CIVE: Context-based Interactive System for Distributed Virtual Environments

Seiie Jang, Youngho Lee, and Woontack Woo
GIST U-VR Lab., Gwangju 500-712, S. Korea
{jangsei,ylee,wwoo}@gist.ac.kr

Abstract

In this paper, we propose CIVE, context-based interactive system for distributed virtual environments that delivers contexts from real world to virtual environment and vice versa. The proposed system consists of ubi-UCAM for generating user's contexts, NAVER for managing virtual environment, and Interface for linking ubi-UCAM with NAVER. The connection between real and virtual world through context is beneficial in the following ways. Firstly, CIVE provides a personalized user interface for virtual environment according to user profile such as identity, age and vernacular. Secondly, CIVE guarantees adaptive access for translating all input signals to context. This enables a user to exploit unencumbered input devices for controlling a shared object in virtual environment while he moves from one virtual system to another. Finally, it provides a mechanism for synchronizing distributed virtual systems that share the context representing changes at remote nodes. The context reduces the inconsistency of representing the same data among heterogeneous systems. Therefore, CIVE plays an important role in implementing VR applications such as teleconference, games and entertainment systems.

Keywords: context-based interaction, personalized user interface, adaptive access, synchronization

1. Introduction

VR technology has mainly been used for special applications such as psychotherapy, pilot-training or combat simulation. However, recent advancements in computer technology and realistic contents have brought VR to our daily life. Examples include teleconference applications, VR games and virtual museum. In this regard, providing natural user interactions in virtual environments has been one of the core issues. For example, collaborative medical workbench (CMW) allows medical students to operate on a virtual heart with pen-type tools [1]. Dragon provides users with conventional interface such as speech recognition, joystick and dataGlove for virtual combat [2]. N.I.C.E allows children to communicate with each other in CAVE through 3D interface such as wand and head tracker [3]. Heritage Alive provides audience with virtual heritage to realistically navigate a historic site [4]. These interaction researches have converged into 3D interface or immersive display. However, in order to provide seamless connection between real and virtual worlds, dynamic management of virtual environment according to user contexts is as important as 3D I/O devices.

In this paper, we propose CIVE, context-based interactive system for distributed virtual environments, which delivers contexts from real world to virtual environment and vice versa. The proposed system consists of ubi-UCAM[5], NAVER[4], and Interface. Ubi-UCAM generates contexts such as user identity, age, language, and gestures and supports context-based applications. NAVER provides user with virtual heritage and senses user's activities in cyber environment. Interface transforms the context generated by ubi-UCAM into event for NAVER, and vice versa. CIVE provides a personalized user interface for virtual environment by exploiting user contexts. They help users to concentrate on virtual environment by providing implicit instead of explicit commands. Secondly, CIVE guarantees adaptive access for translating all input signals to context. This enables a user to exploit unencumbered input devices for controlling a shared object in virtual environment while he moves from one virtual system to another. Finally, it provides a mechanism for synchronizing distributed virtual systems that share context representing changes at remote nodes. The context reduces the inconsistency of representing the same data among heterogeneous systems.

This paper is organized as follows: In section 2, we explain the CIVE architecture and its components. The implementation and experimental results are explained in section 3 and 4, respectively. Finally, the conclusion and future works are discussed in section 5.

This work is supported by KIST in Korea

2. CIVE: Context-based Interactive System for Distributed Virtual Environments

CIVE is a system that connects real world with virtual world by sharing users' contexts. It consists of ubi-UCAM for generating users' contexts, NAVER for managing virtual environments, and Interface for linking ubi-UCAM with NAVER as shown in Figure 1. Interface transforms contexts or commands into events that influence virtual environment. It also converts events into contexts for context-based services in real environment.

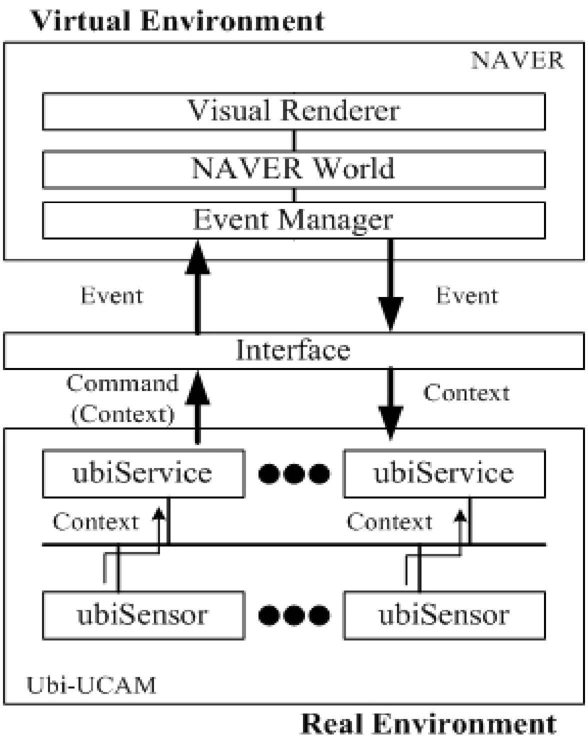


Fig. 1. Architecture of CIVE

2.1. ubi-UCAM

Ubi-UCAM[5], a unified context-aware application model, consists of ubiSensors and ubiServices as shown in Figure 2. Contexts in ubi-UCAM play a role for describing user's situation that may trigger a service without user's explicit commands. Contexts represent user's situation in forms of 5W1H (Who, What, Where, When, Why, and How), so that they can simultaneously be shared among applications. UbiSensor, the sensor of ubi-UCAM, monitors the changes of users or environments in real world and translates sensed signals to low-level contexts. Then, it generates unified context in terms of 5W1H and multicasts the context to applications. UbiService, the application of ubi-UCAM, generates high-level context by merging and analyzing low-level contexts. It periodically collects low-level contexts from ubiSensors 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 automatically triggers a service if correspondence occurs between the integrated context and a user-specified conditional context. In case of the CIVE, ubiService, providing context to handle an object in virtual environment, transfers integrated context to NAVER instead of directly triggering a service. Therefore, ubiService will be implemented as many as the number of objects in virtual environment that are influenced by contexts in real world.

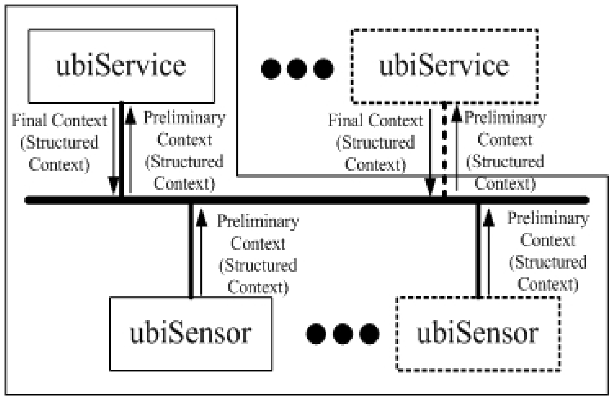


Fig. 2. ubi-UCAM architecture; ubiSensor and ubiService

2.2. NAVER

NAVER[4], networked augmented virtual environment realtime, consists of NAVER world for generating virtual environment, visual renderer for displaying virtual components in 2D or 3D , and event manager for controlling events in virtual environment as shown in Figure 3(a).

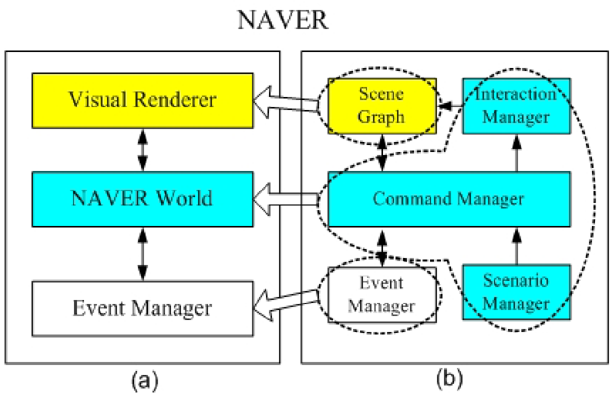


Fig.3. NAVER architecture; (a) conceptual components of NAVER in CIVE (b) components of NAVER

NAVER world consists of scenario manager, command manager, and interaction manager as shown in Figure 3(b). Scenario manager validates the XML script file supplied by the user, and transmits verified 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 action lists. Interaction manager reacts to users' explicit commands from input devices such as a 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 scenario.

2.3. Interface

Interface consists of context service manager and context service provider as shown in Figure 4. Context service manager transforms contexts (representing users' situations) or commands (influencing virtual environment) into event for event manager of NAVER. The event affects objects in virtual environment. In reverse, context service provider converts the event of NAVER to context for ubiServices in ubi-UCAM. The context from NAVER has influence triggering a service as if a context comes from an ubiSensor.

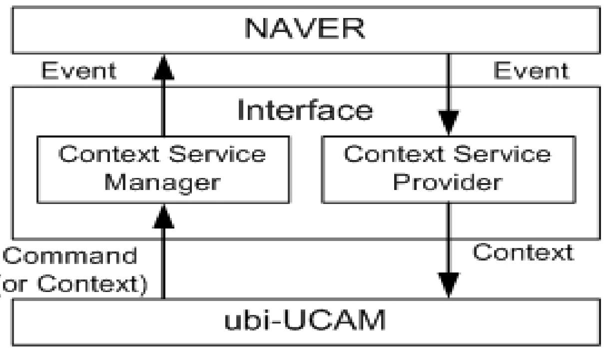


Fig.4. Interface architecture

3. Implementation: Context-based Virtual Heritage

To demonstrate usefulness of context for interaction between real and virtual environments, we have implemented context-based virtual heritage system based on the proposed CIVE. The virtual system enables users to remotely explore cultural heritage instead of taking a long trip to Kyong-ju, the real site for cultural heritage of Shilla dynasty. As shown in Figure 5, it is separated into a guide and user groups that share virtual heritage. Thus, the system provides two kinds of services; one for a guide and the other for general users.

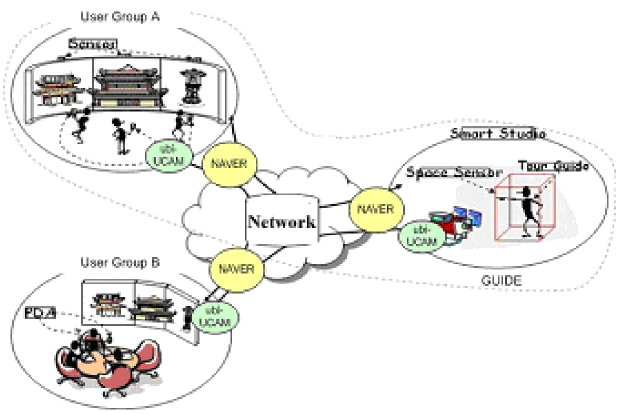
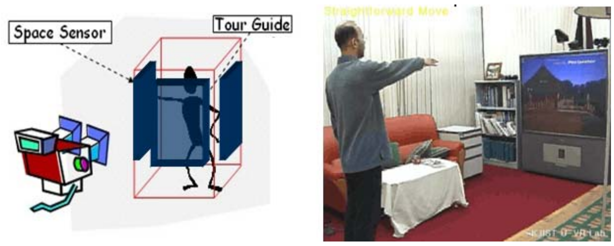


Fig.5. Context-based virtual heritage system

Guide services allow a guide to steer user groups in virtual heritage according to scenario describing tour schedule. A guide is given contexts of user groups, such as group names, ages and common interests, in order to properly guide a tour. As shown in Figure 6, the scene of virtual heritage is changed by guide's gestures such as movements of arms or legs. SpaceSensor based on ubiSensor senses guide's activities through 3D camera and generates guide's context [6]. The guide service based on ubiService generates commands for movement in virtual heritage by analyzing the guide's context. Table 1 shows how to deliver guide's gestures in the form of context, command, or event from ubi-UCAM to NAVER.



(a) (b)

Fig.6. Guide Service; (a) SpaceSensor (b) example of guide service

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

Space Sensor (Contexts)	Guide Service (Commands)	NAVER (Events)
Hands Forward	Moving Forward	Go 1m
Hands Backward	Moving Backward	Back 1m
Left-Hand UP	Turn Left	Left rotation 15°
Right-Hand Up	Turn Right	Right rotation 15°
Both Hand Up	Move to Next Scene	Scene Change

Users can explore virtual heritage with their own PDA. An ubiSensor in PDA provides a user profile containing identity, age, sex, and vernacular to context-based ubiServices such as language-adapted user interface and heritage-information service. In addition, there are other ubiServices such as virtual GPS, group navigation, and virtual memo. Virtual GPS displays current location of user groups, extracted from NAVER, in virtual heritage. This is shown in Figure 7(a). Group navigation allows a user group, operating keypad on PDA, to go around particular sites without a guide as shown in Figure 7(b). Virtual memo [7] enables a user to attach messages on virtual sites on the fly in order to share messages with users in other groups as shown in Figure 7(c).

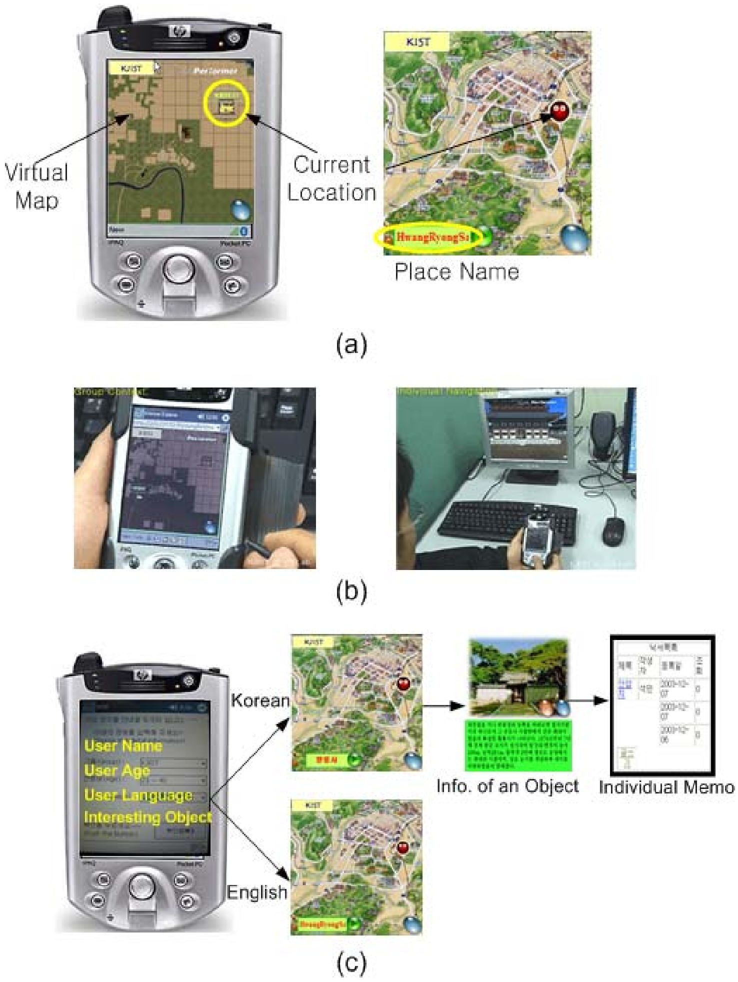


Fig.7. Example of user services; (a) virtual GPS: shows current location of user groups in virtual heritage map (b) group navigation: changes the location of a user group through keypad of PDA (c) virtual Memo: displays information embodied into specific sites or objects in virtual heritage, and augments user's note through BBS (bulletin board system).

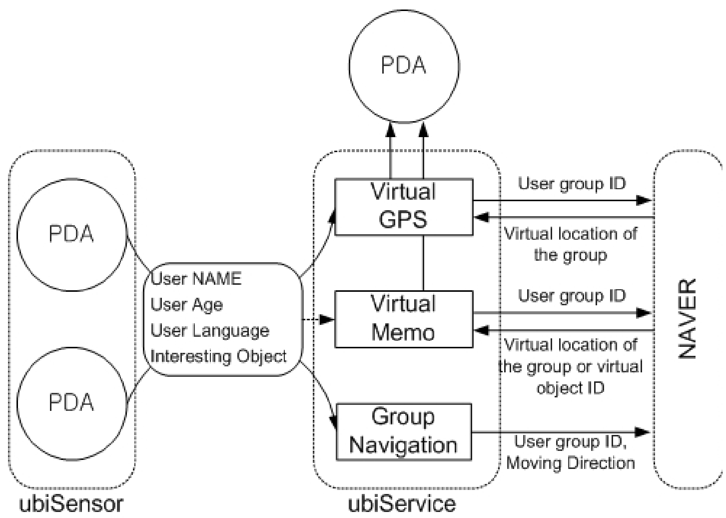


Fig.8. Example of context, command, and event for user services.

As shown in Figure 8, CIVE provides virtual GPS, group navigation and virtual Memo with users' context, contained in their PDA. In addition, keypad inputs that users in a group generate to navigate virtual heritage are merged into one direction at a time. The direction is determined by voting method which chooses the most selected value among user's inputs every 0.5 seconds.

NAVER provides virtual GPS with the location of user groups, in terms of context, whenever each user group moves in virtual heritage.

Location of all user groups is initially synchronized with each other because only the guide changes the virtual heritage scene except during group navigation in special area. However, each user group has its own view point to navigate virtual environment. Therefore, one group can see 3D avatars representing other user groups. For example, in group navigation, a user group is able to meet 3D avatars of remote user groups, and exchanges messages with others by using virtual Memo service. User groups in distributed virtual systems are synchronized by sharing contexts that represent changes at remote nodes, as shown in Figure 9(a). Although the virtual heritage of each user group runs on heterogeneous system, it can be efficiently synchronized with others. For example, when a user group navigates in a virtual environment, the event representing change in group location occurs. Other user groups, using different coordinates to display the group location, accommodate themselves according to the shared context as shown in Figure 9(b).

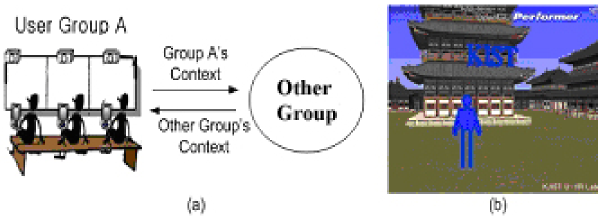


Fig.9. Context-based synchronization among distributed virtual system; (a) synchronization mechanism (b) example of group A's view of group B

4. Experiments

To evaluate the usefulness of CIVE, virtual heritage systems are distributed in three remote nodes. Virtual heritage systems for user services are installed in two nodes where PDA based ubiSensor generating a user context, virtual GPS, group navigation, and virtual Memo are running together. However, one node provides a user group with context-based services, and the other presents a user group with non context-based services. Two user groups, having 5 persons per each group, experienced virtual heritage systems for usability test. Group A was given context-based services and group B was provided non context-based services as shown in Figure 10 (a) and (b), respectively. In addition, virtual heritage system for a guide includes SpaceSensor, detecting guide's gestures, and a guide service, moving all user groups in virtual heritage according to guide's gestures.

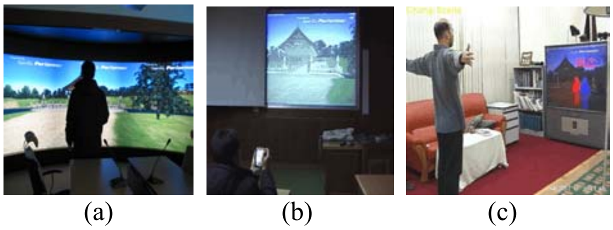


Fig.10. Three virtual heritage systems; (a) the context-based services' node for group A (b) the non context-based services' node for group B (c) the guide service node

We compared the context-based services with non context-based services in order to determine how user contexts affect interaction between a user and virtual environment. 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 a user identity. In case of the non context-based services, group navigation provided same information regardless of the user's age and native language. If a user wanted to get different information, the user had to operate menu of PDA to change information or select a language. Also, virtual Memo offered all messages augmented on objects in virtual heritage, and users could search for all messages.

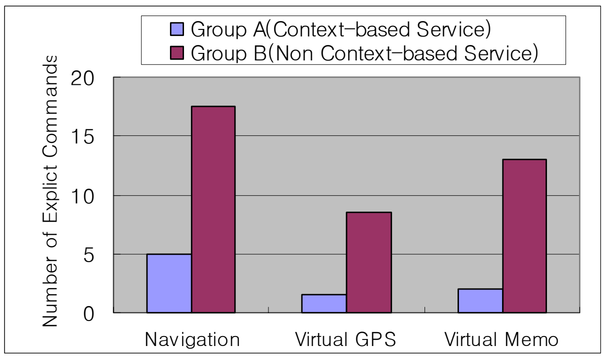


Fig.11. Comparison of number of explicit commands between context-based services and non context-based services. X axis is the average number of user inputs to control services' menu. Y axis represents user services of virtual heritage.

Figure 11 compares average number of explicit commands used by each user group to operate PDA. Users in group A seldom controlled services' menu except to view information of heritage site in navigation, changing map size in virtual GPS, and selecting 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 own messages. As a result, user contexts reduce annoying operations for users while they interact with virtual environment by executing implicit commands to control services.

Also, we questioned all users of both groups after they terminated a tour in virtual heritage in order to verify the relationship between explicit commands and user's attention on virtual environment. All users were asked same 10 questionnaires about virtual heritage they explored. They answered "yes" or "no" to these questions. Figure 12 shows the distribution of the correct answers for both user groups. Users in group A replied more correct answers than those in group B. This was because group A could attend to virtual heritage with convenient interface and personalized information services according to each user context such as age or language.

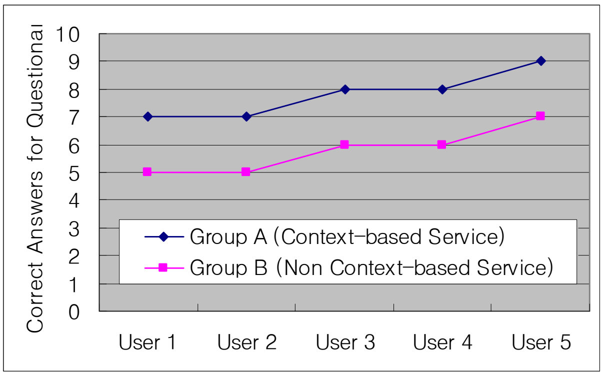
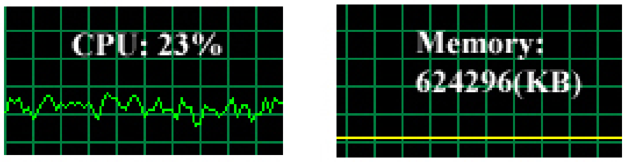


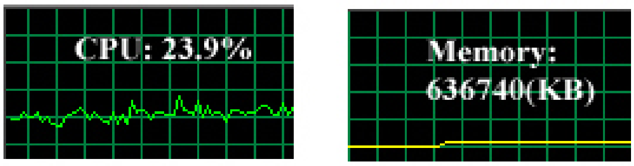
Fig.12. The distribution of correct answers of two user groups to 10 questions about virtual heritage.

As a result of Figure 11 and 12, contexts help users to interact with virtual environment by reducing bothersome tasks and providing personalized services according to each user. Therefore, context, causing objects or virtual environment to change according to each user, may play an important role for handily connecting users in real world to virtual world.

To evaluate overheads that require managing context for virtual environment, we measured the resource usage between two systems: one supports context-based services and the other does not. Two systems have the same hardware and software as follows: Intel Xeon 2.4 GHz (Dual CPUs), 2Gbytes (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 13(a) and (b) show the mean usage of CPU and Memory in non context-based service system and context-based, respectively.



(a)



(b)

Fig.13. Resource usage of virtual environment system with context and not; (a) Non context-based service system – CPU usage: 23%, Memory usage: 624MB (b) Context-based service system- CPU usage: 23.9%, Memory: 636MB.

As shown in Figure 13, context-based service system consumes 0.9% more processing and 12 MB more memory than non context-based service system. However, overheads in context-based service system, caused by managing context, scarcely affect the system performance for providing users with services. Instead, exploiting context may increase the system performance, considering the trade-off between the overhead and the reduction of explicit operations as shown in Figure 10.

For heterogeneous distributed virtual environments, a part of CIVE, which manages 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 with NAVER through an API, called pfCAVE, bridging CAVELib to OpenGL Performer. As shown in Figure 14, one CIVE based on NAVER is installed in Node A, where a polarizing, filter-based passive stereo display is an output device and a keyboard and mouse are input devices as default. The other CIVE, based on CAVELib, is installed in Node B which has a three channel, cluster-based active stereo display and a wand (IS900). Two nodes are connected with 155Mbps network (KOREN) and support 11Mbps wireless network for PDA access.

(a)

(b)

Fig.14. Example of heterogeneous distributed virtual systems; (a) NAVER-based virtual system in Node A: a polarizing, filter-based passive stereo display, keyboard and mouse (b) CAVELib-based virtual system in Node B: a three channel, cluster-based active stereo display, wand and head tracker

We did an experiment to evaluate an effect on seamless connection between users and virtual environments that were distributed in remote sites. We measured the adjusting time and overhead to control avatars of two user groups for navigation: one group, controlling a 3D avatar with their own PDA, moved to another node where they continued to handle the avatar representing a group that they come from. The other group operated a 3D avatar by using default input devices such as keyboard, mouse, or wand after moving from one node to another. The adjusting time is the period of time to get familiar with a new input device in the other node. The overhead is time between entering the other node and handling an avatar.

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)	7-12(sec)

As shown in Table 2, adjusting time took 0-2 (sec) for PDA users and 30-60 (sec) for default-input-device users to be used to handling a group avatar by input devices. PDA users always controlled avatars without any adjusting time because they were already practiced in the input device. Such a short adjusting time was guaranteed by transforming the signal of input device into context that works in different virtual systems. However, there was overhead that users set network to connect PDA to ubi-UCAM located in a node where they had been. Fortunately, this overhead will be removed if ad-hoc network techniques are applied to the configuration during user's movement. In case of the default-input-device users, it took some time to control avatar because they were not accustomed to using new input devices. Whenever they moved to other nodes, they took some time to adjust. Meanwhile, there was no overhead because input devices were already connected with virtual system in a node where they had been. Therefore, context enables a user to exploit unencumbered input devices for controlling a shared object in virtual environment while he moves with his device from one virtual system to another. It may play an important role in seamlessly connecting users with distributed virtual environments.

Finally, we made an experiment in contexts having influence on synchronizing heterogeneous distributed virtual environment systems. We measured the coordinate errors and time delay of an avatar between two heterogeneous systems whenever the group avatar moved in virtual heritage. Since CAVELib and NAVER in their own ways move avatars, each has different coordinates. CAVELib moves an avatar by comparative coordinates that represent a difference between absolute coordinates of an avatar moving before and after. NAVER uses absolute coordinates to change the location of an avatar. We compared two synchronization methods. Context-based synchronization is a way to share context representing absolute coordinates of avatar between two systems. For example, NAVER sends absolute coordinates to CAVELib, and vice versa. Non context-based synchronization is a way to exchange its own coordinates with each other. For example, NAVER sends absolute coordinates to CAVELib and NAVER receives comparative coordinates from CAVELib.

Table 3. Comparison between context-based synchronization mechanism and non context-based synchronization mechanism

	Coordinate error	Time delay
Context-based Synchronization	2-3%	20-23(ms)
Non context-based Synchronization	1-2%	10-15(ms)

The context-based synchronization requires two translating steps at each virtual system. One step is that a system changes its coordinates to absolute coordinates in order to send them to another system. The other step is that a system transforms the received coordinates into its own coordinates. However, the non context-based synchronization demands just one step because each system sends its own coordinates to another and it translates the received coordinates according to its own synchronization. Therefore, the context-based synchronization took higher coordinate error and longer time delay than non context-based synchronization, as shown in Table 3. However, in case of the heterogeneous distributed systems consisting of N nodes, the context-based synchronization may be better than the non context-based. The context-based synchronization requires just N (1: for sending contexts, N-1: for receiving contexts) translations without any information about other systems. The non context-based synchronization demands N-1 translation for converting received coordinates from N-1 nodes. It also manages N-1 piece of information about synchronization of other nodes. Therefore, it totally takes 2(N-1) translation steps to synchronize distributed systems consisting of N heterogeneous nodes. Besides, to add a new node to N heterogeneous nodes, context-based synchronization will work only if a part, which manages context, is included in the new node. In case of the non context-based synchronization, all synchronization mechanisms of N nodes must be updated to be consistent with a new node. As a result, context may play an important role in efficiently synchronizing distributed virtual environments because it reduces the inconsistency of representing the same data among heterogeneous systems.

5. Discussion & Future Works

In this paper, we proposed CIVE, context-based interactive system for distributed virtual environment that delivers user's contexts from real world to virtual environment. To demonstrate the usefulness of context for interaction between real and virtual environment, we have implemented context-based virtual heritage system based on the proposed CIVE. As future works, we will prove how context helps users to concentrate on virtual environment by reducing explicit user commands. In addition, we plan to evaluate, in detail, the effectiveness of context in synchronizing heterogeneous distributed virtual systems.

Acknowledgements the authors gratefully acknowledge the wise input provided by Seokmin Jung, Seokhee Lee, Dongpyo Hong. We thank Dr. Heedong Ko and his lab members for help us to use NAVER.

References

1. G.Goebbels, V.Lalioti, and M.Göbel, "Design and Evaluation of Team Work in Distributed Collaborative Virtual Environments," In Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST 2003 sponsored by ACM SIGCHI & ACM SIGGRAPH), Osaka University, Osaka, Japan, 2003.

2. B.Leibe, T.Starner, W.Ribarksy, Z.Wartell, D.Krum, B.Singletary, L.F. Hodges, "The Perceptive Workbench: Towards Spontaneous and Natural Interaction in Semi-Immersive Virtual Environments," Proceedings of IEEE Virtual Reality 2000.

3. M.Roussos, A.E.Johnson, J.Leigh, C.A. Vasilakis, C.R. Barnes, and T.G. Moher, "NICE: Combining Constructionism, Narrative and Collaboration in a Virtual Learning Environment," in Computer Graphics Vol. 31 Num. 3, pp. 62-63, 1997.

4. H.Ko, M.Park, H.Lee "Conceptual Framework of Tangible Space Initiative and its Application Scenario to Heritage Alive!," Virtual Systems and MultiMedia (VSMM2002), pp. 1001-1007, 2002.

5. S.Jang, W.Woo, "ubi-UCAM: A Unified Context-Aware Application Model," Lecture Note Artificial Intelligence, Vol.2680, pp. 178-189, 2003.

6. D.Hong, W.Woo, "SpaceSensor:Real-Time Gesture Tracking for I-NEXT," International Conference on Artificial Reality and Telexisitence(ICAT03), paper ISSN 1345-1278, pp. 95-99, 2003.

7. S.Jang, S.Lee, W.Woo, "Digital Message Sharing System in Public Places," Proc. of International Conference on Ubiquitous Computing, pp.177-178, 2003.