IEICE TRANSACTIONS

on Information and Systems

VOL.E89-D
NO.5
MAY 2006

A PUBLICATION OF THE INFORMATION AND SYSTEMS SOCIETY

The Institute of Electronics, Information and Communication Engineers
Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3 chome, Minato-ku, TOKYO, 105-0011 JAPAN

IEICE TRANS. INF. & SYST., VOL.E89–D, NO.5 MAY 2006

PAPER

Framework for Personalized User Interface by Sharing User-Centric Context between Real and Virtual Environments*

Seiie JANG†a) *and* Woontack WOO††, *Nonmembers*

SUMMARY In this paper, we propose a framework that provides users with personalized interfaces by sharing user-centric context between real and virtual environments. The proposed framework consists of ubi-UCAM for generating user's contexts, NAVER for managing virtual environment, and rv-Interface for linking ubi-UCAM with NAVER. Firstly, personalized interface helps users to concentrate on their tasks of interest by reducing burdensome menu selections according to user's context. In addition, user-adaptive contents based on user's preferences allow more pleasure personal experiences. Finally, personalized interface with context hand-over mechanism enables users to continuously interact with virtual environments, even if the users move around. According to experimental results, we expect that the proposed framework can play an important role for realizing user-centric VR applications by exploiting personalized interface that adapts to user-centric context.

key words: *personalized interface, user-centric context, context handover*

1. Introduction

VR applications have mainly been applied to limited domains e.g. psychotherapy [1], pilot-training [2], combat simulation [3], etc. However, recent advances in computer technologies and interactive devices have brought VR applications such as games [4], teleconferencing [5] and virtual museum [6] to everyday life. For the deployment of VR applications in daily life, developing easy and natural interfaces for the VR systems has been one of the core issues.

To provide users with easy and natural interfaces, context plays an important role in VR applications. Context enables applications to appropriately react to users without any explicit commands by providing contextual information such as location, identity of user, time of day, user's emotional state, focus of attention, and identities of nearby persons and objects. In the area of ubiquitous computing (ubiComp) [7], many research activities on context have been reported in [8]–[11].

On the other hand, context in VR has been biased toward virtual people or objects in virtual environments. For example, context is mainly exploited for giving life to virtual avatars in virtual environments [12], [13]. Context is often used to enable virtual objects to appropriately react to the changes in virtual environments [14], [15]. However, without considering each user's situation in real environments, the interaction between users and virtual environments frequently causes boredom since VR applications always provide the same contents and interfaces to all users [12]–[16]. In this case, users often get bored by the unchanging contents and interfaces during experiencing virtual environments again and again. In addition, it is not easy for users who have no experience in VR applications to quickly get accustomed to virtual environments. Consequently, it is necessary for VR applications to provide users with personalized interface that adapts itself to the user's context.

In this paper, we propose a framework that provides users with personalized interface by sharing user-centric context between real and virtual environments. The proposed framework consists of ubi-UCAM, NAVER, and rv-Interface. ubi-UCAM generates user's context (e.g. user identity, age, language, gestures, etc.) to execute context-based applications [17]. NAVER generates virtual environments and changes them according to scenarios which are scripted by administrators [18], [19]. rv-Interface transforms contexts or commands into events that influence the virtual environments, and vice versa.

The proposed framework guarantees the following features. Firstly, personalized interface changes user's menus according to user's contexts. This reduces bothersome menu-selecting operations and allows users to concentrate on virtual environments. Secondly, user-adaptive contents are selected according to user's preferences through personalized interface. This allows users to enjoy personal experiences on virtual environments. Finally, personalized interface hands over user's context in a personal device to a new virtual system where the user moves. This enables users to continuously interact with their virtual avatars and devices with increased mobility. According to experimental results, the proposed framework has verified that personalized interfaces adapting to user-centric context help users to concentrate more and enjoy virtual environments and to continuously interact with VR applications for increased mobility.

This paper is organized as follows: In Sect. 2, we explain the proposed framework and its components. The implementation and experimental results are explained in

Manuscript received December 20, 2004.
Manuscript revised September 22, 2005.

†The author is with Computer Mediated Communication Lab., Nagoya University, 464–8601, Japan.
††The author is with GIST U-VR Lab., Gwangju, 500–712, S. Korea.

*This research was supported in part by Culture Technology Research Center (CTRC) at GIST, and in part by Korea Institute of Science and Technology (KIST).

a) E-mail: jang@itc.nagoya-u.ac.jp

DOI: 10.1093/ietisy/e89–d.5.1694

Sects. 3 and 4, respectively. Finally, the conclusion and future work are discussed in Sect. 5.

2. Framework for Personalized User Interface by Sharing User-Centric Context between Real and Virtual Environments

The proposed framework aims at providing personalized user interface by sharing user-centric context between real and virtual environments. It consists of ubi-UCAM, NAVER, and rv-Interface as shown in Figs. 1 and 2. ubiSensor in ubi-UCAM senses changes or situations of users in real environments and delivers them to ubiServices. ubiService provides user's context to some VR services and reacts to changes in virtual environments. Also, VR services in NAVER are affected by user's context and deliver situations of the virtual environment to the real environment. rv-Interface is responsible for transforming contexts to events, and vice versa.

2.1 ubi-UCAM

ubi-UCAM [17], a unified context-aware application model, consists of ubiSensor and ubiService. Contexts in ubi-UCAM play a role for describing user-centric context that triggers a service without user's explicit commands. The context represents user-centric context in terms of 5W1H (Who, What, Where, When, How, and Why). This gives information such as "who is a user in a service environment?", "where is a user in a service environment?", "when is a user provided with a service?", "what is a user paying attention to?", "how is a user making an expression with gestures or actions?" and "why is a user going to trigger a service?" [20]. ubiSensor monitors the changes related to users or surroundings in real world and translates sensed data to low-level contexts. Then, it generates unified context in terms of 5W1H and multicasts the context to applications. ubiService generates high-level context by merging and analyzing low-level contexts. It collects low-level contexts from ubiSensors periodically and applies context fusion methods to each item of 5W1H. As a result of fusing contexts, integrated context filling up items of 5W1H is generated. Then, ubiService triggers a service only if a correspondence occurs between the integrated context and the user-specified conditional context. In case of the proposed framework, ubiService transfers integrated context to NAVER instead of exploiting it to trigger a service. Therefore, a ubiService corresponds to an object in virtual environment that is influenced by real-world contexts.

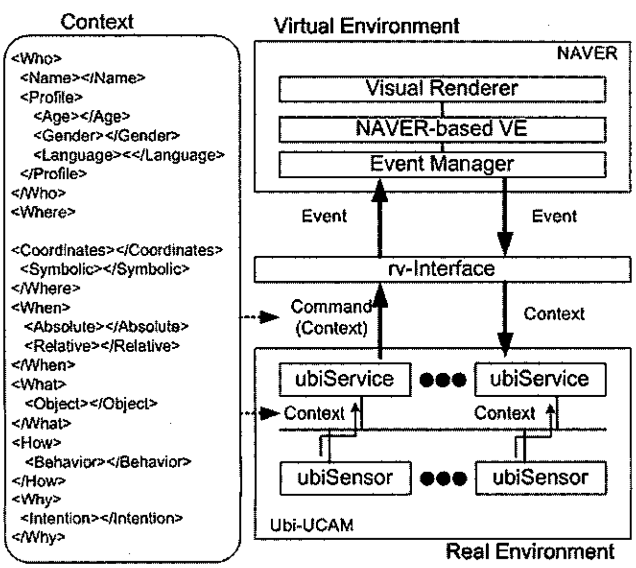


Fig. 1 Framework for personalized user interface by sharing user-centric context between real and virtual environments.

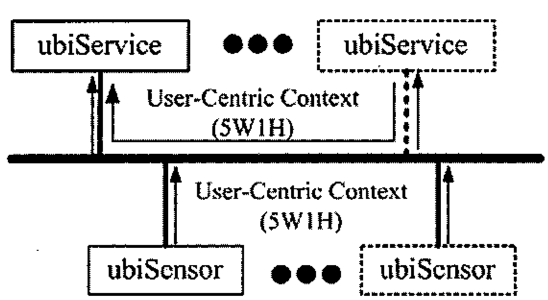


Fig. 2 ubi-UCAM; ubiSensor and ubiService.

2.2 NAVER

NAVER [18], [19], networked augmented virtual environment realtime, consists of NAVER world for generating a virtual environment, visual renderer for displaying virtual components in 2D or 3D, and event manager for controlling events in virtual environment as shown in Fig. 3 (a). NAVER-based VE consists of scenario manager, command manager, and interaction manager as shown in Fig. 3 (b). Scenario manager validates the XML script file supplied by the user, and transmits command lists to command manager. Command manager executes appropriate operations such as building a scene graph, setting environmental conditions and preparing network connections according to the

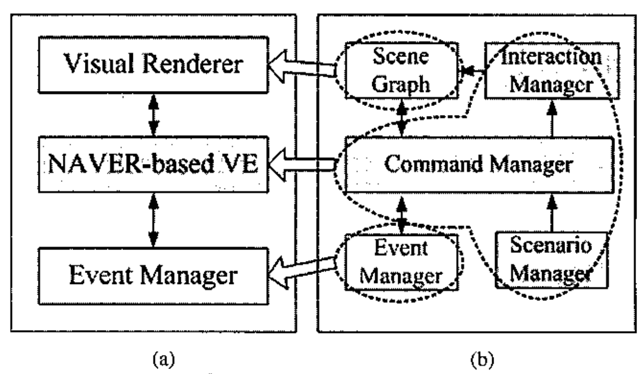


Fig. 3 NAVER. (a) Conceptual components of NAVER in proposed framework. (b) Components of NAVER.

action lists. Interaction manager reacts to users' explicit commands from input devices such as keyboard and mouse. Visual renderer displays components of virtual environment according to scene graph. Event manager controls changes in virtual environment according to users' input or administrator's scenario.

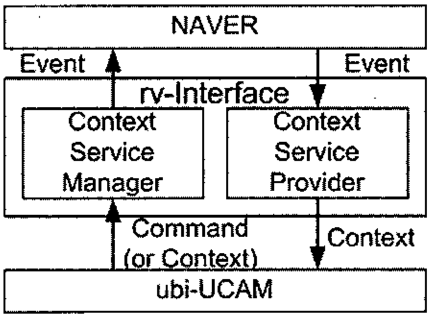


Fig. 4 rv-Interface.

```
<rv-Interface>
        <ubiService> ubiServiceID </ubiService>
                <Context>
                        <Who>Group Name</Who>
                        <What>Input Device</What>
                        <Where>Local Place</Where>
                        <When>Current Time</When>
                        <Why>Tour</Why>
                        <How>Gesture</How>
                </Context>
        <Context_Service_Manager>
                <Event_Type> Event </Event_Type>
                <vrService>VR ServiceID #1</vrService>
                <vrService>VR ServiceID #2</vrService>
                ...
                <vrService>VR ServiceID #N</vrService>
        </Context_Service_Manager>
</rv-Interface>
```

(a)

```
<rv-Interface>
        <vrService>VR ServiceID #1</vrService>
        <vrService>VR ServiceID #2</vrService>
                ...
        <vrService>VR ServiceID #N</vrService>
                <Event_Type> Event </Event_Type>
        <Context_Service_Provider>
                <Context>
                        <Who>Group Name</Who>
                        <What>Input Device</What>
                        <Where>Local Place</Where>
                        <When>Current Time</When>
                        <Why>Tour</Why>
                        <How>Gesture</How>
                </Context>
                <ubiService> ubiServiceID </ubiService>
        </Context_Service_Provider>
</rv-Interface>
```

(b)

Fig. 5 Example of XML-based rv-Interface. (a) Context Service Manager. (b) Context Service Provider.

2.3 rv-Interface

rv-Interface consists of Context Service Manager and Context Service Provider as shown in Fig. 4. Context Service Manager transforms contexts (representing users' context) or commands (influencing virtual environment) into events for event manager of NAVER. An event affects objects in virtual environment. In reverse, Context Service Provider converts events of NAVER to context for ubiServices in ubi-UCAM. The context from NAVER influences triggering of a service in the same way as a context from ubiSensors does.

rv-Interface is an XML-based interface describing relationships between user's contexts of real environments and commands of virtual environments, that are implemented in libraries such as NAVERLib, CAVELib, and VRJuggler. As shown in Fig. 5 (a), Context Service Manager multicasts events containing user's context in a real environment to VR services registered in rv-Interface. Equivalently, Context Service Provider delivers events of VR services in the form of user's context in a virtual environment to all ubiServices that are registered in rv-Interface as shown in Fig. 5 (b).

3. Implementation: Context-Based Virtual Heritage

To demonstrate the usefulness of personalized user interface by exploiting user-centric context, we have implemented a context-based virtual heritage system based on the proposed framework. The virtual system enables users to remotely explore cultural heritage instead of taking a long trip to Gyong-ju, the real site for cultural heritage of Shilla dynasty [21]. As shown in Fig. 6, the implemented system supports a guide and user groups. Thus, the system provides two kinds of services; one for a guide and the other for tourists.

Guide services allow a guide to steer tour groups in virtual heritage according to scenario describing tour schedule. To properly guide a tour, a guide is given contexts of user groups such as group names, ages and common interests. As shown in Fig. 7, the scene of virtual heritage is changed by guide's gestures such as movements of arms or legs.

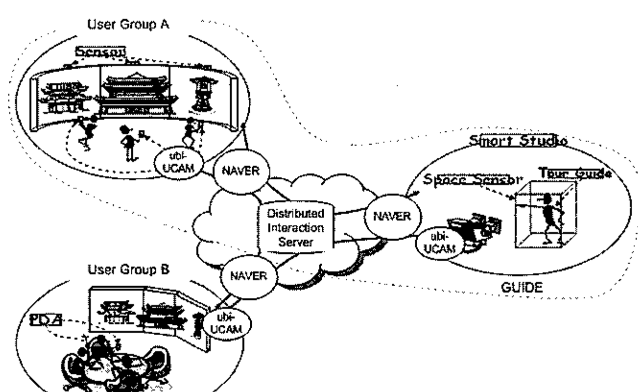


Fig. 6 Virtual heritage system for guide and tourist services.

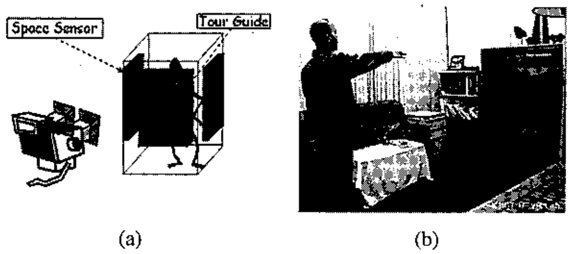


Fig. 7 Guide Service. (a) SpaceSensor. (b) Example of guide service.

Table 1 Example of context, command, and event for guide service.

SpaceSensor (Context)	Guide Service (Command)	NAVER (Event)
<Who> <Name Type= "Guide"> A </Name><Who> <How> <Behavior Type= "Hand"> Forward </Behavior> </How>	If (Who.Name!=null and Who.Type == "Guide") Command.Target = "Scene" If(How.Behavior.Hand =="Forward") Command.action = "Moving Forward"	Move(Scene, Forward, 1m)
<Who>...</Who> <How> <Behavior Type= "Hand"> Backward </Behavior> </How>	If(How.Behavior.Hand =="Backward") Command.action = "Moving Backward"	Move(Scene, Backward, 1m)
<Who>...</Who> <How> <Behavior Type= "Hand"> Left-UP </Behavior> </How>	If(How.Behavior.Hand =="Left-UP") Command.action = "Turn Left"	Rotate(Scene, Left, 15°)
<Who>...</Who> <How> <Behavior Type="Hand"> Right-UP </Behavior></How>	If(How.Behavior.Hand =="Right-UP") Command.action = "Turn Right"	Rotate(Scene, Right, 15°)
<Who>...</Who> <How> <Behavior Type="Hand"> Both-UP </Behavior> </How>	If(How.Behavior.Hand =="Both-UP") Command = "Move to Next Scene"	Move(Scene, Forward, NextScene)

SpaceSensor [22] based on ubiSensor senses guide's activities through a multi-view camera and generates guide's context. The guide service based on ubiService delivers guide's contexts for movement in virtual heritage by analyzing the guide's context. Table 1 shows how guide's gestures flows in the form of context, command, and event from ubi-UCAM to NAVER.

Tourists can explore virtual heritage with their PDAs. A ubiSensor in PDA provides user's identity and the group name. User's identification and group name are represented in terms of "Who" element. In addition, "Who" element contains "Profile" as sub-element, which presents user's age, sex and vernacular for context-based ubiServices such as language-adapted user interface and heritage-information service. Virtual GPS, group navigation, and virtual Memo are supported as other ubiServices. Virtual GPS displays tour-groups location information that is extracted from NAVER in virtual heritage. This is shown in Fig. 8 (a). By operating keypad on PDA, group navigation allows tour groups to go around particular sites without a guide as shown in Fig. 8 (b). Virtual Memo enables a tourist to attach messages to virtual sites on the fly in order to share messages with other tourists in other groups as shown in Fig. 8 (c). In this regard, a tourist can attach messages to virtual sites on the fly.

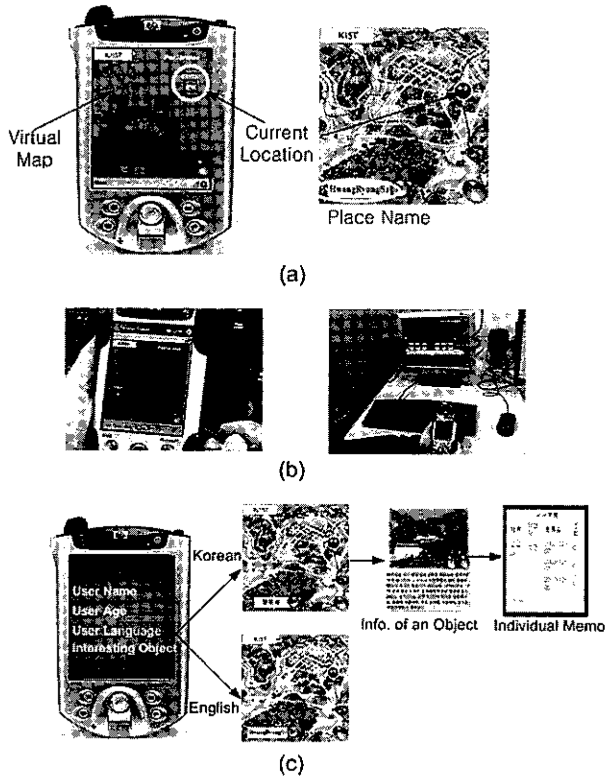


Fig. 8 Example of services for tourists. (a) Virtual GPS-showing current location of tour groups in virtual heritage map. (b) Group navigation-changing the location of a tour group through keypad of PDA. (c) Virtual Memo-displaying information embodied into specific sites.

As shown in Fig. 9, the implemented system provides virtual GPS for group navigation. In addition, the system supports virtual Memo by exploiting tourist's context contained in their PDAs. All keypad inputs that are generated by a group navigating virtual heritage are merged into one direction at a time. The direction is determined by a voting method which chooses the most voted value among users' inputs every 0.5 seconds. NAVER provides virtual GPS with the location of tour groups in the form of context whenever a tour group moves along in virtual heritage.

Location of all tour groups is initially synchronized with each other because a guide is only able to change the virtual heritage scene except for group navigation in a special area. However, tour groups have their view points to navigate virtual environment. Therefore, one group can see 3D avatars representing other groups. This location is represented with 2D or 3D "Coordinates" of "Where" element. In group navigation, for example, a tour group is able to meet 3D avatars of remote tour groups and to exchange messages with others by using virtual Memo service. Tour groups in distributed virtual systems are synchronized by sharing contexts that represent changes at remote nodes as shown in Fig. 10 (a). Although the virtual heritage of each tour group runs on heterogeneous system, it can be efficiently synchronized with others. For example, when a tour group navigates in a virtual environment, the event representing change in group location occurs. Other tour groups using different coordinates to display the group location accommodate themselves according to the shared context as shown in Fig. 10 (b).

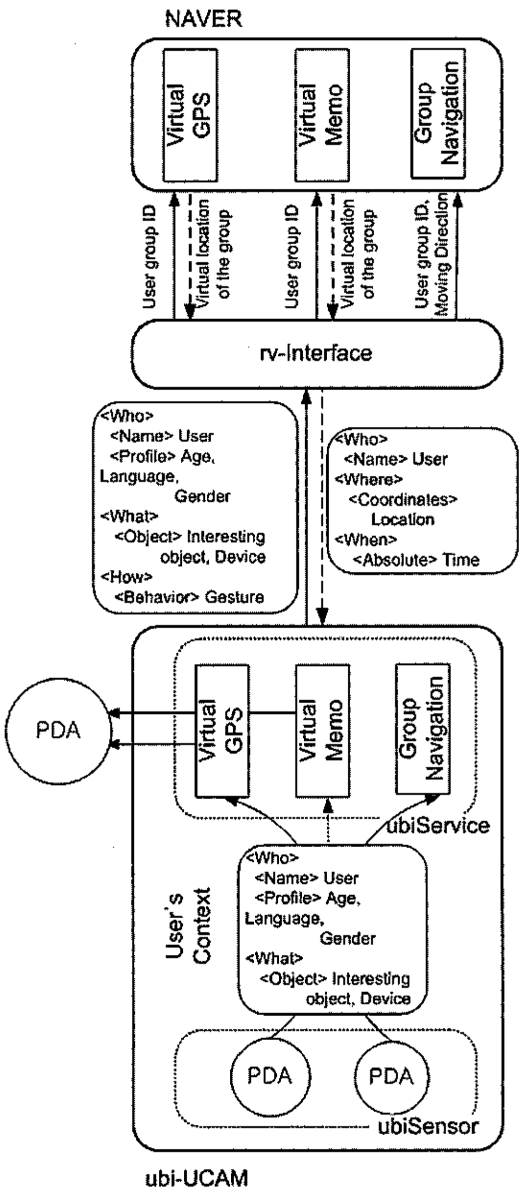
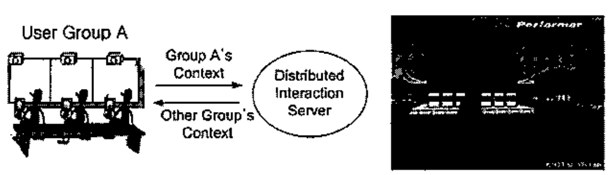


Fig. 9 Example of context, command, and event for tourist services.

4. Experimental Results

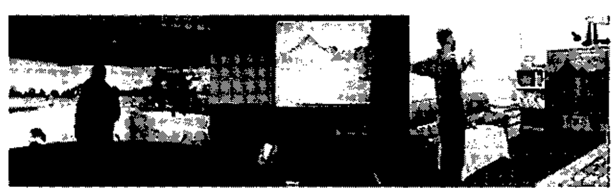
To evaluate the usefulness of the proposed framework, virtual heritage systems are distributed in three remote nodes. Virtual heritage systems for tourist services are installed in two nodes where PDA-based ubiSensor generates tourist's context, virtual GPS, group navigation, and virtual Memo runs simultaneously. However, one node provides a tour group with context-based services, and the other presents tour group with non context-based services. Two tour groups of five persons experienced virtual heritage systems for usability test. The volunteers consist of 6 males and 4 females and all are in 20–30 years-old range. Two males and two females speak English fluently and the rest speak Korean. Group 'A' where three persons select Korean as favorite language and two persons (one male and female) choose English, was given context-based services. Group 'B' was provided with non context-based services as shown in Fig. 11 (a) and (b). In addition, virtual heritage system for a guide includes SpaceSensor that detects guide's gestures. Also, this involves a guide service that makes all tour groups move in virtual heritage according to guide's gestures.

We compared the context-based services with non context-based services in order to determine how personalized interface affects interaction between a user and a virtual application. In case of the context-based services, group navigation provided users with information about sites in virtual heritage according to each user's age and vernacular such as Korean, English, Japanese, and Chinese. In addition, virtual Memo presented a user with personalized notes, selected from many messages embodied in objects, according to user's identity. In case of the non context-based services, group navigation provided the same information to all users in a group regardless of user's age and native language. If a user wanted to get different information, the user had to operate PDA menu to change information or select a language. Also, virtual Memo offered all messages augmented on objects in virtual heritage, and users could search in all messages.



(a) (b)

Fig. 10 Context-based synchronization among distributed virtual system. (a) Sharing group context among remote nodes. (b) Example of group A's view on group B.



(a) (b) (c)

Fig. 11 Example of virtual heritage systems. (a) Context-based services' node for group 'A'. (b) Non context-based services' node for group 'B'. (c) Guide service node.

Figure 12 compares average number of explicit commands used by each group to operate PDA. Users in group 'A' seldom manipulated services' menu to view information of heritage site in navigation except to change map size in virtual GPS and to select one of their own messages. However, users in group 'B' often operated services' menu. For example, users selected information or language whenever they moved in navigation. Users inputted group's name to find current location in virtual heritage. Also, users scrolled several times to find their messages. As a result, user's contexts reduce annoying operations while they interact with virtual environment by executing implicit commands to control services.

In addition, we questioned all users of both groups after they completed a tour in virtual heritage in order to verify the relationship between explicit commands and user's attention in virtual environment. All users were asked the same 10 questions to test their memory on the information of site they explored. They answered '*yes*' or '*no*' to these questions.

Figure 13 shows the distribution of the correct answers for both user groups. We did statistical evaluation since 10 users in two groups seemed to be small. We assumed that a normal population distribution (t-distribution) is approximately close to the correct answer distribution of two groups. In case of group A, the mean is 8.2 (μ_A), and the standard deviation is 1.30 (σ_A). In case of group B, the mean is 4.4 (μ_B) and the deviation is 1.81 (σ_B). Consider testing "H_0: $\mu_A = \mu_B$" against "H_a: $\mu_A > \mu_B$" by using the test statistic $T = (\overline{X} - \mu_B)/(S/\sqrt{n})$. Knowledge of the test statistic's distribution when H_0 is true allows us to construct a rejection region for which the error probability is controlled at the desired level. This means that P(error) = P(H_0 is rejected when it is true) = $P(T \geq t_{\alpha,n-1}$ when T has a t distribution with $n-1$ df) = α. In this case, the number of sample (n) is 5, $\overline{X} = \mu_A$, and $S = \sigma_A$. If these are applied to the test statistic, then T_A becomes 10.49. If we set the alpha value (α) to 0.01 and $t_{0.01,4}$ to 3.365, then T_A falls in the rejection region (10.49 > 3.365). This means that H_0 is false. Therefore, we can conclude that group 'A' has higher score than group 'B' at significance level 0.01 (99%). Also, this means that users in group 'A' give more correct answers than those in group 'B'. This is because group 'A' could attend to virtual heritage with convenient interface and personalized information services according to each context such as age and language.

To evaluate overheads involved in managing personalized interface for virtual environment, we measured the resource usage for each system: one supports context-based services and the other does not. Two systems have the same hardware and software which is as follows: Intel Xeon 2.4 GHz (Dual CPUs), 2 Gbytes (Memory), Wildcat7210 (Graphic card), and Window2000 (OS). Virtual environment is implemented by NAVER 1.0.2 based on OpenGL Performer 3.0.2. Context is generated by ubi-UCAM based on JAVA SDK 1.3. Figure 14 (a) and (b) show the mean usage of CPU and Memory on systems with non context-based and context-based service, respectively.

As shown in Fig. 14, system with context-based service consumes 0.9% more processing power and 12 MB more memory than the one with non context-based service. However, overheads on system with context-based service,

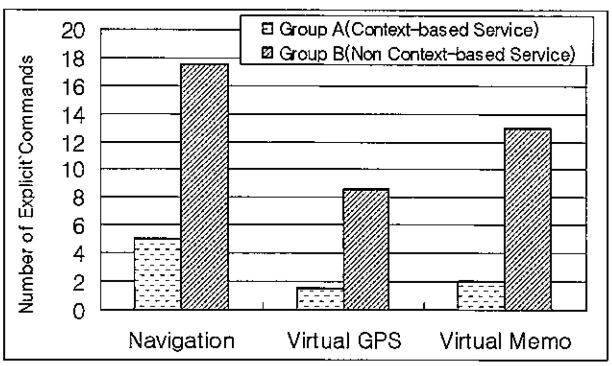


Fig. 12 Explicit-command comparison between context-based services and non context-based services. X axis is the average number of user inputs required to control services' menu. Y axis represents user services of virtual heritage.

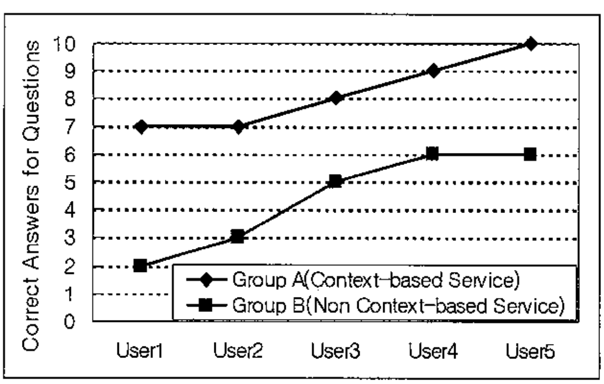


Fig. 13 Correct answers distribution of two groups to 10 questions to test user's memory on virtual heritage. (User 1–2 choose English, User 3–5 select Korean).

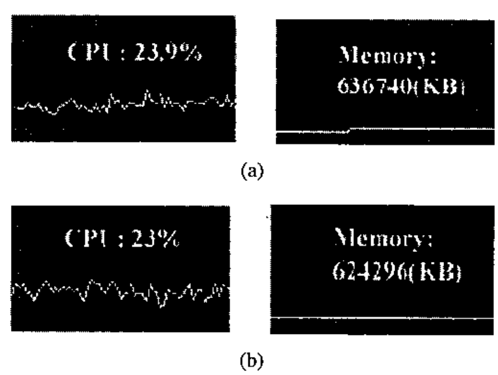
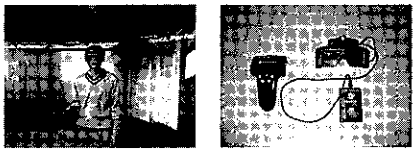


Fig. 14 Comparison of resource usage of virtual heritage nodes between context-based virtual heritage node and non context-based virtual heritage node. (a) Context-based virtual heritage node-CPU usage: 23.9%, Memory: 636 MB. (b) Non context-based virtual heritage node-CPU usage: 23%, Memory usage: 624 MB.

(a)



(b)

Fig. 15 Examples of heterogeneous distributed virtual systems. (a) NAVER-based virtual system at node 'A': a polarizing, filter-based passive stereo display, keyboard and mouse. (b) CAVELib-based virtual system at node 'B': a three channel, cluster-based active stereo display, wand and head tracker.

caused by context management, scarcely affect the system performance to provide users with services. Instead, exploiting context increased the system performance, considering the trade-off between the overhead and the reduction of explicit operations.

For heterogeneously distributed virtual environments, the management part of virtual environment is implemented by NAVER and CAVELib. NAVER, based on OpenGL Performer, renders 3D models by using Model Loading and Scene Graph. CAVELib represents the same 3D models in NAVER through an API, called pfCAVE, thus bridging CAVELib to OpenGL Performer. As shown in Fig. 15, one implemented system in NAVER is installed at node 'A'. At this node, a polarizing filter-based passive stereo display is an output device and a keyboard and mouse are default input devices. The other system, implemented in CAVELib, is installed at node 'B' which has a three-channel cluster-based active stereo display and a wand (IS900).

We did an experiment to evaluate an effect on context handover while moving among distributed remote sites. We measured the adjustment time and overhead to control avatars of two groups for navigation: One group handed over user's context in personal device to new node after moving. As a result, the group continued to handle the avatar representing the group that they come from. The other group operated a 3D avatar by using default input devices which was static to each node. The adjustment time is the period of time to get familiar with a new input device at the other node. The overhead is time interval between entering the other node and handling an avatar.

As shown in Table 2, adjustment time is 0–2 (sec) for PDA users and 30–60 (sec) for default input-device users while handling a group avatar with input devices. PDA users always controlled avatars without any adjustment time because the same avatar was handed over after user's movement by exploiting user's context in PDA. PDA delivers user's profile ("Who"), location ("Where"), and avatar ("What") to new node. In addition, users were already familiar with the input device. However, there was overhead for setting up network to connect PDA to ubi-UCAM located in a node where the user was located. Fortunately, this overhead will be removed if ad-hoc network techniques are applied for configuration during user's movement. Meanwhile, users of default input-device took some time to control avatars because they were not accustomed to using new avatar and input devices. Whenever they moved to other nodes, they took some time for adjustment. Meanwhile, there was no overhead because input devices were already connected with virtual system in a node where the user was located. Therefore, context enables users to continuously manipulate their avatars and input devices after moving from one virtual system to another. Therefore, the personalized interface through context handover may play an important role in seamlessly connecting user's experiences among distributed virtual environments.

Table 2 Comparison of adjusting time and overhead between context-based access and non context-based access.

	Adjusting time	Overhead
PDA users	0-2(sec)	5-9(sec)
Default-input-device users	30-60(sec)	0-2(sec)

5. Conclusion and Future Work

In this paper, we proposed a framework that provides personalized user interface by sharing user-centric context between real and virtual environments. By implementing context-based virtual heritage system according to the proposed framework, we have verified that personalized interfaces adapting to user-centric context help users to concentrate more on enjoyable virtual environments and to continuously interact with VR applications for increased mobility. As future work, we will extend the framework to seamlessly connect real environments with virtual environments by exchanging various contexts, i.e. users, environments, and computation.

Acknowledgments

We gratefully acknowledge the wise input provided by Seokmin Jung, Seokhee Lee, and Dongpyo Hong. In addition, we thank Dr. Heedong Ko and his lab members in KIST for helping us to exploit NAVER.

References

[1] J. Lapointe and J. Robert, "Using VR for efficient training of forestry machine operators," Education and Information Technologies, vol.5, no.4, pp.237–250, 2000.
[2] W. Kuan and C. San, "Constructivist physics learning in an immersive, multi-user hot air balloon simulation program (iHABS),"

Educators Program from the 30th Annual Conference on Computer Graphics and Interactive Techniques, pp.1–4, 2003.
[3] D. Hix, J.E. Swan Ii, J.L. Gabbard, M. McGee, J. Durbin, and T. King, "User-centric design and evaluation of a real-time battlefield visualization virtual environment," Proc. IEEE Virtual Reality, pp.96–103, 1999.
[4] M. Cavazza, S. Hartley, J. Lugrin, and M. Le Bras, "VR based entertainment & education: Alternative reality: A new platform for virtual reality art," Proc. ACM Symposium on Virtual Reality Software and Technology, pp.100–107, 2003.
[5] A. Sellen, B. Buxton, and J. Arnott, "Using spatial cues to improve videoconferencing," Proc. SIGCHI Conference on Human Factors in Computing Systems, pp.651–652, 1992.
[6] R. Wojciechowski, K. Walczak, M. White, and W. Cellary, "Production: Building virtual and augmented reality museum exhibitions," Proc. Ninth International Conference on 3D Web Technology, pp.135–144, 2004.
[7] M. Weiser, "The computer for the 21st century," Scientific American, vol.265, no.3, pp.66–75, 1991.
[8] R. Hull, P. Neaves, and J. Bedford-Roberts, "Towards situated computing," Proc. 1st International Symposium on Wearable Computers (ISWC'97), pp.146–153, 1997.
[9] A.K. Dey, "Context-aware computing: The CyberDesk project," Proc. AAAI 1998 Spring Symposium on Intelligent Environments (AAAI Technical Report SS-98-02), Palo Alto, CA, pp.51–54, AAAI Press, 1998.
[10] N. Davies, K. Mitchell, K. Cheverst, and G. Blair, "Developing a context-sensitive tour guide," 1st Workshop on Human Computer Interaction for Mobile Devices, 1998.
[11] A.K. Dey, Providing Architectural Support for Building Context-Aware Applications, Ph.D. Thesis, College of Computing, Georgia Institute of Technology, Dec. 2000.
[12] F. Li and M.L. Maher, "Representing virtual places," A Design Model for Metaphorical Design, ACADIA2000, 2000.
[13] M.L. Maher, N. Gu, and F. Li, "Visualisation and object design in virtual architecture," CAADRIA2001, ed. J.S. Gero, S. Chase, and M. Rosenman, pp.39–50, 2001.
[14] J.S. Gero, "Conceptual designing as a sequence of situated acts," in Artificial Intelligence in Structural Engineering, ed. I. Smith, pp.165–177, Springer, Berlin, 1998.
[15] R. Saunders and J.S. Gero, "A curious design agent," CAADRIA'01, ed. J.S. Gero, S. Chase, and M. Rosenman, pp.345–350, 2001.
[16] C. Demiralp, D.H. Laidlaw, C. Jackson, D. Keefe, and S. Zhang, "Subjective usefulness of CAVE and fish tank VR display systems for a scientific visualization application," IEEE Visualization Poster Compendium, Seattle, WA, 2003.
[17] S. Jang, and W. Woo, "ubi-UCAM: A unified context-aware application model," Lecture Note Artificial Intelligence, vol.2680, pp.178–189, 2003.
[18] H. Ko, M. Park, and H. Lee, "Conceptual framework of tangible space initiative and its application scenario to heritage alive!," Virtual Systems and MultiMedia (VSMM2002), pp.1001–1007, 2002.
[19] P. Changhoon, H. Ko, and T. Kim, "NAVER: Networked and augmented virtual environment architecture; design and implementation of VR framework for Gyeongju VR theater," Computers & Graphics, vol.27, pp.223–230, 2003.
[20] S. Jang and W. Woo, "Unified context representing user-centric context: Who, where, when, what, how and why," Proc. International Workshop ubiPCMM05, Tokoyo, Sept. 2005, CEUR Workshop Proceedings, ISSN 1613-0073, online CEUR-WS.org/vol-149/
[21] H.-J. Lee, J. Kim, H. Ahn, S.C. Ahn, I.-J. Kim, H.-G. Kim, and H. Ko, "VR experience design in tangible space: Heritage alive!," IEA2003.
[22] D. Hong and W. Woo, "A 3D vision-based ambient user interface," International Journal of Human Computer Interaction, 2006. (Under Reviewing)

Seiie Jang received his B.S. degree in CS from Sogang University, Seoul, Korea, in 1997 and M.S. degree in CS from Sogang, Seoul, Korea, in 1999. He received his Ph.D. in Information and Communication from Gwangju Institute of Science and Technology (GIST), Gwangju, Korea, in 2005. He currently works as post doctor in Nagoya University. His research interests include user-centric context modeling, context management in personal space and place, applications for ubiquitous computing, HCI, etc.

Woontack Woo received his B.S. degree in EE from Kyungpook National University, Daegu, Korea, in 1989 and M.S. degree in EE from POSTECH, Pohang, Korea, in 1991. He received his Ph.D. in EE-Systems from University of Southern California, Los Angeles, USA. During 1999-2001, as an invited researcher, he worked for ATR, Kyoto, Japan. In 2001, as an Assistant Professor, he joined Gwangju Institute of Science and Technology (GIST), Gwangju, Korea and now at GIST he is leading U-VR Lab. 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.