the tool based on the extended UsiXML specification

are described in Sect. 4. The user test execution function is described in Sect. 5. The automated usability assessment functions are described in Sect. 6. And a case study of the usability assessment of the UI is indicated in Sect. 7. Several challenges left in the study are discussed in Sect. 8. Section 9 describes the summary of the research.

## 2 Related works

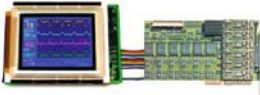


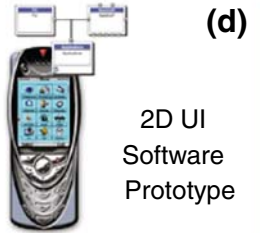

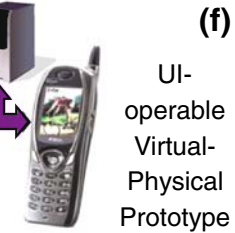
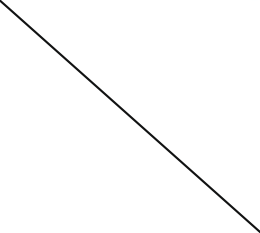


### 2.1 Classification of prototypes

Currently, different types of prototypes are used in the design of information appliances. A prototype of a particular type is used depending on the development stage, the purpose and allowable period of the assessment, the cost of fabrication, etc. Figure 2 summarizes the types of the prototypes [2,3]. The information appliance mainly consists of the housing and UI software system. Some types are being used in practice, but others are still in study phase. Therefore, the prototypes differ according to the type of the housing and the UI software shown in Fig. 2.

The hardware prototype (Fig. 2a) and “hardware in the loop” (HIL) prototype are used to study the feasibility of the hardware and software implementations of the appliances. However the fabrication of these prototypes is relatively expensive due to the development of embedded software, and UI operable prototypes are only available in the late stage of the design. The external appearance of the UI is not implemented in them. Therefore the prototypes are not usually used in the usability evaluation of the appliances.

In the physical functional prototype (Fig. 2c), the housing is physically fabricated and the UI software is implemented in the real hardware placed in the housing. This prototype gives the best fidelity in the usability assessment if it is used

**Fig. 2** Prototypes for developing information appliances

		Housing		
		None	Virtual	Physical
UI Software	Implemented	 <p><b>(a)</b> Hardware Prototype</p>	 <p><b>(b)</b> HIL Prototype</p>	 <p><b>(c)</b> Physical Functional Prototype</p>
	Virtual	 <p><b>(d)</b> 2D UI Software Prototype</p>	 <p><b>(e)</b> UI operable 3D Digital Prototype</p>	 <p><b>(f)</b> UI-operable Virtual-Physical Prototype</p>
	Physical		 <p><b>(g)</b> 3D CAD Model (3D digital mockup)</p>	 <p><b>(h)</b> Physical Housing Mockup</p>

in user-test. However the fabrication cost of this prototype is the most expensive among the ones shown in Fig. 2.

On the other hand, the 3D digital mock-up (Fig. 2g) and the physical housing mock-up (Fig. 2h) realize the 3D shape of the housing and the placement of the physical UI elements such as buttons. But they are not UI operable, and are useless from the cognitive aspect of the usability assessment.

The 2D UI software prototypes shown in Fig. 2d are being used in the usability assessment in the manufactures. Commercial prototyping tools with UI simulation functions are available such as Maestro and RapidPLUS, and are used in the development of appliances such as mobile phones. The tools for the conceptual design phase where functioning UI prototypes can be modeled from rough sketches of the UI were also proposed in [4, 5]. Moreover, there are tools which have usability assessments functions [6, 7] as well.

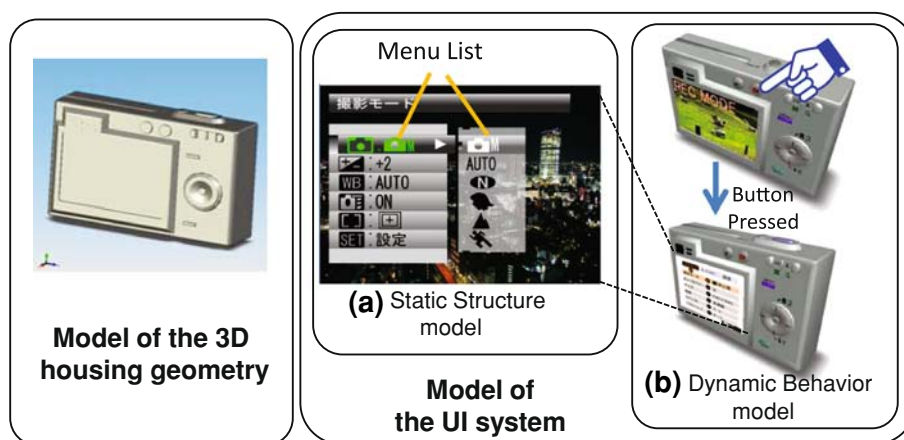
However, these tools were aimed to prototype the UIs of the appliances where their UI elements are placed 2-dimensionally such as mobile phones. Therefore, UI simulation functions are only limited to the 2-dimensional one. In case of the user test for portable information appliances whose UI elements (buttons, dials, LCDs, etc.) are placed in 3-dimensions, such as digital cameras or handy video cams, UI operating situation in the 2D prototype is far from that

of the real appliance, and the feeling of interaction is not satisfactory for the subjects of the user test.

On the other hand, UI operable 3D digital prototypes have been developed as shown in Fig. 2e. They consist of the virtual 3D housing model (3D CAD model) and the UI simulation function which can work on the 3D housing model. In spite of lacking a physical housing, the UI simulation function working on the 3D housing enhances the reality of interaction in the user test. Most of the 3D digital prototypes are built and run using commercial Web3D authoring tools and their players [8–10].

To make UI operable 3D digital prototypes suitable for the user test, Kuutti [11] added a logging function of user events to a VRML viewer, and applied the tool to a user test of an advanced mobile phone. They suggested that 3D digital prototypes could enhance authenticity, interaction reality and reliability of design evaluation compared with the 2D UI software prototype. They also suggested that the digital prototypes can be delivered around the world simultaneously and can be used for eliciting feedback from a very large number of users. Meanwhile, they pointed out that the actual size, real shape, feel of material and touch, weight and its relation to the human body cannot be estimated in the 3D digital prototypes. Physical ergonomics and the impact

**Fig. 3** Modeling constituents of the UI operable 3D digital prototype



of the user's movements in various steps of the use sequence also cannot be evaluated accurately. They suggested that a 3D prototype is suitable for testing and evaluating the logics of the UI behavior. Unfortunately, test execution and usability evaluation after the test were not automated in their system.

On the other hand, Bruno [12] developed a UI operable 3D digital prototype system where the user puts on a head mounted display to manipulate the UI of 3D electric cooker model using the Data-Glove. Their results showed a significant analogy of user performance measures between a 3D digital prototype and a real product. Most of the measures concern the types and the number of mistakes committed while carrying out the tasks. Unfortunately their system needs the expensive VR device, and the UI system is the one which can be modeled in 2-dimension. The dedicated functions of modeling the UI behavior and evaluating the usability were not fully discussed in the paper.

The UI operable virtual-physical prototype which corresponds to Fig. 2f has a physical housing prototype and a UI simulation function. The UI simulation function usually runs in an external host computer. The subject can directly grasp a physical housing and operate the UI elements on it. This prototype allows for relatively reliable test results on the user performance. However, these prototypes need an extra cost of fabricating housing and of hardware connections between the housing and the host computer. Mixed reality or virtual reality techniques such as image-projection of the UI screen have been used to eliminate the hardware connections between the displays and host computers [13–16]. However, expensive and specific devices such as head-mounted displays, magnetic position and orientation trackers, etc. are needed to realize the prototypes. Moreover, in these studies, how to model the UI system and evaluate the usability was not discussed as main objectives. Correlation of user performance measures between the 3D digital prototype and a real product were not fully investigated.

## 2.2 Issues of the UI operable 3D digital prototypes

To keep the fabrication cost of a UI-operable prototype relatively low and to assure reliability of the user test results to some extent in the early design stage, UI operable 3D digital prototypes are more likely to be suitable for testing and evaluating the logics of the UI, and for clarifying weaknesses and what needs improvement in the UI design.

As shown in Fig. 3, the modeling of UI operable 3D digital prototypes consists of two parts; the model of the UI system and of the 3D housing geometry. Moreover the model of the UI system is divided into the static structure model and the dynamic behavior model. The static structure model of UI describes graphical properties of individual 2D components displayed on the UI screen such as menu-list, button, list-box, slider and image component, and also describes containers for the component layout such as window and tabbed dialog-box. While the dynamic behavior model of UI describes how graphical properties of the 2D components of the UI change in interaction and enables us to simulate the state change of the UI part in the appliance.

Conventional UI operable 3D digital prototypes were built and run using the Web3D authoring tools and their players [8–10]. However, the following technical issues remain when we use the Web3D as the UI operable 3D digital prototype for user test and usability assessment:

1. *Lack of the static structure model of the UI:* The static structure of the 2D components displayed on a UI screen such as menu list or icon placements cannot be directly modeled in the Web3D formats. So a huge number of digital image files representing snapshots of the UI screen must be built using the 2D drawing or painting tools before the UI simulation and the user test. This preparatory work makes the simulation turn-around very slow.
2. *Lack of the dynamic behavior model of the UI:* The Web3D formats usually provide script-based control

function which enables 3D kinematic animations, change of the graphical properties of 3D objects and texture mapping, etc. But the function cannot simulate the dynamic behaviors of the 2D components displayed inside the UI screen. The We3D formats do not also provide any declarative dynamic behavior model of the UI system which is based on state-based or event-based formalisms. These formalisms of the UI fit to the screen transition diagrams in early UI design stage [17, 18], and are indispensable to the specification. The lack of the declarative dynamic behavior model forces UI designers to code the behavior using programming language. But the designers are not necessarily programming professionals, and the task makes the cost of UI simulation and user testing expensive.

3. *Lack of the functions for user test and usability assessment:* The Web3D formats do not provide any functions of user test execution and usability assessment based on the operational logs. Doing these works manually on the digital prototype makes the usability assessment time-consuming and the assessment results inaccurate.

To solve these issues, the dedicated functions of modeling the static structure of the UI-screens, of modeling the event-based or state-based dynamic behavior of the interaction, and of supporting the computer-aided test execution and the usability assessment must be added to the traditional Web3D authoring tools and players. Therefore, in this paper, we propose a computer-aided tool which has these dedicated functions worked on the 3D digital prototypes. The functions are described in the following sections.

### 3 UsiXML and its extensions

#### 3.1 UsiXML [19, 22]

Several XML-based user interface mark-up languages have been recently proposed to make UI prototyping of PC applications reduced and structured: UIML [20], XUL [21], UsiXML [19, 22] and XAML [23]. Each of them specifies models for defining the UI and allows us to describe the UI model in declarative terms. They enable the UI designers or UI programmers to specify what elements are to be shown in the UI and how should they behave in XML documents. This concept becomes an advantage for defining 3D digital prototypes of handy information appliances from the following standpoints:

1. The static structure of the 2D component objects displayed on the UI screen is explicitly and declaratively modeled and described by the XML document. The

snapshot of the UI screen can be automatically drawn in the simulation based on the static structure model if we realize the real-time rendering function for the model. It can eliminate the preparatory work of the UI simulation, and makes its turn-around efficient.

2. The dynamic behavior of the UI interaction has to be described by script or ordinary programming language in most the of UI mark-up languages (UIML, XUL and XAML). However, in the UsiXML, the behavior can also be explicitly described as an event-based model. The model can eliminate the coding of UI dynamic behavior simulation if an execution function of the behavior model in the simulator of the 3D digital prototype is realized.
3. The user test and the usability assessment can be automated if the static structure and the dynamic behavior models of the 3D digital prototype are reused for analyzing the property of the subject's operations in the usability assessment functions. It can make the cycle of prototyping-testing-redesigning very efficient.

Therefore, we introduced UsiXML to our study, because it can describe the dynamic behavior model of the UI in a declarative way and model the UI at a different level of abstraction.

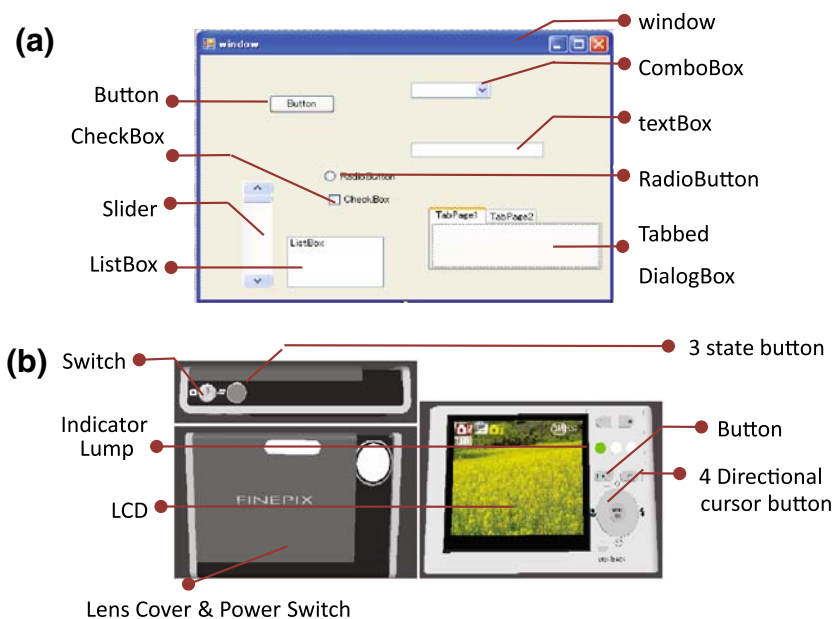
UsiXML was originally proposed by Vanderdonck et al. [19]. It aims at expressing a UI built with various modalities of interaction working independently. The UI development methodology of UsiXML is based on the concept of model driven architecture (MDA). UsiXML specifies the UI development process as the transformations among four models; *Task&Concepts*, *Abstract UI (AUI)*, *Concrete UI (CUI)* and *Final UI (FUI)*. The *Task&Concepts* model specifies the tasks to be executed by the users and the information to be input in the tasks. AUI specifies the abstract structure of the UI which performs the tasks independent of the modalities such as graphical, vocal, audio, etc. CUI mainly specifies the appearances of the UI independent of the platforms. FUI is an implementation of the UI running on a specific platform such as applications of html, MFC, JavaSwing, etc. Plenty of experimental automated tools were also developed which support the UsiXML development methodologies [22].

#### 3.2 Extensions of the UsiXML

##### 3.2.1 Issues of the UsiXML from the aspect of 3D digital prototypes

The concept and specification of UsiXML is advanced in UI prototyping, but it has still the following issues when we directly use it for developing UI operable 3D digital prototypes and for the usability assessment:

**Fig. 4** Types of UI component objects modeled in the static structure model. **a** Static structure model of WIMP-based UI objects; **b** Static structure model of physical UI objects



1. As shown in Fig. 4a, the CUI of UsiXML specifies the static structure model of 2D UI component objects displayed on the UI screen, but the current CUI only specifies the static structure model for WIMP (Windows-Icons-Menus-Pointers)-type GUI. In UsiXML, there are no specifications for the physical UI elements such as buttons, dials, lumps and dials placed on the information appliance's housing shown in Fig. 4b, which are essential for modeling the UI operable 3D digital prototypes.
2. Many experimental automated tools have been developed for UsiXML. However, there is no 3D UI prototyping and simulation tool available to UsiXML at present. The event-based dynamic behavior model is specified in the CUI, but it has not been reported yet how the dynamic behavior of the UI is described concretely, nor how the model of the CUI should be implemented on a particular Web3D format.
3. UsiXML has been originally developed for UI prototyping, but at present there is no specification and no supporting tool concerning user testing and usability assessments which utilize the UI prototype. Therefore, we cannot incorporate the functions of user testing and usability assessment into UI prototyping based on UsiXML.

### 3.2.2 Extensions of the CUI model of UsiXML

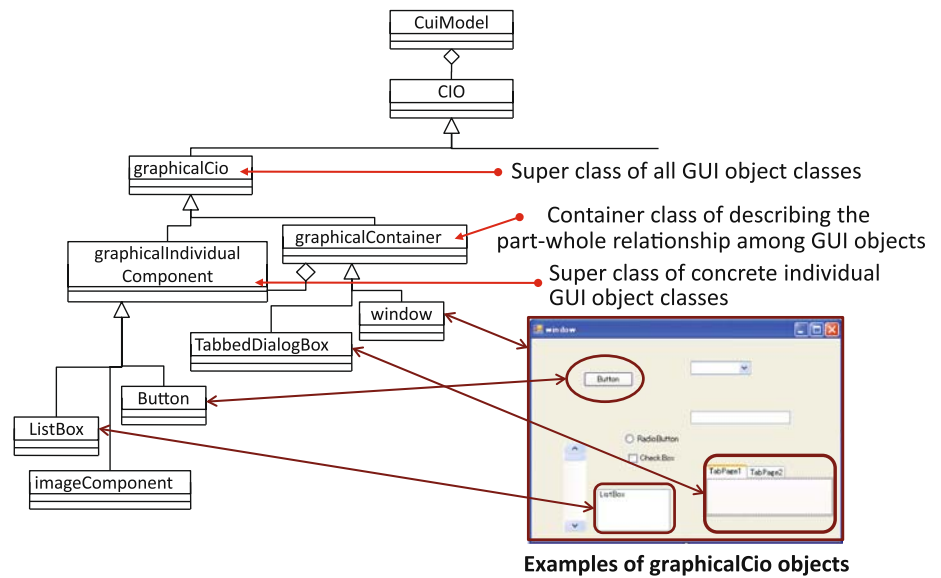
The current specifications of the CUI in UsiXML mainly consist of the static structure model of the objects displayed on the UI screen and the dynamic behavior model of the interactions of the UI. The static structure model further consists

of the UI object model and the object container model. The UI object model expresses the individual GUI component object displayed on the UI screen such as buttons, list-boxes, image components, etc., while the object container model does the whole-part relationships among the multiple GUI objects such as a window and a tabbed dialog box. On the other hand, the dynamic behavior model consists of the event-based rewriting rules of the UI screen in interaction and of the procedures and variables to refer to the internal data of the device.

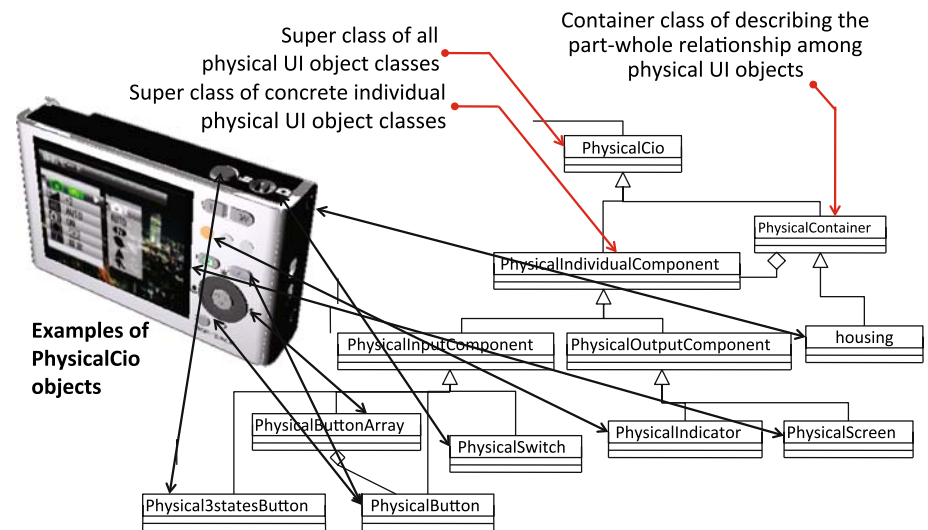
In this research, we extend this static-structure-model part of UsiXML so as to fit it to the UI operable 3D digital prototyping. Figure 5 indicates the UML class diagram and its example which expresses a part of the original CUI model structure in the UsiXML. In the structure, *GScraphicalCio* is the super class of all GUI object classes in the model, and *GraphicalContainer* is the container class for the GUI objects. The concrete classes of the GUI objects and the object containers of UI are defined as a subclass of these two classes.

Figure 6 indicates the class diagram and its example of our extension of the CUI model. We extended the class structure of the CUI model to express the physical UI objects such as physical buttons, dials and lumps placed on the 3D geometric model of the appliance' housing. First, we added a new class *PhysicalCio* to the same class hierarchy level as one of the *GraphicalCio* class. Then we further created two new classes of *PhysicalIndividualComponent* and *PhysicalContainer* as subclasses of the *GraphicalCio*. The *PhysicalIndividualComponent* class expresses the one for modeling each the physical UI object, and the *PhysicalContainer* class does the physical housing of the appliances which play a role of the virtual container in aggregating the physical UI

**Fig. 5** UML class diagram and its example of the original CUI model structure in UsiXML



**Fig. 6** UML class diagram and its example of our extended CUI model structure in UsiXML



objects. Moreover, as the subclasses of *PhysicalIndividualComponent*, we added a *PhysicalButton* class and *PhysicalScreen* class to the subclasses of *PhysicalIndividualComponent* in order to express concrete buttons and LCDs placed on the housing. Figure 6 shows the correspondence between the physical UI objects in a digital camera and the classes describing them.

### 3.2.3 Design of the XML document structure of the extended CUI model

The current version of the UsiXML does not specify the explicit XML encoding rule of the CUI model. Therefore, we specified a tag structure of the XML document of our extended CUI model independently. This tag structure is imported to the 3D UI operable prototyping functions and is used for the 3D UI simulation. Figure 7 shows an example

of the tag structure and the correspondence between the XML documents and their presentations in the UI screen image.

The structure consists of a *<CuiModel-Presentation>* tag and a *<CuiModel-Behavior>* tag. The former represents our extended static structure model of the CUI which express the objects displayed in the UI screen, while the latter does the dynamic behavior model which corresponds to the UI screen transition. And concrete CUI objects are described inside these two tags in our XML document.

To describe the UI screen transition, we set up a *<TransformationRule>* tag inside the *<CuiModel-Behavior>* tag which describes the general graph rewriting rule mechanism defined in the original UsiXML specification. As shown in Fig. 8, in the *<CuiModel-Behavior>* tag, we put the pair of a condition tag *<lhs>* and an action tag *<rhs>* together by each tag corresponding to the subclass of *PhysicalInputComponent* class. The condition tag expresses a condition where

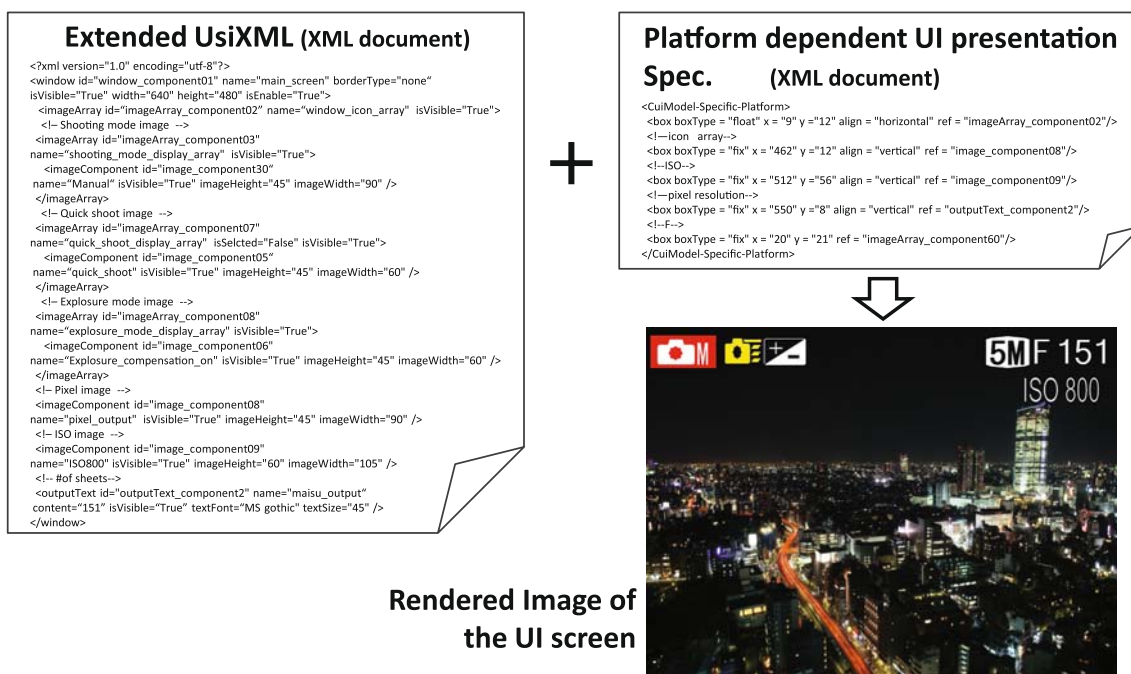


Fig. 7 An example of the tag structure in XML document of the extended UsiXML

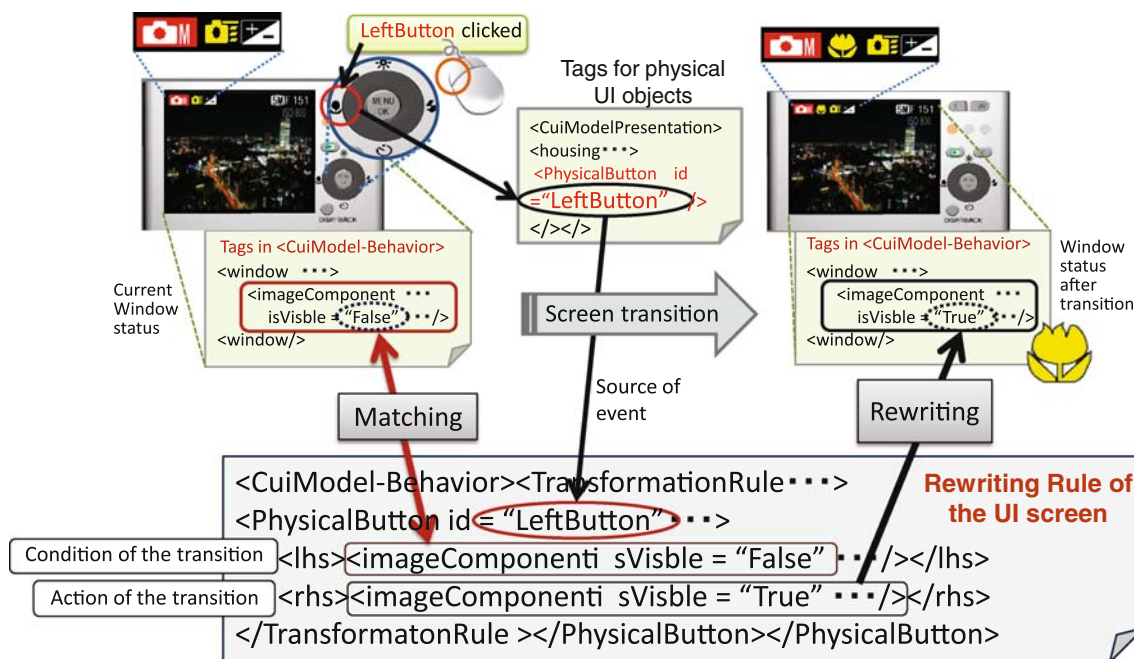


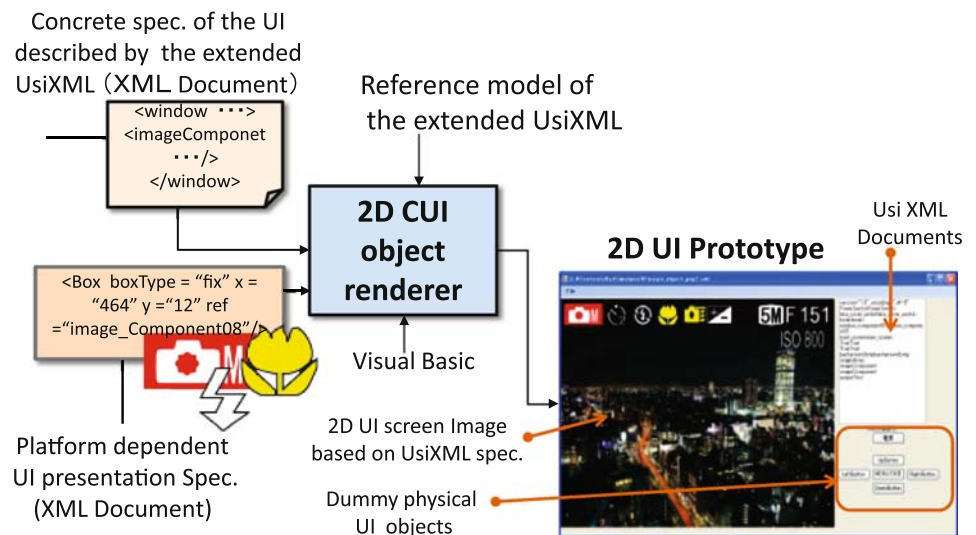
Fig. 8 UI Screen transition rule described in XML document of the extended UsiXML

the screen transition occurs because of an event coming from the physical UI objects. And the action tag expresses the state of the UI screen after the screen transition occurs.

In the user-test execution function, if an event occurs in an object belonging to any subclass of the *PhysicalInput Component* class, the function tries to find a *<TransformationRule>* tag which has the same tag id as the source of the

event from all tags inside the *<CuiModel-Behavior>* tag. And if the current tag and its attribute values in the *<CuiModel-Presentation>* tag are exactly identical to the ones written in the *<lhs>* tag in the *<TransformationRule>* tag, then these attribute values are overwritten as the one in the *<rhs>* tag. This overwriting mechanism implements the UI screen transition on the 3D UI operable prototype based on UsiXML.



**Fig. 9** Functions of 2D CUI object renderer

At present, a sophisticated visual editor which helps UI designers easily build and modify the extended UsiXML models in a graphical way is under development. Therefore, a simple text editor is used for editing the XML documents of the static structure and the dynamic behavior models.

#### 4 3D UI operable prototyping function

##### 4.1 2D CUI object renderer

We developed a 3D UI operable prototyping function where the extended UsiXML is combined with the 3D CAD models of the housings and the UI interaction were simulated based on the dynamic-behavior model of the UsiXML. The 3D UI operable prototyping function consists of the 2D CUI object renderer and the 3D UI operable simulator which is a remodeling of Web 3D player (Virtools [10]).

The 2D CUI object renderer is a VisualBasic application developed by ourselves. Figure 9 shows the function of the renderer. The renderer interprets the XML document of the extended UsiXML and accepts another XML document which defines the platform dependent UI presentation specification such as the concrete position of each GUI object and object containers on the UI screen. It renders the 2-dimensional UI screen image on the fly according to the UI screen transition rule described in the XML document. It also renders the dummy physical UI objects on the same 2D UI screen. So the renderer also enables the UI designer to do the 2D UI software prototyping when the renderer is used alone. The 2D CUI object renderer is executed during the 3D UI simulation cooperating with the Web 3D player to provide the main function for the UI simulation.

The 2D CUI object renderer enables UI designers to eliminate their preparatory works of generating a huge

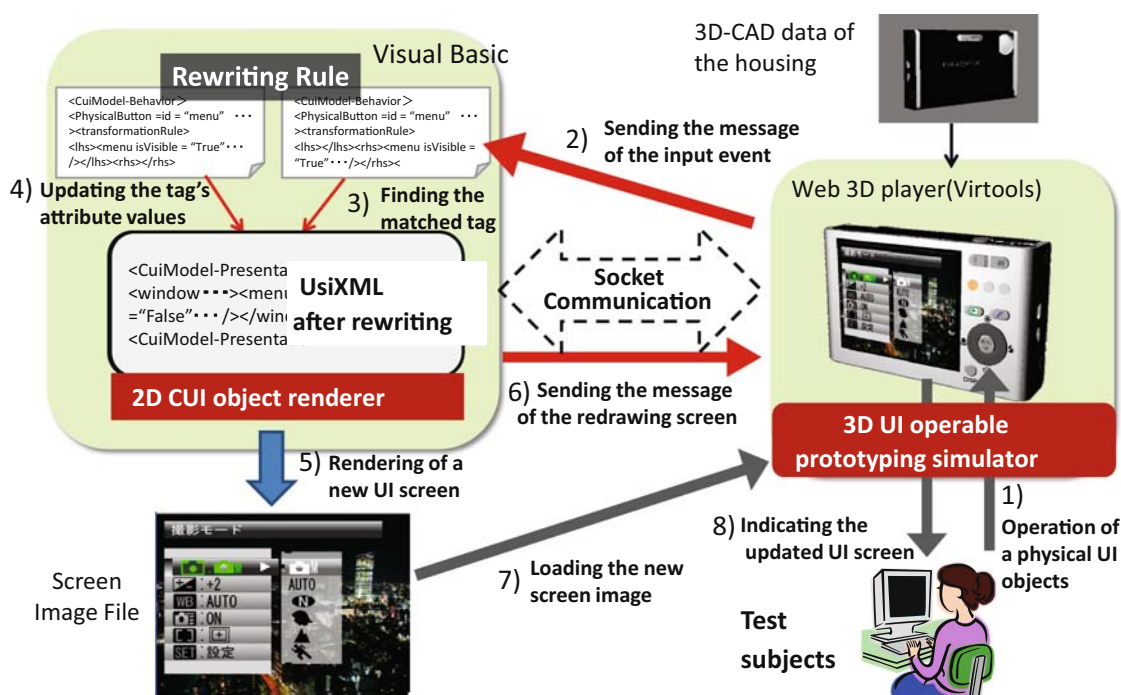
number of snapshot images of the UI screen, and makes the turn-around time of the 3D UI simulation short.

##### 4.2 3D UI operable prototype simulator

Figure 10 shows the 3D UI operable prototyping function. The function consists of the 2D CUI renderer and the 3D UI operable prototype simulator which is a commercial Web 3D player (Virtools [12]). We remodeled the players so that the Web 3D player runs with the CUI renderer simultaneously, and they exchange events via socket communication during the 3D UI simulation for the user-test.

In the preliminary step of the 3D UI simulation, 3D CAD data of the housing is imported to the Web 3D player from 3D-CAD systems (CATIA, Solidworks, etc.) in the format of 3D-XML [24]. The 3D-XML is a universal lightweight XML-based format for the sharing of 3D CAD models. The format includes the assembly and each part has its own unique part-name. In the model, a 3D object which is the source of an event or the destination of an action is modeled as a single part. A button, a switch knob, an LCD screen and an LED are typical examples of these objects.

In the Virtools player, a UI designer links an event and an action described in the *<lhs>* tag and the *<rhs>* tags in the *<TransformationRule>* tag in the dynamic behavior model of UsiXML to messages of the Virtools player. A message consists of a unique event-name, part-name and event-type. For an example, an event of “*button\_1\_pushed*” in the UsiXML model is tied to a message consisting of “*message-1*” (event-name), “*button-part-1*” (part-name) and “*on\_clicked*” (button-type). Consequently this linking operation builds all logical links between the messages in the Virtools and events or actions in dynamic behavior model of UsiXML.



**Fig. 10** Functions of 3D UI operable prototyping function

As shown in the processing sequence in Fig. 10, the 3D UI simulation is executed as the following procedure:

1. The user manipulates the 3D housing model of the appliances and operates a physical UI object such as a 3D button placed on the housing model using the mouse on the player.
2. The operation of the physical UI object is recognized as an event in the player and it sends a unique message of the event to the 2D UI object renderer via socket communication.
3. The renderer analyzes the incoming message to pick up the event, and tries to find a `<TransformationRule>` tag which has the same tag id as the source of the event from all tags in the `<CuiModel-Behavior>` tag.
4. If the current tag and its attribute values in the `<CuiModel-Presentation>` tag are identical to the ones written in the `<lhs>` tag in the `<TransformationRule>` tag, then these attribute values are updated to the new ones according to the content of the `<rhs>` tag.
5. The renderer then redraws a new UI screen image after the screen transition in a scratch file according to the updated attribute values.
6. The renderer sends another message of the screen redraw event to the player.
7. The player loads the new screen image from the updated scratch file.
8. The texture rendered on the face in the 3D housing model which corresponds to the UI screen changes to the new screen image.

By repeating the procedure, the 3D UI simulation on the 3D housing model which cooperates with the 2D UI simulation is realized in the prototyping function. The UI simulation rule is completely described in the XML document of UsiXML in declarative way, and the simulation execution is completely managed in the developed renderer.

## 5 User test execution function

### 5.1 Test task and task model

In the user test, a subject is asked to complete a set of tasks by operating the UI, and actual operations for the task are analyzed to evaluate the usability. In the standard [25], a *task* is defined as “the activities undertaken to achieve a goal”. The goals and tasks are set by usability professionals before the test.

In our tools, we designed a test task model and made a logical link between the test task model and the dynamic behavior model of the UI to automate the usability assessment. Figure 11 shows the test task model. This task model is originally developed for the “state-based” dynamic behavior model of the UI screen [2].

A *task* consists of 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 state where a correct sequence of UI operation must pass. A start state and a goal state refer to a *state* in the UI behavior model.

Generally multiple correct operations of the UI exist in order to achieve a goal. Therefore multiple task routes can

Fig. 11 Test task model

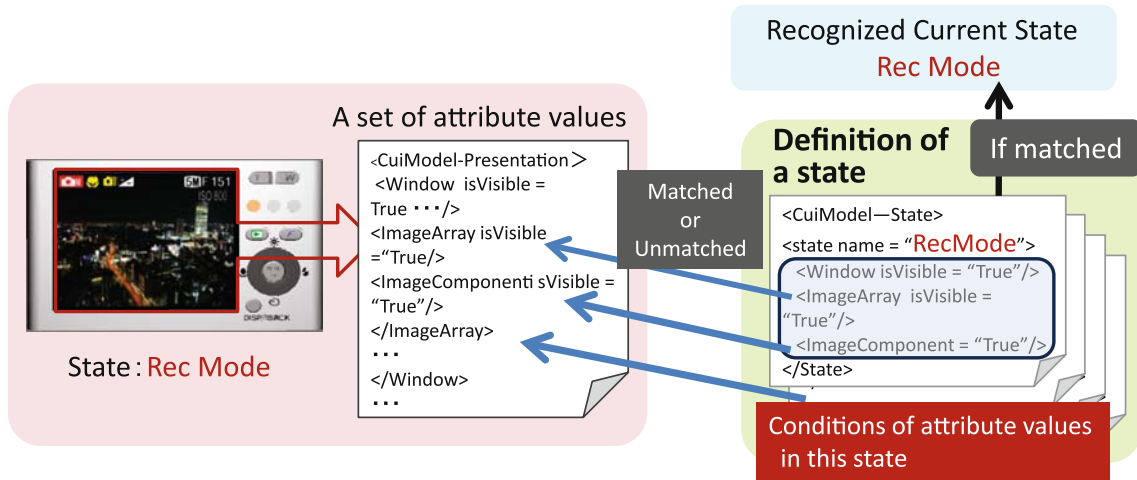
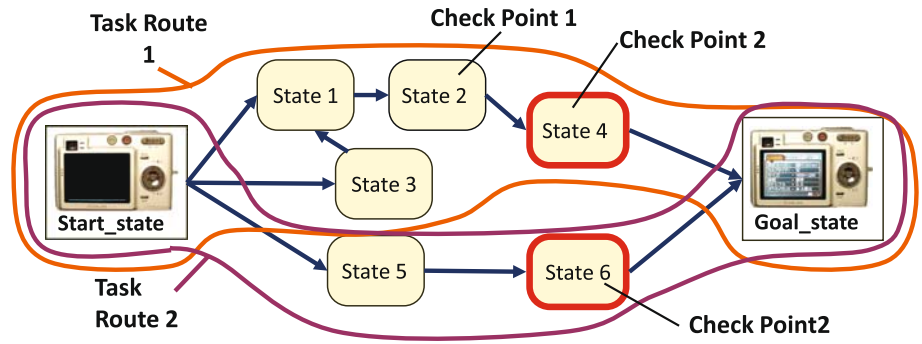


Fig. 12 State evaluation function

be allowed for one task in the model. Moreover, lower and upper bounds for the number of operations or an allowable elapsed time between every two neighboring checkpoints can be also defined. If the elapsed time of a subject’s operation stays within these bounds, the operation is judged to be correct. In this way, usability professionals can adjust a range of correct operations in the task when determining the number of error operations and the task completion.

We developed a tool where a usability professional can interactively set the tasks. To define the task, he/she has to only pick up the start state, goal state and checkpoints on the screen of the VISIO [3]. A short text indicating a goal of the task to the subjects during the test is also input. The defined tasks are exported to a XML document of the test task data.

### 5.2 State evaluation and operation logging

The dynamic behavior model of the UsiXML is expressed by a set of transformation rules which describe how the attribute values of the objects displayed in the UI screen have to be changed in response to the input event. Therefore it is the “event-based” dynamic behavior model, and the model does not have the explicit notion of “state” of the UI.

However, in the user test, the task model was originally designed for the “state-based” dynamic behavior model of the UI system [2]. And the notion of the state is indispensable for defining test task, recording user operation logs and identifying missed operation as described in Sect. 5.1. Without the notion of the state, it is very difficult for the usability professionals to capture the situations of user operations.

To solve the problem, we added the state evaluation function in the user test execution function. Figure 12 shows an example of the state evaluation function. In this function, a set of conditions which describes attribute values of a CUI object to be taken in a particular “state” are defined in an XML document in advance. And the function always evaluates whether the condition holds or not in the UI simulation of the user test. If a set of attribute values of the displayed CUI object is perfect match for the condition, then the function reports that the UI transits to the certain predefined state.

In some cases, there might be several different modes with a same set of attribute values in the behavior model. In this case, our state evaluation function cannot identify these modes as different states. Inserting an extra attribute for distinguishing one mode from others into the original set of attributes can solve this problem.

**Fig. 13** 3D digital prototype used in a user-test



Using this mechanism, the user test execution system can recognize which state the UI is in during the simulation. We also developed the operation logging function based on the state evaluation function. The logging function records all subject's operations in the form of the combination of a time, a previous state, a next state and an input event coming from the user interaction. The function saves these logs in a XML document. We integrated the state evaluate function and the operation logging function with the user test execution function, and enabled the usability professionals to manage the user operation log for every task during the user test.

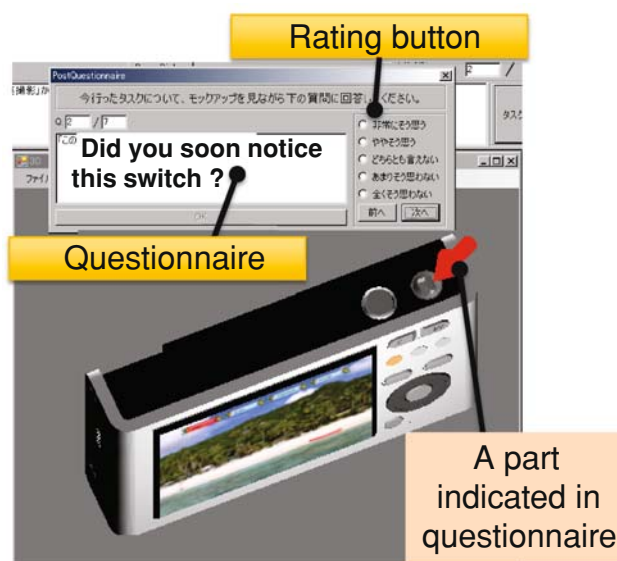
### 5.3 Test execution

Shown in an example of Fig. 13, at the beginning of every new test session, the user test execution function indicates a goal of each task in the form of an imperative sentence. “Switch shooting mode from still to video” and “set the self-timer for 20 sec” are the typical examples of the goal. The goal is indicated on the other window just above the 3D digital prototype.

The subject is asked to complete the task by operating the UI of the 3D digital prototype in the session. The subject manipulates the 3D prototype by rotating, translating and enlarging it and operates the physical UI objects on the prototype by clicking or dragging them with a mouse in the virtual space. If the physical UI objects, such as buttons, are located on a different surface that cannot be seen from the 2D UI screen such as LCD, the subject have to rotate the prototype so as to make these objects face him/her.

During the UI operation, the operation logging function records a sequence of state-transitions of the UI as a list of combinations of state and event together with the time stamps.

At the end of the every test session, the operation logging function compares the actual sequence of state-transitions



**Fig. 14** Digital questionnaire on the digital prototype

with all pre-defined task routes, allowable number of operations and elapsed time between checkpoints. Then the logging function judges whether the subject's operation of the session for this task was correct or not, and identify which checkpoint state the subject made mistakes in his/her operation.

If the operation is judged to be wrong, a set of digital questionnaires are progressively popped up on another screen at the end of the test session. One questionnaire is displayed corresponding to one checkpoint state at which the subject made a mistake. A portion on the UI object in the 3D digital prototype related to each questionnaire is automatically pointed by the tool as shown in Fig. 14.

The subject is asked to answer to each questionnaire by choosing his/her impression from five grades. For example, when the questionnaire is “Did you soon notice this button?”,

the rating is the one of “*Strongly agree: 5*”, “*Agree: 4*”, “*Yes and No: 3*”, “*Disagree: 2*” and “*Strongly disagree: 1*”. The subject answers this rating only by clicking one of the radio buttons placed on the questionnaire as shown in Fig. 13. The rating is stored to clarify what needs improvement in the design in the usability assessment function. The detail of this digital questionnaire is explained in Sect. 6.3.

## 6 Usability assessment function

### 6.1 User performance measures

The usability assessment function investigates the operation log data by comparing it with the test task and the dynamic behavior model of the UI. The function outputs the measures of usability assessment as a result of the analysis. The analysis can be automatically processed, and the function outputs measures of the user performances.

We adopted the following four performance measures based on three basic notions of usability (effectiveness, efficiency and satisfaction) defined in [25]:

- the number of user events inputted in each task and in each subject,
- elapsed time in each task and in each subject,
- personal task achievement ratio, and
- Scores for SUS questionnaire [26].

### 6.2 Operational log analysis chart

An actual sequence of operations is compared with one of the correct task routes defined in the test task, 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 15 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 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 checkpoints 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 the depth of the chart becomes larger, the usability professionals can easily recognize at a glance which states the subject did more missed operation in the task.

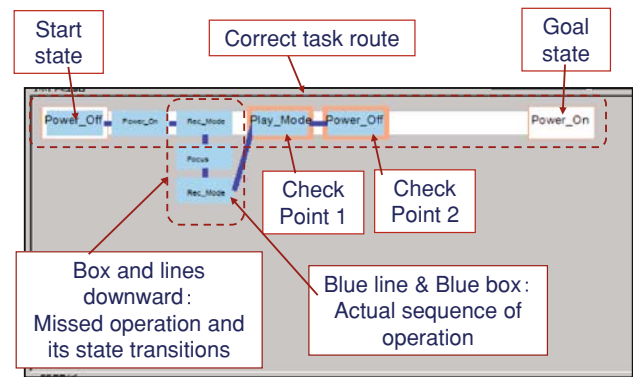


Fig. 15 Operational log analysis chart

### 6.3 Digital questionnaires

The operational log analysis chart can clarify particular states in the UI screen transition where many subjects made errors in the UI operations. However, the chart cannot provide enough information to enable UI designers to find the cause of missed operations and to redesign the UI in order to improve the user’s performance.

To solve this problem, the digital questionnaire execution function has been developed to identify the causes of the missed operations. The structure of the digital questionnaires is built based on an extension of the cognitive walkthrough (CW) method which is dedicated by the human–computer interaction (HCI) model.

#### 6.3.1 Cognitive walkthrough

The CW is one of the usability inspection methods where analysts simulate the user’s cognitive behavior and answer a series of evaluation questions for each step of a task [27]. It can detect usability problems by presuming the cognitive behavior path which the expected users will take to accomplish given tasks. To presume the user’s behavior path, the professionals themselves instead of the users answer the questionnaire which is built based on the behavioral model of the user in the cognitive science. Therefore the CW is effective to clarify what the cause is of the wrong operations and what needs improvement in the UI design.

The CW is one of the appropriate usability evaluation methods at the initial design stage of UI development. However, the quality of the evaluation greatly depends on the knowledge of the usability professionals, and the method sometimes overlooks a part of the problems if only the usability professionals execute the walkthrough. To eliminate the drawback, it is desirable that the end users are directly involved in the walkthrough in addition to the usability professionals.

**Table 1** Extended HCI model and the questionnaires

Cognition process of extended HCI model	Items of questionnaires based on extended HCI [28]	Items of questionnaires used in our system
Formation of Intension of manipulation	Does the user try to accomplish the correct action?	Did you easily understand what you should do for the appliance by reading the task?
Perception of objects	Can the user perceive the object to be manipulated?	Did you soon notice this [input element name]?
Interpretation of objects	Can the user understand that the perceived object is the correct object?	Did you soon understand that you should operate this [input element name]?
Perception of actions	Can the user perceive his/her actions of manipulation?	Did you soon understand how you should operate this [input element name]? (by pushing, sliding, etc.)
Interpretation of actions	Can the user come up with actions of manipulation which should be applied to the object?	N.A.
Execution of actions	Can the user certainly execute the correct action?	N.A.
Perception of the effect	Can the user notice the change of the state?	Did you soon notice that the state of this [output element name] changed by your operation?
Interpretation of the results	Can the user understand what state the system is after the state change?	Did you easily understand how the state of the appliance changed as a result of the operation by observing the state change of [input element name]?
Evaluation of the results	Can the user understand that he/she advances toward the solution of the task by observing the system state?	Did you soon understand whether your operation is correct or not by observing the state change of [input element name]?

However, the questionnaires currently used in the walk-through are built for the professionals and the form of the questionnaires is abstract and complex; for example, “Will the user associate the correct action with the effect to be achieved?” These questions are difficult to answer properly unless one has expertise in cognitive science and usability evaluation. To that end, the questionnaire used in the improved CW should be more specific so that end users can easily understand what the questions mean if it is used along with the user test.

### 6.3.2 Digital questionnaire execution function

Considering the above discussion, we implemented a function of executing “digital questionnaires” in our usability assessment function to identify causes of the wrong operations and what needs improvement based on answers returned from the users who are using the UI-operable 3D digital prototype.

To construct the digital questionnaires, we first introduced the extended HCI model. The extended HCI model [28] is an extension of the CW method whose questionnaire is formulated based on an extended model of Norman’s HCI [29] that explicitly distinguishes between object and action, and between perceiving and understanding. The questionnaires were originally used for the usability evaluation of Web sites. The questionnaires based on the extended HCI model are easier to understand for end users. A list of the questionnaires is shown in Table 1.

However, as shown in Table 1, the questionnaires based on the extended HCI model still have abstract expressions. We further make them more straightforward and concrete when using them in the user tests of information appliances so that the end users can understand them more easily as shown in Table 1. For example, a questionnaire of the extended HCI model which examines the perception of the object to be manipulated is expressed as “Will users be able to perceive the object to be manipulated?”. We changed it to more plain one; “Did you soon notice this button?” when applying it to the test of a digital camera.

Moreover, we also implemented a function that automatically points to a 3D object which corresponds to “this button” or “here” on the 3D digital prototype when indicating a certain portion in the questionnaire as shown in Fig. 14. This function enables end users involved in the test to understand each item of the questionnaire more easily, and also enables usability professionals to save a quite bit of manpower for constructing the questionnaire.

The proposed digital questionnaire enables many end users to take part in the CW evaluation and to answer the questions by actually manipulating the 3D digital prototype whose UI can work as same as the final appliance does. This feature can greatly increase the reliability of the user test’s results.

For defining the digital questionnaire, as shown in Fig. 16, the usability professionals assign one questionnaire to a check point in the test task model. The professional also specifies a particular portion of the UI screen image or the 3D model of

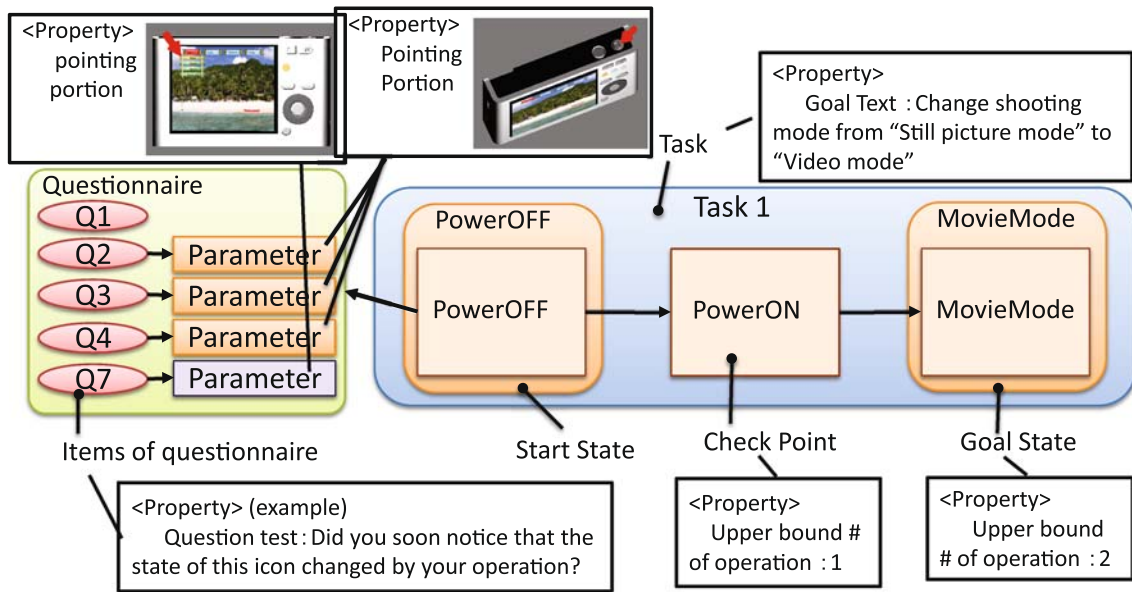


Fig. 16 Association of the digital questionnaire with the test task model

the housing which should be indicated on the 3D prototype. The standard template of the questionnaire is predefined as a stencil with standard properties in the VISIO, and the usability professionals can paste the stencil to the particular task, which is graphically displayed in the Visio, and input the sentence of a question in the property value [3].

## 7 A case study of usability assessment and usability improvement by redesign

### 7.1 Task setting

A case study was done which consisted of the user test, the usability assessment and the UI redesign based on the results of the assessment. A compact digital camera (Fuji FinePix-Z1) on the market shown in Fig. 17 was selected for the user test.

The goals of the case study were:

- to investigate whether the UI operable 3D digital prototype and our tool can clarify the weaknesses in the UI design where many subjects often make mistakes in their UI operation,
- to investigate whether the digital prototype and the tool can clarify why many subjects often make mistakes in the operation and what needs improvement in the UI design,
- to investigate how small the differences in the assessment results are between the digital prototype and the real product, and
- to evaluate how efficiently the redesign of the UI can be done using the UsiXML and our tool.

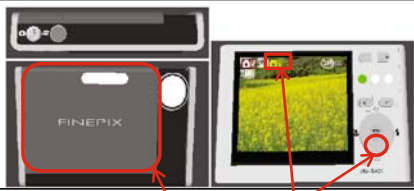


Fig. 17 The appliance for the user test

A UI operable 3D digital prototype of this camera was built as shown in Fig. 13 and used for the assessment. For the digital prototype, we modeled the dynamic behavior model of the camera UI which has 34 states and 224 transitions including neighboring transitions of correct operation sequences of the task. A real product was simultaneously used for the assessment, and the result was compared with that of the 3D digital prototype.

The prototype was operated by 14 subjects (male and female students of age 20–29) who had not used the same model. Seven subjects took part in the test using the UI operable 3D digital prototype, and the other seven subjects used the real camera.

We defined the task of “Setting a self-timer to 10s from power-off state” in the test which is one of the occasionally used operations. In the task, as shown in Fig. 18, first the user has to turn on the power switch by sliding the front lens

Task goal	Setting a self timer to 10 sec from power-off state.
Minimum # of operation	2
Physical UI objects to be operated	
Objects indicated in Questionnaires	<ul style="list-style-type: none"> <li>• Front lens cover</li> <li>• Downward cursor button</li> <li>• Self time icon displayed in LCD</li> </ul>

**Fig. 18** The task of the user test

cover, and then to switch the mode from the manual shooting to the self-timer setting with 10 s by pushing a downward cursor button once. If the camera reaches to this goal state (self-timer 10 s), a small circular white icon which symbolizes the self timer appears on the top of the LCD. The subject has to notice that this icon indicates the goal state and that he/she completed the task.

## 7.2 Analysis of operational patterns

To investigate the subjects' actual operational patterns both of the 3D digital prototype and the real product, actual sequences of operations including missed operations of each subject are put together. The sequences are schematically drawn as a "summarized operational log analysis chart".

This chart can be made by superimposing an operational log analysis chart for one subject shown in Fig. 15 onto ones of the other subjects. In this summarized chart, the notation of correct and missed operations is the same as the one in the personal version described in Sect. 6.2. But the width of a directed path on the chart is proportional to the number of subjects who passed over the transition corresponding to the line. Therefore, a wider directed path indicates that more subjects passed over the routes of operation to complete the task.

Figure 19 shows the two summarized operational log analysis charts; the chart for the subjects who operated the 3D digital prototype (Fig. 19a) and the other for the ones who did the real product (Fig. 19b).

Both analysis charts show the clear facts that:

- at the "Rec\_Mode" state, many users stepped into the wrong path aiming to the "Shooting mode" state instead of the correct path to the "Time\_10sec" state,
- even at the "Timer\_10sec" state which is a goal of the task, many users went past the state and continue operating to reach the "Time\_2sec" state, and

- The pattern of missed operations of subjects who operated the 3D digital prototype is very similar to that of the real product.

From this comparison, the differences of the operational patterns were small between the UI operable 3D digital prototype and the real product. It was also shown that the 3D digital prototype could find the weakness in the UI design where many subjects often take missed operations similar to the those missed operations performed by the ones using the real product.

## 7.3 Analysis of digital questionnaires

When reading only from the summarized operational log analysis chart, we could not discover the reasons why so many subjects made mistakes in those particular states and what needs improvement in the UI design. So we further analyzed the rating from the subjects in the digital questionnaires indicated on the 3D digital prototype.

Moreover, the ratings obtained in the digital prototype were compared with those in the real product. To the subjects who used the real product, the questionnaires were manually indicated to them, and the ratings were written by themselves.

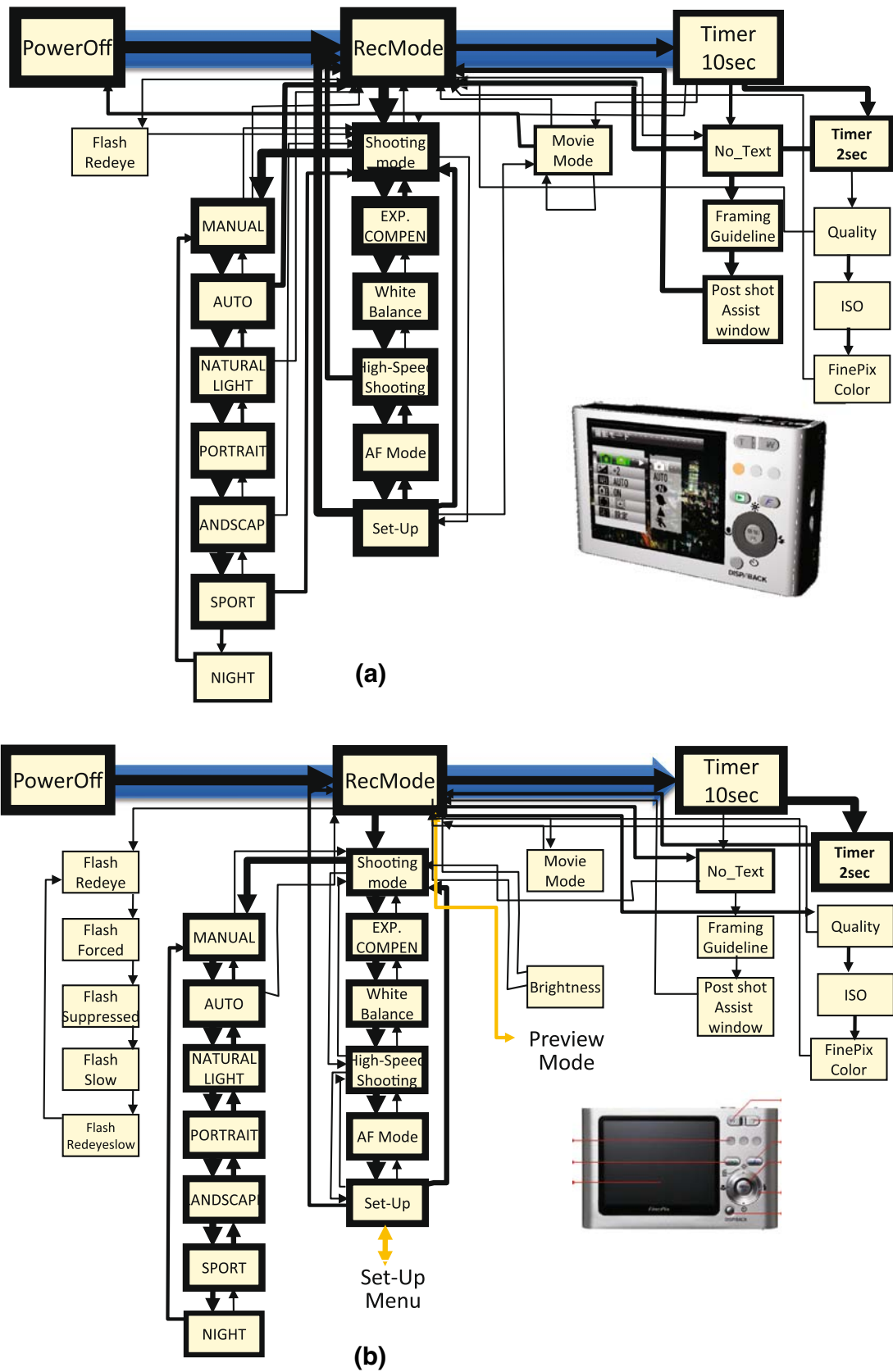
The average ratings from the subjects for the digital questionnaires indicated at the "Rec\_Mode" state and the "Timer\_10sec" state were shown in Fig. 20. "Rec\_Mode" state means that the camera is in the manual shooting mode, and "Timer\_10sec" state that it is in the self-timer setting mode with 10 s and is the goal state.

The ratings at the "Rec\_Mode" state from the 3D digital prototype suggest that many subjects could recognize a down cursor button itself, but could not notice that they could move to the self-timer setting mode from the manual shooting mode by pushing this cursor button. Therefore, from the rating analysis, we found that the small symbol indicating the self-timer printed on the housing surface near the cursor button needs to be changed to a new one which can give us the self-timer function at a glance.

On the other hand, the ratings at the "Timer\_10sec" state suggest that the subjects could notice the change of the system status caused by their operations, but could not understand what occurred in the camera and whether they correctly accomplished the task. This means that this white icon displayed on the LCD in Fig. 20 could be noticed by many subjects, but did not enable them to notice that the self-timer settings had already been set to 10 s. Therefore, from the rating analysis, we finally found that the small timer icon indicated on the LCD in the original UI design need to be improved to the new one which can give us the setting value of the self-timer at a glance.

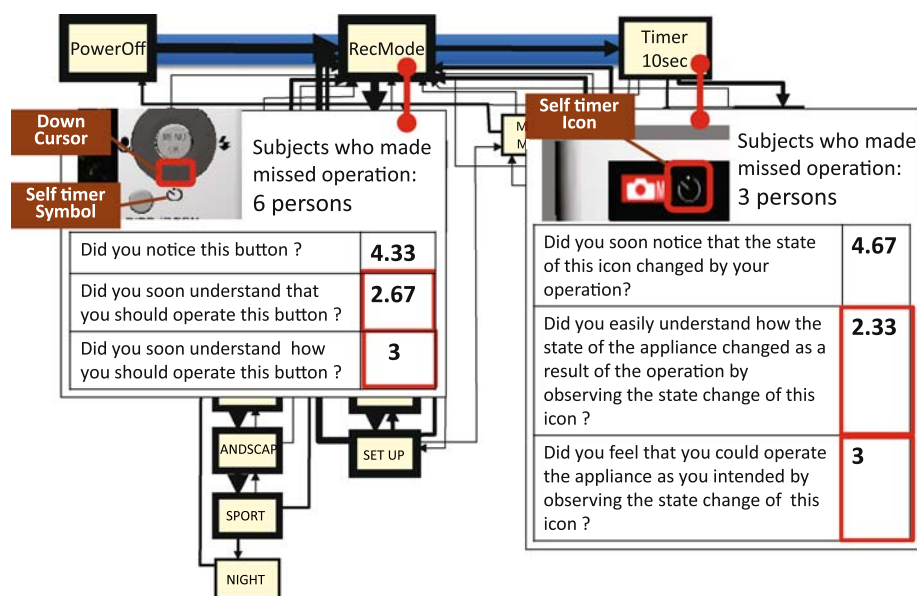
Figure 21 shows the difference in the ratings in three questionnaires between the subjects who used the UI operable





**Fig. 19** Comparison of the summarized operation log analysis charts of the user test. **a** Operational patterns of the 3D digital prototype (no. of subjects: 7). **b** Operational patterns of the real product (no. of subjects: 7)

**Fig. 20** Average rating for the digital questionnaires at 2 states in question in the original UI design



3D digital prototype and the ones who used the real product. There are strong correlations of the ratings between the digital prototype and the real product in all three cases. Therefore, the combination of the UI operable 3D digital prototype and the digital questionnaires enables the UI designers to reveal what needs improvements in the UI and how they should be improved as minutely as a real product does.

#### 7.4 UI redesign based on the digital questionnaires

The rating analysis in the digital questionnaires described in Sect. 7.3 revealed that there are two candidates which should be redesigned in the original UI design; a small symbol indicating the self-timer printed on the housing, and the small self-timer icon indicated on the LCD shown in Fig. 20.

Based on the analysis, we evaluated how efficiently the redesign of the UI can be done using the UsiXML and our tool. In this study, only the second redesign candidate was implemented.

As shown in Fig. 22, we redesigned the shape and color of the timer icon to the new ones so that the background color becomes conspicuous and the timer setting value is explicitly drawn in the icon. An additional test was executed for the new four subjects who used the 3D-digital prototype with the redesigned icon. The result of the test showed that two of four subjects could complete the task without wrong operations. And the rest could also complete the task with small wrong operations. Moreover, the result of the ratings of the digital questionnaire of the redesigned icon indicates that more subjects could easily find that the self-timer settings had already been set to 10s. In this redesign work, it only took 10 min to redraw the icon image and 1 min to rewrite a small part of the tag contents in the XML document of the UsiXML.

From the whole results of the case study, we obtained the following conclusions:

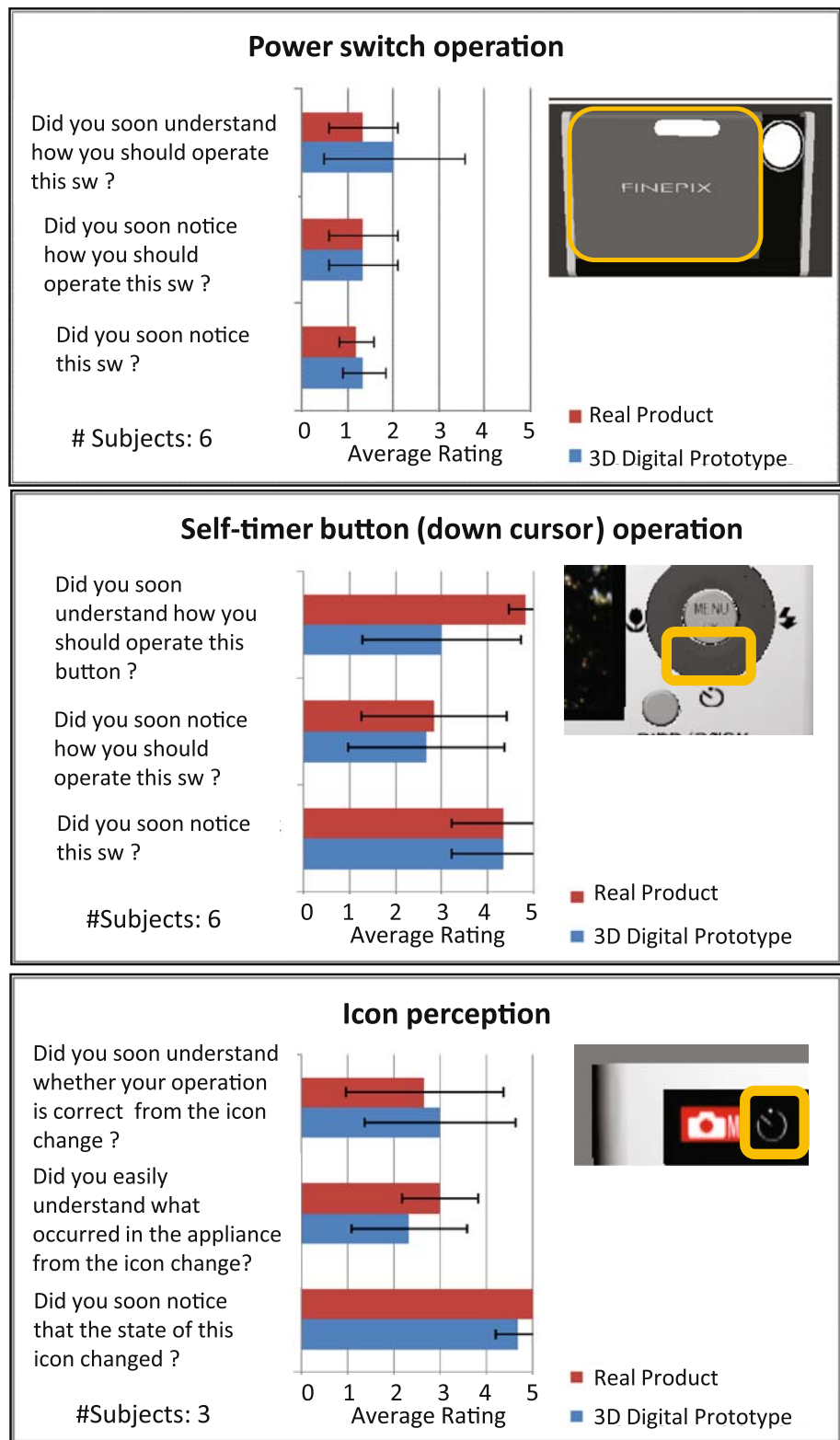
- The summarized operational log analysis chart based on the 3D digital prototype enabled UI designers to discover the weakness in the UI design where many subjects make mistakes, and also that the digital questionnaires enabled them to clarify what needs improvement and how they should be improved in the design.
- There was a strong correlation of the operational log analysis chart and the ratings in the questionnaire between the 3D digital prototype and the real product. Therefore, the UI operable 3D digital prototype could replace a real product or a physical prototype while keeping the ability to discover the usability problems of the UI logic.
- The UI operable 3D digital prototype based on the UsiXML and the automated usability assessment functions can complete the works of prototyping-test-redesign more efficient than the current manual based assessment can.

Therefore, the goals of the case study were fully achieved.

## 8 Discussions

The objective of this research is to indicate that the 3D UI operable digital prototype might streamline prototyping and usability assessment of information appliances. So the technology is still in its fundamental stage, and is not in practical use by the manufactures. Several issues which still need to be solved and their possible solutions are shown in the following:

**Fig. 21** The differences in the ratings in three questionnaires between the 3D digital prototype and the real product

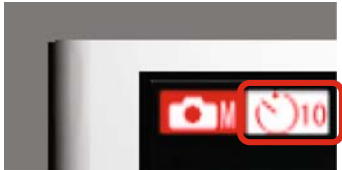


### 8.1 Specifications of the dynamic behavior

In most XML-based user interface mark-up languages, the dynamic behavior part of the UI system is defined by

programming languages and done in procedural way. On the contrary, in this study, the dynamic behavior was modeled based on the event-based and state-based formalisms and was described in a declarative way. The formalism and the

## Redesign of the Timer Icon

		Subjects: 2 persons
Did you soon notice that the state of this icon changed by your operation?	4	
Did you easily understand how the state of the appliance changed as a result of the operation by observing the state change of this icon?	4	
Did you feel that you could operate the appliance as you intended by observing the state change of this icon?	3.5	

**Fig. 22** Improved average rating for the digital questionnaires at “Time 10sec” state in the modified UI design

declarative modeling ability provide us with the following advantages in UI prototyping and usability assessment.

First, in order to obtain meaningful information about the weakness of the UI design from the user-test on the digital prototype, a certain amount of complex dynamic behavior of the UI must be modeled in the prototype so that the subjects can operate it in the same way as they can do in the real product. Usually, the specifications of the dynamic behavior are written as screen transition diagrams in the early design which are based on the state-based formalism [11, 17, 30]. The declarative models based on these formalisms can describe a more complete specification of the UI behavior than a computer program [30]. Second, the explicit declarative modeling of the dynamic behavior based on the formalism enables us to easily exchange the UI behavior information among the computer-aided tools of prototyping, test-execution and usability assessment. However, the information exchange becomes very difficult when using the computer program as behavior information. Third, the declarative model enables the tools to discover not only whether the subjects make mistakes in operations but also to what extent they took the missed operations. However, it is very difficult to discover them if the dynamic behavior is described by the program. Fourth, the formalism provides us the graphical notations of the dynamic behavior model such as Statecharts. They enable the UI designer to capture and express

the UI behavior intuitively using the visual authoring environment [3]. Fourth, the declarative modeling can eliminate the programming task by the UI designers. It can make the turn-around of prototyping-testing-redesigning reduced considerably.

At present, the sophisticated visual authoring tool, which can help UI designers easily build and modify the extended UsiXML models in a visual way is under development, and has to be developed as our future work.

### 8.2 Assessment of physical ergonomics

In this prototyping tool, the subjects can only manipulate the 3D digital prototype by a mouse. Using a mouse as an input device is equal to using one finger to operate an appliance which is floating in the air [11], and some kind of ergonomic measures such as operational times may degrade. They can be considered as the efficiency measure which is one aspect of usability. Moreover, when an appliance has multi-finger inputs in the UI, the 3D digital prototype with single mouse operation may give inaccurate measures.

However, as indicated in our test results, there was a strong correlation of the missed-operational patterns between the 3D digital prototype and the real product. Therefore, the digital prototype can still give a reliable effectiveness measure which is another aspect of usability, if the appliances have multi-finger inputs. The dynamic behavior model is still effective for evaluating the effectiveness even in above cases.

Precise assessments of physical ergonomics are beyond the study of this paper. One possible solution for evaluating them in a fully digital way is introducing the digital human model such as the “digital hand” which we have already presented in another study [31].

### 8.3 Extendibility to multi-modal interactions

Up-to-date information appliances such as the i-Phone have the UI system with recognition-based inputs (speech or gestures). Our 3D digital prototype does not have a function for accepting such kinds of input. One of the possible solutions is to add the functions of a UI operable virtual-physical prototype to the digital prototype where the subject can manipulate a physical housing as the gesture input device. A smart recognition algorithm will also be needed which classifies gestures or speech from an unspecified number of general subjects into several input events [30]. Mixed reality technology will be also useful for projecting the UI screen image on a physical housing [13, 14]. Even in this case, the static structure and the dynamic-behavior model of the UI proposed in our study can be still effective as the information base of prototyping, user-test and usability assessment.

#### 8.4 Considering the effect of UI design on user sensations

From aspect of user satisfaction in usability assessment, UI design of information appliances must provide the users with a good sensation and impression. In our proposed UI redesign method, the digital questionnaires enabled the UI designers to clarify how problems in the initial detailed UI design should be improved in the redesign. However, in the method, how the UI redesign affects user sensations cannot be clarified. Solving the problem is beyond the purpose of this paper but must be achieved in the future. One possible solution is that user's sensory characteristics such as operational comfort must be identified and expressed quantitatively and that these characteristics must be linked to more high-level model constituents in preliminary UI design such as Task & Concepts and AUI in UsiXML. Recently, Doré et al. proposed a unique approach for preliminary design to find links between a sensory characterization and a function characterization of a product [32,33]. They showed an effective example of the approach to the parabolic ski design, and introducing their approach in our tool is likely to enable UI designers to evaluate the influence of redesign of UI on user sensations. In this case, we should carefully observe how the influence on the user's sensations depends on "context of use" of usability assessment in preliminary UI design.

## 9 Conclusions

A 3D tool for digital prototyping and usability assessment of information appliances was newly proposed in our study. The results are summarized as follows:

1. The UsiXML was extended so that UI designers could define the static-structure model including the physical UI elements of the appliances in a declarative way. The real-time 2D rendering function based on the extended UsiXML was developed to reduce the preparatory work of 3D UI simulations.
2. The UsiXML was extended so that UI designers can define the dynamic behavior model based on an event-based formalism in a declarative way. It was shown that the model was effectively used in the realistic 3D UI operable simulation of the information appliances.
3. The user test and usability assessment could be automated based on the static-structure and dynamic behavior model. It was shown that the operational log analysis chart and digital questionnaires enabled usability professionals to easily discover the weakness of the UI design and extract the cause of the weakness. The weakness could be effectively corrected by modifying the declarative descriptions of the UI models.
4. A case study of the 3D digital prototyping and usability assessment of a digital camera was done. It was found that the differences in the subjects' operational patterns and the ratings in the questionnaire were small between the 3D digital prototype and the real product. It was suggested that the UI operable 3D digital prototype could replace a physical prototype while keeping the ability to discover the weaknesses of the UI logic design.

In our future works, more experimental studies have to be done for a greater number of appliances. An introduction of scalable vector graphics to the 2D rendering function of the UI screen is further needed. Functional integration of the UI operable virtual-physical prototype with the 3D-digital prototype is also needed to support the whole process of prototyping.

## References

1. ISO13407:1999: Human-centred design processes for interactive systems (1999)
2. Kanai, S., Horiuchi, S., Shiroma, Y., Kikuta, Y.: Digital usability assessment for information appliances using user-interface operable 3D digital mock-up. In: Fischer, X., Coutellier, D. (eds.) *Research in Interactive Design*, vol. 2, HUCEID2006\_p235. Springer, Paris (2006)
3. Kanai, S., Horiuchi, S., Shiroma, Y., Yokoyama, A., Kikuta, Y.: An integrated environment for testing and assessing the usability of information appliances using digital and physical mock-ups. *Lect. Notes Comput. Sci.* **4563**, 478–487 (2007)
4. Landay, J.A., Myers, B.A.: Sketching interfaces: toward more human interface design. *IEEE Comput.* **34**(3), 56–64 (2001)
5. Lin, J., Newman, M.W., Hong, J.L., Landy, J.A.: DENIM: finding a tighter fit between tools and practice for web site design. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2000)*, Hague, pp. 510–517 (2000)
6. Avrahami, D., Hudson, S.E.: Forming interactivity: a tool for rapid prototyping of physical interactive products. In: *Proceedings of the 4th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, London, pp. 141–146 (2002)
7. Lee, K.P.: User-participatory testing using web. In: *Proceedings of 11th International Conference on Human-Computer Interaction*, Las Vegas, vol. 4 (2005)
8. Brutzman, D., Daly, L.: X3D: Extensible 3D Graphics for Web Authors. Morgan Kaufmann, San Francisco (2007)
9. ISO/IEC 14772-2:2004: Information technology—Computer graphics and image processing—The Virtual Reality Modeling Language (VRML)—Part 2: External authoring interface (EAI) (2004)
10. Liu, D., Le Lacheur Sales, S.: *Virtools Fundamentals*. Axis 3D Technology Inc., Taipei (2007)
11. Kuutti, K., Battarbee, K., Sade, S., Mattelmaki, T., Keinonen, T., Teirikko, T., Tornberg, A.-M.: Virtual prototypes in usability testing. In: *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, vol. 5(5), pp. 5029–5035 (2001)
12. Bruno, F., Mattano, R.M., Muzzupappa, M., Pina, M.: A new approach to participatory design: usability tests in virtual environment. In: Fischer, X., Coutellier, D. (eds.) *Research in Interactive Design*, vol. 1, VC200560. Springer, Paris (2005)

13. Verlinden, J., Van den Esker, W., Wind, L., Horvath, I.: Qualitative comparison of virtual and augmented prototyping of handheld products. In: Proceedings of International Design Conferences, Dubrovnik, pp. 533–538 (2004)
14. Nam, T.-J.: Sketch-based rapid prototyping platform for hardware-software integrated interactive products. In: CHI '05 Extended Abstracts on Human Factors in Computing Systems, Portland, pp. 1689–1692 (2005)
15. Takahashi, H., Shimazaki, S., Kawashima, T.: Augmented reality system for development of handy information device with tangible interface. *Lect. Notes Comput. Sci.* **4563**, 564–573 (2007)
16. Aoyama, H., Kimishima, Y.: Development of system using mixed reality technology for evaluating designability and operability of product. In: Fischer, X., Coutellier, D. (eds.) *Research in Interactive Design*, vol. 2, WoIM2006188. Springer, Paris (2006)
17. Kerttula, M., Tokkonen, T.K.: Virtual design of multi-engineering electronics systems. *IEEE Comput.* **34**(11), 71–79 (2001)
18. Horrocks, I.: *Constructing the User Interface with Statecharts*. Addison-Wesley, Harlow (1999)
19. Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., Jaquero, V.L.: UsiXML: a language supporting multi-path development of user interfaces. *Lect. Notes Comput. Sci.* **3425**, 200–220 (2005)
20. Phanouriou, C.: UIML: A Device-Independent User Interface Markup Language. Ph.D. thesis, etd-08122000-19510051, Virginia Polytechnic Institute and State University (2000)
21. Bullard, V., Smith, K.T., Daconta, M.C.: *Essential XUL Programming*. Wiley, Hoboken (2001)
22. Vanderdonckt, J.: A MDA-compliant environment for developing user interfaces of information systems. *Lect. Notes Comput. Sci.* **3520**, 16–31 (2005)
23. MacVittie, L.: *XAML in a Nutshell*. O'Reilly Media Inc., Sebastopol (2006)
24. *3D XML User's Guide*. Ver 1.0. Dassault Systems Inc., Paris (2008)
25. ISO9241-11:1998: Ergonomic requirements for office work with visual display terminals (VDTs)—Part 11: Guidance on usability (1998)
26. Brooke, J.: SUS: a quick and dirty usability scale. In: Jordan, P.W., Thomas, B., McClelland, I.L., Weerdmeester, B. (eds.) *Usability Evaluation in Industry*, pp. 189–194. Taylor & Francis, London (1996)
27. Polson, P.G., Lewis, C., Rieman, J., Wharton, C.: Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. *Int. J. Man-Mach. Stud.* **36**, 71–73 (1992)
28. Hori, M., Kato, T.: A modification of the cognitive walkthrough based on an extended model of human-computer interaction (in Japanese). *Trans. Inform. Process. Soc. Jpn.* **48**(3), 1071–1084 (2007)
29. Norman, D.A.: *Cognitive engineering*. In: Norman, D.A., Draper, S.W. (eds.) *User centered systems design: new perspectives in human-computer interaction*, pp. 31–61. Lawrence Erlbaum Associates, Hillsdale (1986)
30. Hartmann, B., Abdulla, L., Mittal, M., Klemmer, S.R.: Authoring sensor-based interactions by demonstration with direct manipulation and pattern recognition. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, San Jose, pp. 145–154 (2007)
31. Endo, Y., Kanai, S., Kishinami, T., Miyata, N., Kouchi, M., Mochimaru, M.: Virtual ergonomic assessment on handheld products based on virtual grasping by digital hand. *SAE Trans. J. Passeng. Cars Electron. Electr. Syst.* **116**(7), 877–887 (2008)
32. Doré, R., Pailhes, J., Fischer, X., Nadeau, J.-P.: Identification of sensory variables towards the integration of user requirements into preliminary design. *Int. J. Ind. Des.* **37**(1), 1–11 (2007)
33. Doré, R., Pailhes, J., Fischer, X., Nadeau, J.-P.: Identification of design variables and criterion variables towards the integration of user requirements into preliminary design. *Int. J. Prod. Dev.* **4**(5), 508–529 (2007)