Interactive Edutainment System with enhanced Personalized User Interface Framework

Youngho Lee and Woontack Woo, *Member*, IEEE

Abstract — *In this paper, we present the Responsive Multimedia System, an interactive edutainment system.. The system consists of an AR-based tabletop tangible user interface (ARTable) and a virtual environment manager (VEManager) integrated by enhanced Personalized User Interface Framework. In this system, a user interface and virtual environments are integrated seamlessly by sharing user-centric context. To share user-centric context efficiently, our system optimizes the number of network connection between objects in real and virtual environments. The synchronization problem between ARTable and a virtual environment is solved by calculating coordinates with additional information such as collision with terrain and physics law in a virtual environment as well as user-centric context. Moreover, it reorganizes responses of an object according to a user-centric context and condition-action rules. To demonstrate the effectiveness of the proposed system we developed an application on the subject 'Dream of Mee-luck'. According to the experimental results, we found that the proposed system could be a promising technology that enables users to experience an interactive story. In the future, the proposed system can play an important role in developing interactive applications for edutainment.*

Index Terms - Responsive Multimedia System, ARTable, Personalized User Interface

I. INTRODUCTION

With the increasing interest of edutainment computing, various multimedia technologies have been developed during the last decade. New human-computer interfaces, real-time graphics, immersive displays, 3D sound, character animations, and artificial intelligence allow one to realize novel interactive edutainment systems. Thus, researchers have designed and implemented interactive edutainment systems with novel multimedia technologies. However, developing an interactive edutainment system is still a challenging problem since we need to consider how to merge these multimedia technologies for creating and telling interactive stories for interactive edutainment systems.

Many interactive edutainment systems have been developed for supporting user interaction with contents [1-7]. The DiamondTouch table provided touch as an interaction method among multiple users and it was applied to a multi-user map viewer [4]. The reacTable was used as a musical instrument [5], and the Planar Manipulator Display was used for simulating the furniture layouts in an interior space [6]. The Dialog Table promoted social interactions among visitors and provided access to the museum's multidisciplinary collections [7]. However, they are focused on direct user interaction without reflecting user-centric context such as the user's background knowledge and interest. They do not mention interactions with virtual objects in a 3D virtual environment even they discussed about 2D digital contents such as pictures and map. In addition, the lack of intelligent responses limits the depth of interactions between users and virtual objects.

In this paper, we present the Responsive Multimedia System (RMS), an interactive edutainment system. The RMS consists of an AR-based tabletop tangible users interface (ARTable) and a virtual environment manager (VEManager) integrated with the enhanced Personalized User Interface Framework (ePUIF). In this system, modified publisher-subscribe algorithm composes a multicasting group of objects dynamically and distributes context from a sensor among the group. Context-based navigation algorithm is an injective mapping from 2D control interface to 3D virtual environments. It calculates uncontrollable free variables of coordinates in virtual environment with user-centric context [13]. In addition, condition-action rules are applied to decide responses of an object according to a user-centric context.

In this system, a user interface and virtual environments are integrated seamlessly by sharing user-centric context. In order to share user-centric context efficiently, our system optimizes the number of network connection between a large number of objects in real and virtual environments. The synchronization problem between ARTable and a virtual environment is solved by calculating coordinates with additional information such as collision with terrain and physics law in a virtual environment as well as user-centric context. Moreover, it reorganizes responses of an object according to a user-centric context and condition-action rules.

To demonstrate the usefulness of the proposed system, we implemented a virtual heritage which presented a legend of *Unju Temple* in S.Korea. We named it '*Dream of Mee-luck: Aspiration for a new dawn*' [10]. Fig. 1 shows the overall system architecture of the Responsive Multimedia System. To experience the virtual heritage system, users can select a specific doll according to their preference. Then, users follow the guide of interactive viewer to visit several places which show events adapted to the user's profile acquired from digital questionnaire. Users can experience personalized story by manipulating the control object interacting with interactive

This research is supported by the UCN Project, the MIC 21st Century Frontier R&D Program in Korea.

Y.Lee is with the Department of Information and Communications, GI ST (e-mail: ylee@gist.ac.kr)

W.Woo is with the Department of Information and Communications, GIST (e-mail: wwoo@gist.ac.kr)

Contributed Paper
Original manuscript received March 29, 2007
Revised manuscript received April 12, 2007

viewer and virtual objects. We believe that the proposed system can be applied to several application areas, such as historical education, interactive entertainment and games.

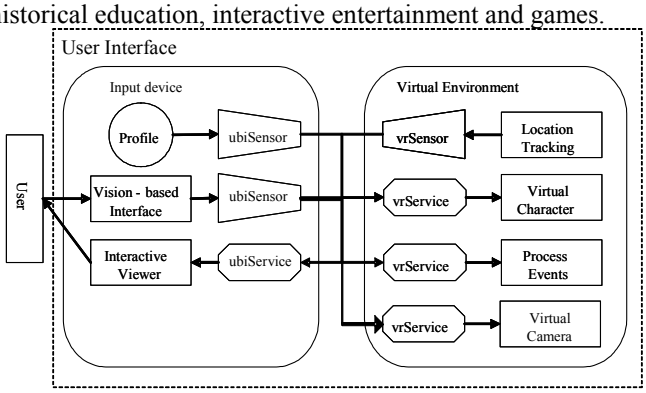


Fig. 1. Overall system architecture of the proposed Responsive Multimedia System

This paper is organized as follows. In Section 2, we explain the enhanced Framework for a Personalized User Interface and its functions. In Section 3, we describe ARTable and VEManager of RMS. In Section 4, we explain an implementation and the story presented in our system. In Section 5, experimental results are discussed. Finally, conclusions and future work are presented in Section 6.

II. ENHANCED FRAMEWORK FOR A PERSONALIZED USER INTERFACE

In this section, we explain the enhanced personalized user interface framework (ePUIF) [8]. The ePUIF is a framework for developing a personalized user interface which allows a user to access and customize smart objects in real and virtual environments and to experience personalized responses of the objects. We assume that there are hundreds of sensors and services in real and virtual environments.

Objects compose a user interface in real and virtual environments and are classified into sensors and services according to their logical function. All of them are connected to a ubi-,vrSensor and a ubi-,vrService in the ePUIF [9][12]. The ubi-,vrSensor acquires contexts from active area and delivers the contexts to services. Then, the ubi-,vrService fuses the contexts, calculates parameters for application and triggers the actions. Moreover, contexts related audience is saved in the user profile. It delivers certain services to initialize and optimize whole system through the ePUIF.

The ePUIF exploits user-centric context as a protocol of communication. User-centric context is defined as "*user-centric information among a variety of contexts in a service environment that is interpreted, in terms of 5W1H, by applications*"[13]. The user-centric context is classified into '*preliminary context* (*PC*)', '*integrated context* (*IC*)', '*final context* (*FC*)', and '*conditional context* (*CC*)' [13]. The *CC* is categorized into two types, *user conditional context* and *service conditional context*. It is described by users and system developers respectively.

ePUIF has a symmetric structure integrating ubi-UCAM and vr-UCAM. ubi-UCAM and vr-UCAM stand for a unified context-aware application model for real and virtual environments respectively as shown in Fig 2. Each UCAM is composed of Sensor and Service. Sensors extract the user's *preliminary context (PC)* containing limited information about changes in the virtual environments. Services periodically integrate contexts containing meaningful contextual information, and compares the *integrated context (IC)* with *conditional context (CC)* defined by developers. Finally, the service decides on the state of a virtual object, and delivers its decision to the virtual object.

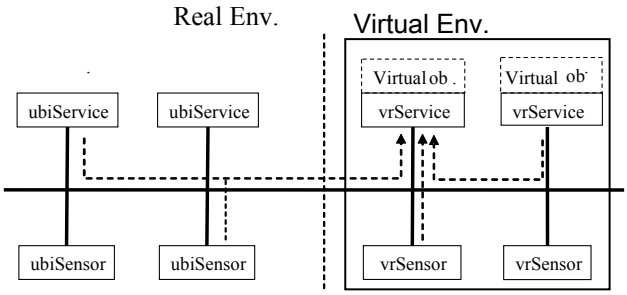


Fig. 2. Enhanced framework for a personalized user interface

A. Model definition

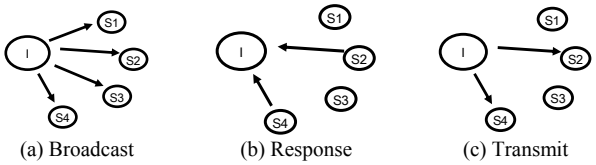
The ePUIF is defined by a four-tuple $U =< I, S, E, C >$, where I is a set of sensors, S is a set of services in both real and virtual world, C is a set of *Conditional Contexts*, and $E\ (\subseteq (I \times S) \cup (S \times I))$ is a set of connection lines between sensors and services. We do not distinguish virtual sensors or services with real sensors or services in this model. The connections E between sensors and services are represented by an adjacency matrix. It is a matrix with rows and columns respectively labeled by services (I) and sensors (S), with a 1 or 0 in position (I_i , S_s) denoting whether I_i and S_s are connected or not. Each sensor I_i has one of rows.

B. Accessing Objects

We need to support seamless connection between sensors or services in real and virtual environments. Network self-configuration manager (SCM) has the role to compose a multicasting group dynamically and to distribute a context among the group. It is embedded in each sensor and service. SCM applies a publisher-subscriber mechanism. When a sensor generates a *PC*, it notifies to services in the same active area. If a service needs to receive the *PC*, it responds to the notification. Then, the sensor can make multicast group list. Finally, the sensor sends the *PC* to the services in the list. Fig. 3 illustrates this mechanism and Fig. 4 shows an example of the adjacency matrix.

The self-configuration manager is composed of three modules; context publisher, configuration manager, and context subscriber. The configuration manager checks active area, type, and state of vrServices and vrSensors. That information is stated in them by developer. The context provider composes multicasting group whenever vrSensor generates a *PC*. After composing the group, it delivers the *PC* to registered vrServices over the network. The context

subscriber decides to join or not to join the multicasting group according to the information in the configuration manager. After it decides to join, it continuously receives the *PC* from vrSensors.



(a) Broadcast (b) Response (c) Transmit

Fig. 3. Publisher-Subscriber Mechanism

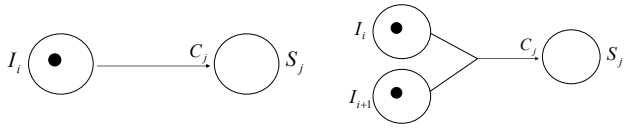
$$\begin{array}{c} \\ I_1 \\ I_2 \\ \cdots \\ I_M \end{array}\begin{array}{c} \begin{array}{cccc} S_1 & S_2 & \cdots & S_N \end{array} \\ \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{pmatrix} \end{array} \Rightarrow \begin{array}{c} \\ I_1 \\ I_2 \\ \cdots \\ I_M \end{array}\begin{array}{c} \begin{array}{cccc} S_1 & S_2 & \cdots & S_N \end{array} \\ \begin{pmatrix} 1 & 1 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & 1 \\ 1 & 0 & \cdots & 0 \end{pmatrix} \end{array}$$

Fig. 4. Adjacency matrix of connections E

C. Action Customization and Intelligent Responses of a Object

To customize a reaction of objects, we describe a *CC* in a Service. *CCs* are if-then rules that trigger events. For example, we can write a *CC* with a meaning of "if the location of a user A is 'where', then trigger the action b."

The service decides the action of a virtual object. After getting *PCs* from Sensors, they are first integrated to form and *IC*, as a complete description of the current context. The service then starts to compare the *IC* with *CC*s. Firstly, it checks sub-context (5W1H) of *IC* and *CC*. Secondly, it finds a matching context through an 'OR' operation. Thirdly, it compares sub-fields between contexts with the same template through an 'AND' operation. Lastly, it selects an action for the object as described in the then-part of the matching *CCs* list. In this moment, *CC* should be included in the PC logically. When the comparison result is true, service provider triggers actions of an object. Fig. 5 shows reaction model of an object. Input from sensor I_i satisfying condition C_j activates a service S_j. In Fig. 5 (b), two inputs from sensor I_i, I_{i+1} satisfying condition C_j activates a service S_j.



(a) one condition case (b) two conditions case

Fig. 5. Reaction model.

III. ARTABLE AND VEMANAGER

A. ARTable

ARTable is an AR-based table-type tangible user interface [11]. It consists of table frame, camera, LCD projector, workstation, and control objects as shown in Fig. 6. The table frame has a semi-transparent display screen. The LCD projector projects images on the screen. Since it is semi-transparent, the control object put on the screen is tracked by a camera tracker under the table. Control objects have familiar shapes such as doll, toy, etc.

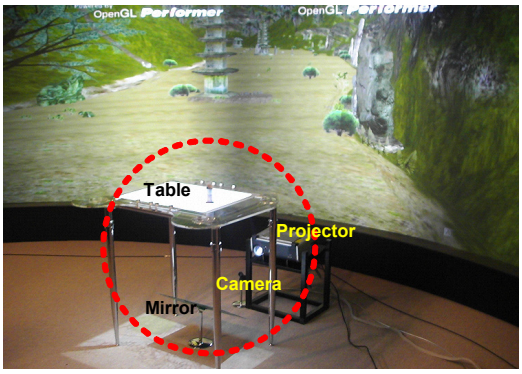


Fig. 6. ARTable is an AR-based table-type tangible user interface

Fig. 7. A user puts the control object on the interactive viewer.

The semi-transparent screen of ARTable allows object tracker to track a control object on the screen while an image is projected onto it. The method is as follows: Initially, it does not display images. After putting a control object onto the table, the object tracker finds ID and position. Then, the interactive viewer displays images except the circle centered on the position of the object. Thus, it can detect the position and ID of control objects from below while it displays images.

ARTable provides users with interaction methods by combining basic control object movement with an interactive viewer. It allows users to move a control object on the surface above the table as shown in Fig.7. Below the object, it has AR-marker so that the camera tracker tracks the 3D position of a control object. From this configuration, it allows daily-life interaction. In our virtual heritage application, for example, we carry out navigation. Users watch a geometric map which is shown by interactive viewer and put the control object on the specific location on the map they want to see. Or we can input or select information by putting control object on a button shown on the table for conventional interaction.

B. VEManager: Virtual Environment Manager

VEManager is a set of virtual objects, such as a virtual camera and animated characters. As mentioned in section II, all virtual objects are connected with vrServices. A virtual camera is a motion camera or a set of still cameras which are designed to

behave as a motion camera, or a camera taking images of objects in the virtual world. An animated character has a finite state machine so that the animation can be triggered by stimulating actions.

The displacement and viewing parameters of a virtual camera in 3D space are represented in x, y, z coordinates. The followings are specific parameters in graphic libraries.

- camera position P(x,y,z) : 3D position in a virtual environment
- camera orientation O(dx,dy,dz) or Aim Point P(x,y,z)
- camera parameters : parameters such as focal length (wide angle, telephoto lens), depth of field, and binocular convergence.
- viewing properties: fog, light attenuation, spot light, etc.

We apply a finite state machine with state and transition condition to animated characters. The state means the possible actions of a virtual object. The transition condition means the condition to change state from current to next state. It can be defined, where is a set of states, is a set of transitions, and is a set of arrows from current state to the next state. We can mark the initial state of a virtual object. Starting from the initial mark, if a transition condition is satisfied, we select next states or actions. The current state of virtual character is p1. If it receives a trigger signal t1, it moves to the next state p2.

C. Context-based Navigation Algorithm

Since we connect a small size 2D map on ARTable with a large scale 3D terrain model in a virtual environment, synchronization problems have occurred. Many variables have to be calculated or predicated from insufficient information provided by the 2D controller in order to put the virtual camera in a certain position. We follow two steps to calculate 3D coordinates as shown in Fig. 8. Firstly, height and view angle are decided based on physics law and geometric information in a virtual environment. Basically, collision detection is applied. Secondly, user's preference and scenario are used for adjusting coordinates of specific virtual objects. This part are triggered by ePUIF according to the given condition-action rule and user-centric context.

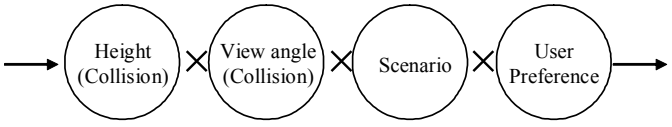


Fig. 8. 3D coordinates is calculated through above steps. Height and view angle are from collision detection. Scenario and user preference can also adjust coordinates.

We can represent a coordinate change from 2D surface to 3D space with the function described in (1). The u and v are coordinates of surface and θ_z is rotation angle of z-axis. x, y, z are coordinates of 3D space and $\theta_x, \theta_y, \theta_z$ are rotation angles. We need to calculate 3D coordinate and orientation. Since, we only know 2D coordinate and rotation angle in z-axis. Furthermore, we should apply several variables for viewing properties such as fog, FOV, etc.

$$G:(u,v,\theta_z)\rightarrow(x,y,z,\theta_x,\theta_y,\theta_z) \quad (1)$$

Let's assume that there is a 3D surface S and the point $U=(X(u,v),Y(u,v),Z(u,v))$ is projected onto the point $u=(u,v,1)$ in 2D plain. Then, we can get the (2), where p is a scale factor, $T=(p_x,p_y)$ is a translation matrix, C is a 3 x 1 translation matrix, and R is a 3 x 3 rotation matrix. Here, if we assume K = R = C = I, the point on surface S is $(u,v,f(u,v))$.

$$u=\begin{pmatrix} p & 0 & p_x \\ 0 & p & P_y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} R & -RC \\ 0 & 1 \end{pmatrix}U=KR[I\,|\,-C]U \quad (2)$$

To solve the coordinate problem, firstly, we calculate scale p = (unit in ARTable)/(unit in VE) by measuring the metric in ARTable and a virtual environment. The translation and rotation matrix is calculated by selecting 3 corresponding points. In order to simply the calculation, we change the camera coordinate system to an arbitrary N x M rectangular coordinate system. When camera detects movement of an object on table, it delivers changed coordinate N x M to VE over the network. At the VE System, we receive the coordinate and translate it to the 3D coordinate system from N x M.

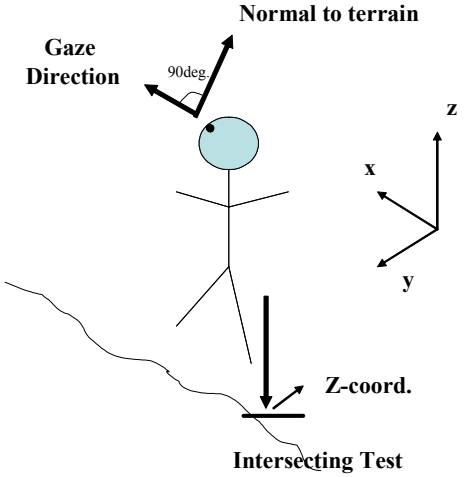


Fig. 9. z-coordinate and viewing angle of a virtual character or a virtual camera.

Another problem is that the 2D Map provides x- and y-coordinates without z-coordinate. Moreover, viewing direction is not provided. In order to calculate z- coordinate, we exploit collision detection between a virtual camera and 3D terrain. We make a straight line to down direction N_g = (0, 0, -1) and detect intersection with terrain model. In order to calculate viewing direction, we calculate normal vector N_a of terrain surface and get θ_x as shown in (3). Thus, avatars can climb the mountain as shown in Fig. 9.

$$\theta_x=\cos^{-1}\left(\frac{N_g}{N_a}\right),\ \theta_y=0,\ \theta_z=R_z \quad (3)$$

Virtual camera and character's parameters are planed as a form of condition-action rule. As show (4) and (5), state of user interface and user-centric context can be used for condition-action rules.

- If StateofViewer(map1 || map2 || map3) ∧ objectTracker(dosun || dongja) = (x,y) → vrcharmoveto(dosun || dongja, (x,y)) (4)
- If StateofViewer(map1 || map2 || map3) objectTracker(dosun || dongja) = (x,y) vrcameramoveto(dosun, (x,y,z +10), (h,p,r)) (5)

IV. IMPLEMENTATION

A. Legend of Unju Temple – Dream of Mee-luck

Unju Temple is a temple located in Daecho-ri, Hwasun-gun, Jeonnam Province in Republic of Korea [10]. At the end of Unified Shilla Dynasty, Reverend Priest Doseon tried to construct thousands of Buddhist statues and stone pagodas in a single night. By implementing the legend, we try to show wishes of ancient people, who put tremendous amount of effort building Buddhist structures, and the beauty of *Unju Temple*. We apply the following scenario: *Users watch animation of the legend of Unju Temple. After understanding the legend, they move in front of ARTable and virtual environment. They start to navigate virtual Unju temple. There are several events to unfold the story in specific position in virtual Unju Temple* .

TABLE II shows the type of objects in the proposed system and TABLE I shows events based on the legend of Unju temple.

TABLE I
SCENARIO IN SPECIFIC LOCATION

Location	Events description
Great tree in front gate	When a user gets close to the Great Tree, the user can hear sounds of wind and birds.
Road of pagoda	A bird sings. The fog disappears.
Cherry blossom	The cherry three reacts according to user's gesture and context.
Construction rock	Stars twinkle up in the sky.
Chilsung Rock	A user can put his/her lotus light.
Lying Buddha statue	A user can watch the lying Buddhist at one sight from higher position. When enough number of lotus lights are turned on, lying Buddhist starts to stand up.

TABLE II
TYPE OF OBJECTS IN REAL AND VIRTUAL ENVIRONMENTS

	Type of Object
ARTable	Control_Object1, Control_object2, ObjectTracker, InteractiveViewer
Virtual Environment	Dongja, Dosun, Lady, Lying Buddhist LotusLight[6], Weather(day, night, fog), MessageBoard, SoundPlayer

B. Sensor and Service

We implemented sensors and services based on the ePUIF. RMS is categorized into sensors and services, regarding the scenario and its functions.

TABLE IV and TABLE III list an example of sensors and services. The sensors capture fluent values; the state of objects and the services performs an action. The services include if-then rules in the form of 'precondition – action' as shown in (6). Precondition is a condition given by sensors. Several services can be triggered simultaneously since each service has own rules.

- If vrSymbolicLocation(dosun, Great Tree) = true, then SoundPlay(wind || birds) (6)

TABLE III
VRSENSORS AND VRSERVICES IN A VIRTUAL ENVIRONMENT

	Name	Description
vrSensor	vrLocationTracker(c) = (x,y)	The current position of virtual character c on VE is (x,y)
	vrSymbolicLocation (x, location)	dist(location, x)<10
	VRDay(t)	It is day in current time t.
	VRNight(t)	It is night in current time t.
vrService	vrcharmoveto (c,(x,y))	A character c moves to position (x,y).
	vrcameramoveto(c, (x,y,z), (a,b,c))	A virtual camera moves to position (x,y,z) and orientation (a,b,c).
	vrweatherchange(x, -x)	Change day (or night) to night (or day).
	vrlotuslighton(c,l)	A character c carries the lth lotus light.
	vrlotuslightoff(c,l)	A character c puts the lth lotus light.
	ChangFogLevel(x)	It changes fog level as x. (0 – 1000)
	SoundPlay(x)	It plays sound file x.

TABLE IV
UBISENSORS AND UBISERVICES IN ARTABLE

	Name	Description
ubiSensor	object Tracker(c)=(x,y)	The current position of control object c on ARTable is (x,y)
	UserAns(x)	X can be Beginner, Intermediate, Expert, Never_scene, Just_know, Well-know
ubiService	interactive Viewer(x)	It displays image x on the ARTable.

C. Personalization of User Interface

The procedure of the personalization is question-answer, making user profile, and adaptation. We select rule-based filtering since it is proper for short time exhibition and easy to expect results. Then, through a user interface, we ask several questions and gather the answers to make a user profile. At first, an author (storyteller) makes digital questionnaires to acquire familiarity and background knowledge of a user. The author also makes rules for the responses of virtual objects. A user's preference is captured when he/she selects one of control objects. TABLE V shows the list of user information from this process.

TABLE V
USER-CENTRIC CONTEXT IN OUR IMPLEMENTATION

User's context	Category
Familiarity	No, A little, Well-known
Background knowledge	Beginner, Intermediate, Expert
Preference	Priest Dosun, Young Monk

From the answers of the user, vrSensor multicasts personal information and interaction data in the form of user-centric context. The user-centric context is delivered to vrServices in a user interface and a virtual environment through a network. Then the vrServices decide the actions of virtual objects such as interactive viewer, character animation, events, etc. At this moment, comparing a user profile with rules given by storyteller, the user interface and events are re-configured. For instance, when a child uses the system, the user interface is simplified. Formula (7) is an example.

- If UserAns(Beginner) ∧ UserAns(No_background) → InteractiveViewer(map1) (7)

The position of virtual camera is calculated from following steps. Firstly, we calculate the 3D coordinate of the virtual camera based on the physical laws and geometric information in a virtual environment as described in section III. C. Secondly, we consider the story context. For example, we calculate the 3D camera position for viewing specific events. Thirdly, we apply user' familiarity and preference to change the position.

D. User Experience according to User's familiarity, background knowledge and Preference

Participants started interaction with control objects and the interactive viewer on ARTable. The participant watched control objects whose shape is easily found in our daily life on the ARTable. There were two kinds of dolls: priest and young monk. The participant could select one of dolls as his/her preference. If he selected the young monk, the screen view slightly moved up and down as human walking on the ground (Fig. 10. (a)). If the priest was selected, the 3D character moved toward front as it was riding on a cloud (Fig. 10. (b)).

(a) (b)

Fig. 10. Virtual characters reflecting preference

The participants input personal information on ARTable. In this process, we can get personal information and a user can learn how to utilize ARTable. We used this personal information, familiarity and background knowledge, to initialize the overall system. For example, if the participant was a young child (beginner), we showed the preview of the temple and played the narration to explain the legend. We also selected a simple sequence of events and a low interaction level because for a young child, it was difficult to follow the full story. Additionally, a child's interaction was restricted to specific positions that guide the presented story. If the participant was an adult (expert), we omitted preview and explanations. The StoryMap was selected based on background knowledge. Fig. 11 shows selection of a StoryMap according to a user's background knowledge. Fig. 12 shows the procedure for the personalization of ARTable.

(a) (b) (c)

Fig. 11. Selection of StoryMap according to user's background knowledge

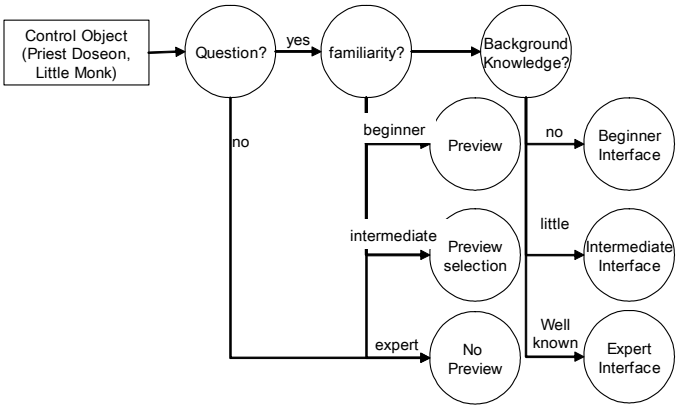


Fig. 12. User interaction in ARTable

The participant experienced personalized events in a virtual environment following the sequence illustrated in Fig. 13. The first position where a participant put his/her doll was the entrance of *Unju Temple* at StoryMap. The participant watched the big tree shaking by wind and listened to the sound of a bird moving around the tree. He/she could recognize the recommended position with a shining effect on the StoryMap. When the participant placed the doll at a specific position on the map, the virtual scene moved to that position. He could then select another position for further navigation.

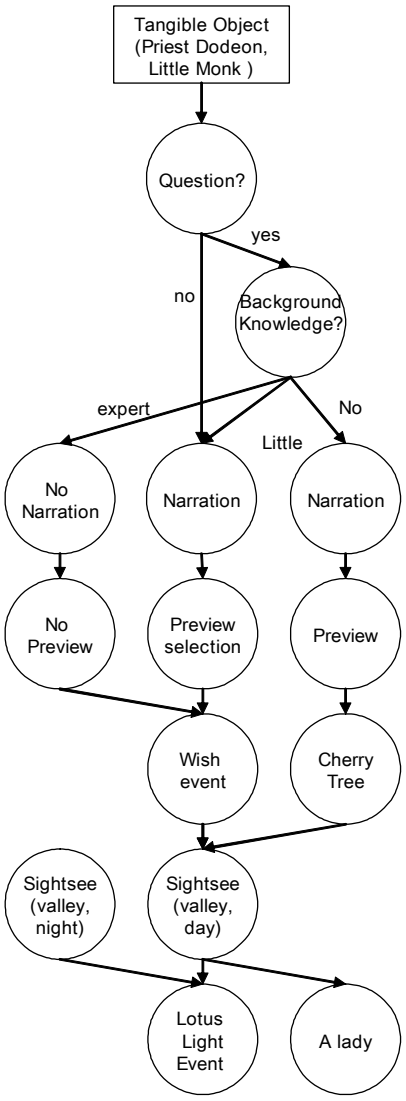


Fig. 13. User interaction and event sequence

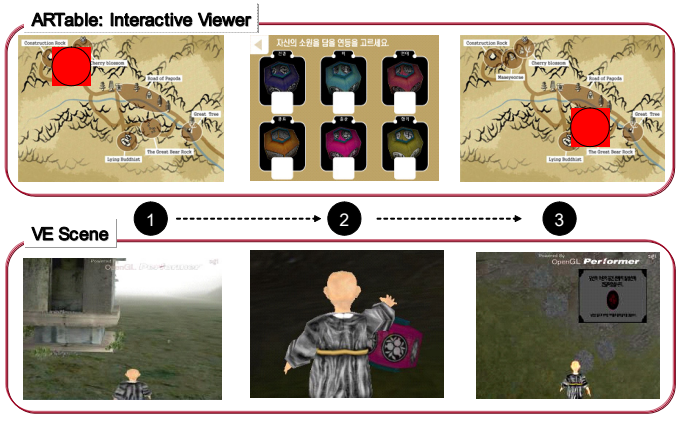


Fig. 14. A Lotus light: public's hope and wish. Upper figure shows events in ARTable and below figures are events in VE.

After watching scenery on the path, the participant could move to the location where cherry tree blossom stands. In this position, the participant could pray for his/her personal wishes. They can select one of wishes displayed on the ARTable; health, beauty, love, richness, childbirth, passing an examination. After selecting wishes, it became night. And the priest or young monk carried a lotus light which represented the wish. The participant experienced several events at the next location. When a participant arrived at the rock construction site, he could watch the whole sight of '*Unju Temple*' and a special event occurred. Finally, the participant reached the lying Buddhist. According to the legend, when thousands of Buddhist statues were made, the lying Buddhist stands up right. In our story, when 10 lotus lights hang on the tree, the lying Buddhist stands upright. Fig. 14 illustrates the event sequence for the lotus lights.

V. PERFORMANCE EVALUATION

A. Coordinate synchronization

We assigned a control object tracking resolution as 2mm since we need to remove jitters caused by light condition. The size of storymap displayed on the screen was about 50cm x 65 cm. However, the size of terrain was about 2300 x 1500 (unit). Thus when users move control objects about 1cm, virtual avatars move about 20m in VE. We applied linear interpolation from starting point to final point. Since a frame rate was about 10f/s, we calculated position in every 0.1 second. Fig. 15 shows interpolation result. From point 0 to 22, virtual character was moved 2.2 (unit).

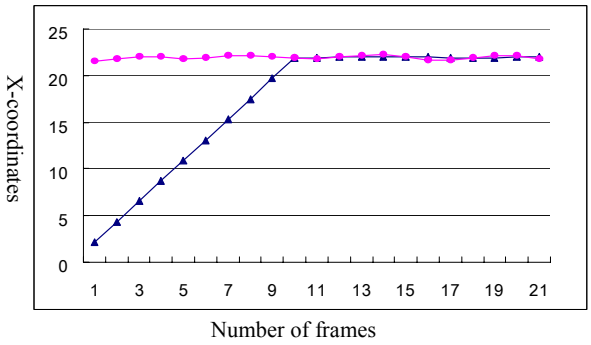


Fig. 15. Number of Frames before and after filtering

We faced a jittering problem which was caused by collision with terrain, camera tracking resolution, network delay and jitter. Since 3D terrain model was made with triangular mesh, there was a big angle change between mesh and mesh. Camera tracking algorithm also showed instability because of light condition. To reduce this effect, we implemented Gaussian filter which used linear interpolation. Since frame rate was 10f/s, it processed coordinate in every 0.1 second.

Table VI shows average and stand deviation when we applied Gaussian filter or not. For this measurement, we moved control object from arbitrary starting point. As results, we can see that height didn't show big changes. Its standard deviation was changed from 3.33 (unit) to 3.15. However, the gaze control result showed that we can reduce much instability, since the rotation error make big distance changes. For example, if gaze direction changes one degree and target

virtual object is located 200 (unit), it makes 0.1 x 200 = 20 (unit) errors.

TABLE VI
STANDARD DEVIATION BEFORE AND AFTER FILTERING

User's context	Height standard deviation	Viewing angle standard deviation
Without filter	3.33 (unit)	12.90 (deg)
With filter	3.15 (unit)	7.90 (deg)

B. Performance evaluation of Accessibility

The reaction of virtual objects is diversified according to rules given by developers. Let's assume that there are N sensors, M services, and each service has R actions. Then we can expect R x (N x M) reactions in a virtual environment in maximum. Moreover, if we apply joint rules, the number of possible reactions is increased. In addition, each virtual object can show different reactions on the same input since they may have different conditional contexts. However, the developers have to consider lots of cases when they design events.

Let's assume that G is a set of partitions with the equal number of elements. In the extreme case, we need N x M network connections to deliver information from N sensors to M services. The ePUIF separates services into several groups according to their active area. Thus, the number of connection lines can be reduced to N x (M/|G|) connections, where |G| is a number of groups. However, there was a network delay due to the publisher-subscriber protocol used.

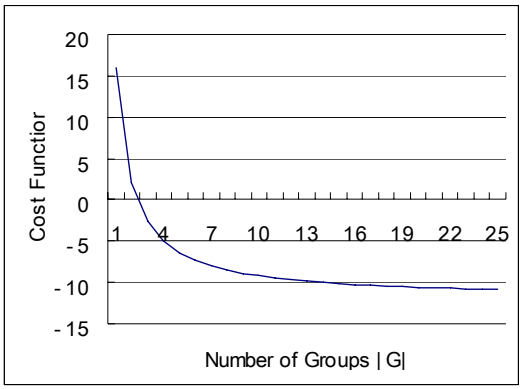


Fig. 16. Simulation of accessibility according to number of groups |G| where p is 8 bit and q is 20 bit

Equation (8) and (9) describe cost of the two cases respectively, where p is a packet size for publisher-subscriber protocol and q is a packet size for data transmission. Definitely, p is smaller than q.

$$(N \times M) \times q \tag{8}$$

$$(N \times M) \times p + (N \times {M}/{|G|}) \times (p+q) \tag{9}$$

From (8) and (9), we can get cost function (10). If the (10) holds, we can say that we enhanced the accessibility. The p was 8 bit because we used UDP and the q was about 20 bit in our experiment. Fig. 16 shows the simulation result that the cost function is less than zero when the number of groups |G| is greater than three.

$$(1+{1}/{|G|}) \times p - (1-{1}/{|G|}) \times q < 0 \tag{10}$$

C. Usability test

We made up questions to 48 persons who are researchers. Their interest research areas were ubiquitous computing, user interface, and virtual reality. We asked usability of ARTable and RMS in order to verity the enhancement of user interface. Fig. 17 is a result of the research. 37.5% (18) persons estimated 'very easy' to use ARTable. 50% (24) persons chose 'good' and 6 persons are 'difficult'. In this survey, 63.5% (30) persons did not satisfy enough. They pointed out that the user interface is needed to be improved at the commercial product level. After they experience personalized user interface, 39.5 % (19) persons put on premium on personalized responses of virtual objects that reflect their background knowledge and preference. The persons who chose 'good' instead of 'no meaning' were increased. This result shows that the proposed RMS could contribute to enhance users' experience.

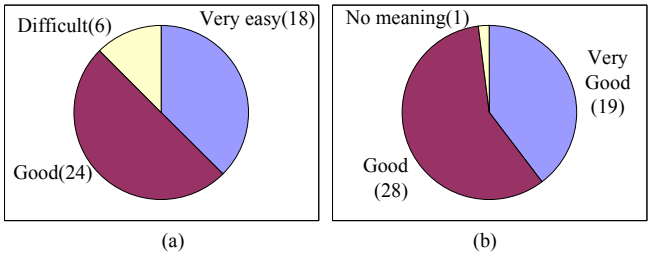


Fig. 17. Survey results. (a) ARTable usability test, (b) Personalized responses of RMS

VI. CONCLUSION

In this paper, we presented the Responsive Multimedia System (RMS), an interactive edutainment system. The RMS provides a personalized user interface by allowing a user to access and customize objects and to experience responses of them. In order to achieve this goal, we utilized ePUIF to share user-centric context such as user's background knowledge and preference in real and virtual environments. We enhanced user navigation by applying context-based navigation algorithm and presented intelligent responses of virtual objects based on condition-action rules. We implemented an interactive system, 'Dream of Mee-luck' to show the effectiveness of the proposed system. In this application, users can experience the personalized user interface, virtual character animation, and virtual events. However, because of matching much information from sensors and many rules inside services, complexity and uncertainty problems are caused. In the future, we have plans to study cognitive agents in order to enhance intelligent responses of virtual objects in the RMS.

REFERENCES

[1] M.Roussos, A.Johnson, J.Leigh, C.Barnes, C.Vasilakis, T.Moher, "The NICE project: Narrative, Immersive, Constructionist/Collaborative Environments for Learning in Virtual Reality." *ED-MEDIA/ED-TELECOM 97*, pp. 917-922, 1997

[2] A.Bobic, S.Intille, J.Davis, F.Baird, C.Pinhanez, L.Campbell, Y.Ivanov, A.Sch, A.Wilson, "The KidsRoom: A Perceptually-Based Interactive

and Immersive Story Environment,". *PRESENCE:Teleoperators and Virtual Environments*, pp.367-391, 1999.
[3] M. Matsushita, M.Iida, T.Ohguro, "Lumisight Table: A Face-to-Face Collaboration Support System That Optimizes Direction of Projected Information to Each Stakeholder" *Computer Supported Cooperative Work (CSCW)*, ACM Press, 2004
[4] P.Dietz, D.Leigh, "DiamondTouch: A Multi-User Thoch Technology," *User Interface Software and Technology (UIST)*, pp.219-226, 2001
[5] S.Jorda, "Sonigraphical Instrument: From FMOL to the reacTable*," *New Interfaces for Musical Expression (NIME)*, Montreal, Canada, pp.70-76, 2003.
[6] D.Rosenfeld, K,Perlin, M.Zawadzki, "Planar Minipulator Display," *SIGGRAPH Emerging Technologies*, San Diego, CA, 2003
[7] M.Walczak, M.McAllister, J.Segen, P.Kennard, "Dialog Table" *in Strangely Familiar: Design and Everyday Life exhibition*, Walker Art Center, Minneapolis, June 2003.
[8] Y.Lee, Sj.Oh, Y.Suh, S.Jang, W.Woo, "Enhanced Framework for a Personalized User Interface based on a Unified Context-aware Application Model for Virtual Environments," *IEICE TRANS. FUNDAMENTALS/COMMUN./ELECTRON./INF. & SYST*, 2007. (accepted)
[9] S.Jang, W.Woo, "ubi-UCAM: A Unified Context-Aware Application Model," *LNAI/LNCS (Context)*, vol. 2680, pp. 178-189, 2003.
[10] Y.Lee, D.Kim, Y.Lim, K.Kim, H.Kim, and W.Woo , "Dream of Mee-luck: Aspiration for a New Dawn," *LNCS (ICVS)*, vol.3805, pp. 282-285, 2005.
[11] Y. Park, W.Woo, "The ARTable: An AR-based Tangible User Interface System", *LNCS(Edutainment)*, vol.3942, pp.1198-1207, 2006.
[12] D.Hong, C.Shin, Sj.Oh, W.Woo, "A New Paradigm for User Interaction in Ubiquitous Computing Environment," *ISUVR*, pp. 41-44, 2006.
[13] S.Jang, W.Woo, "Unified Context Representing User-Centric Context: Who, Where, When, What, How and Why," *ubiComp workshop (ubiPCMM)*, 2005.

Youngho Lee received his BS in Dept. of Mathematics from Korea Advanced Institute of Science and Technology (KAIST), Daejon, Korea, in 1999 and M.S. degree in the Dept. of Information and communication from Gwangju Institute of Science and Technology (GIST), Gwanju, Korea, in 2001. He now is a Ph.D student in DIC, GIST since 2002. His research interests include Context-aware computing, HCI, virtual reality, artificial intelligence, interactive storytelling, etc.

Woontack Woo received his BS in Electronics Engineering from Kyungpook National University in 1989 and his MS in Electronics and Electrical Engineering from POSTECH in 1991. In 1998, he received his Ph.D. in Electrical Engineering-Systems from University of Southern California (USC). In 1999, as an invited Researcher, he joined Advanced Telecommunications Research (ATR), Kyoto, Japan. Since Feb. 2001, he has been with the Gwangju Institute of Science and Technology (GIST), where he is an Associate Professor in the Dept. of Information and Communications (DIC) and Director of Culture Technology Research Center (CTRC). His research interests include 3D computer vision and its applications including attentive AR and mediated reality, HCI, affective sensing and context-aware for ubiquitous computing, etc.