ICAT 2004

vr-UCAM : Unified Context-aware Application Module for Virtual Reality*

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

Abstract

In this paper, we propose context-aware module for virtual reality (VR) to provide more personalized and suitable interaction in current situation. The proposed module, vr-UCAM, is the extended version of ubi-UCAM, a unified context –aware application model for ubiquitous computing (ubiComp) environment. The vrSensor and vrService in vr-UCAM establish dynamic connections with ubiSensor and ubiService in ubi-UCAM. Then, ubiService and vrService share the unified context generated by ubiSensors and vrSensors in order to provide personalized services according to user's context such as situation or surroundings. To demonstrate usefulness of the proposed module, we included it to NAVERLib, XML-based VR framework on a PC cluster. Also, we applied it to the virtual heritage tour system. According to the experimental results, it is evident that vr-UCAM can play an important role in providing more personalized services in virtual environments (VEs).

Key words: ubi-UCAM, Virtual Reality, Ubiquitous Computing

1. Introduction

In ubiComp environment, omnipresent sensors and computing resources will provide context-aware services to satisfy user's everyday needs [1]. Moreover, VEs will be ubiquitous in our daily life. Also, the sensors including user's wearable devices and tangible interfaces will be used with conventional input devices such as keyboard, mouse and wand. In this environment, human and cyber space interaction (HCI) should be based on the user's context information (i.e., situation or surroundings) for more personalized feedback.

Over the last few years, various research activities on context-aware model have been reported. For example, ACE (Adaptive Control of Home Environment) is a system to control temperature and lighting conditions at home by training the daily life pattern of the residents using Neural Net [2]. Both Easy Living and AwareHome have showed how context information can be used in the home environment [3,4]. Also, there are a lot of works about virtual reality systems. For example, VR Juggler is an open source platform to provide a uniform VR application environment [5]. Avango is toolkit for creating immersive applications and NAVERLib helps programmer implement scenario-based VEs more eaily [6,7]. However, the existing studies did not consider a standard VR framework of personalized interaction with various sensors in ubiComp environment.

In this paper, we propose context-aware module for VEs to provide more personalized interaction. The proposed module, vr-UCAM, is the extended version of ubi-UCAM into VEs [1]. vrSensors and vrServices in vr-UCAM establish dynamic connections with ubiSensors and ubiServices in ubi-UCAM. Then, vrServices receive context of real environments (REs) generated by ubiSensors and vrSensors send context of VEs to ubiServices in order to provide personalized interaction according to a user's situation or surroundings.

To demonstrate usefulness of the proposed module, we applied it to Heritage Alive, virtual heritage tour system and compared it with previous system [8]. In a subjective test of the quality of the interaction, the system with vr-UCAM received higher point than the previous system. Moreover, the module helped tourists remember more historic relics after tour. According to a subjective test of quality degradation in case of multi-users, the personalized interaction provided by vr-UCAM can prevent from decreasing the sense of the real under the condition of the multi-users.

This paper is organized as follows: In Section 2, we define context-aware interaction for VEs and in Section 3 we describe the proposed vr-UCAM. The experiment and evaluation are explained in Section 4. Finally, conclusion and future works are discussed in Section 5.

---

* This work was supported by KIST and GIST.

2. Context -aware Interaction

Context-awareness means that one is able to use context information [9]. A system is context-aware if it can extract, interpret and use context information. In this paper, we define context-aware interaction as interaction between REs and VEs based on context information available at the time of the interaction. Fig. 1 shows comparison between context-aware and conventional interaction. Although in recent years, a significant consideration in developing VE has been to enhance interaction technology, the existing researches have been mainly interested in immersive and real-time application. However, context-aware interaction considers not only input from devices but also other information from ubiquitous sensors for more personalized and suitable interaction in current situation. Moreover it allows seamless interaction between REs and VEs. In conventional interaction, User immerses VE but disconnected with REs. If we apply context information in REs to VEs as possible as we can, the condition of VEs is close to REs. Thus, we can reduce discontinuities.

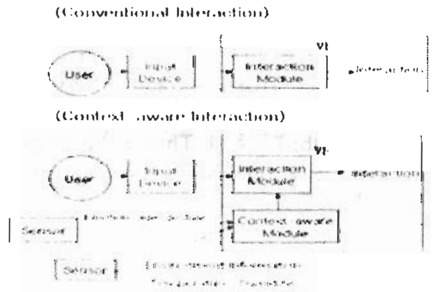

Fig. 1. Comparison between context-aware and conventional interaction.

In order to implement context-aware interaction, sensing context in REs and communication with VEs are the most important. The first prototype of vr-UCAM is designed and implemented to solve those problems.

3. vr-UCAM

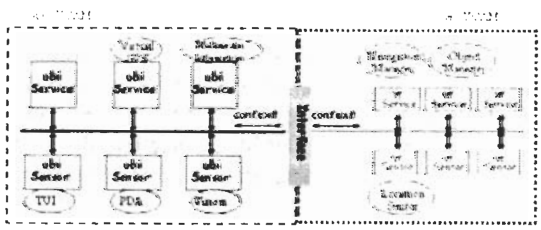

Fig. 2. VE framework based on ubi-UCAM and vr-UCAM.

The proposed vr-UCAM consists of vrSensor and vrService, as shown in Fig 2. In Fig 3, vrSensor consists of three modules: Detection module, Preliminary Context module, and Network module. Detection module monitors the activities of VE and sends the sensed signal to Preliminary Context module. Preliminary Context module creates a preliminary context by analyzing the signals. The preliminary context is expressed by 5W1H (Who, What, Where, When, How, Why), which can be acknowledged by all services, as shown in Fig. 4 and Table. 1. Network module establishes dynamic connections with services and multicasts the preliminary context to the connected services.

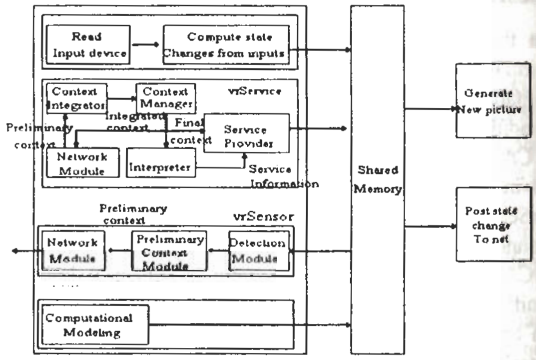

Fig. 3. vrSensor and vrService in vr-UCAM. vr-UCAM has multithread architecture.

Fig. 4. Context format.

vrService consists of five main modules. Network module supports dynamic connections with sensors. Context Integrator collects preliminary contexts for a given time from a set of sensors connected to the vrServices. Also it determines the integrated context by merging the preliminary contexts. Then, Context Manager compares the integrated context with all conditional contexts defined by user in Interpreter. If a match is found, it retrieves the name of service associated with the conditional context and passes the name and the corresponding final context to Service Provider. Final context is generated by refining the integrated context with the current state of vrService.

Table. 1. Context definition.

Context	Field	Definition
Who	String User_Name Double User_ID_Num	User name User ID
What	String object_Name Double object_ID	Object name Object ID
Where	URL Place_URL Coordinates (X,Y,Z)	Location URL Position
When	Enumeration Time Time Absolute_Time	Relative time Absolute time
How	int Action_ID	Behavior ID
Why	Enumeration emotion	Emotion

We consider two kinds of interaction between REs and VEs., as shown in Fig. 5, i.e, interaction from ubiSensor to vrService, and interaction from vrSensor to ubiService. Multicasting and dynamic service discovery are used for interaction in vr-UCAM. The nodes such as ubiSensor and vrSensor make multicasting groups through an embedded ad-hoc networking module. In order to create a multicasting group, each node generates metadata representing its working area. Based on the metadata, ubiServices and vrServices decide whether they join the multicasting group or not. Therefore sensors or services can dynamically connect to others without any centralized network management [1].

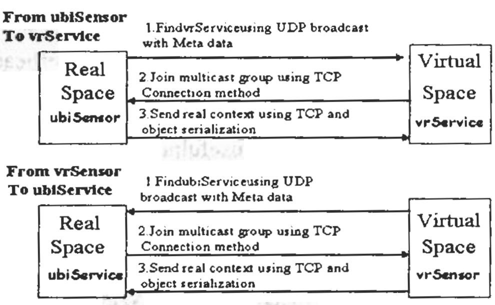

Fig. 5. Context-based Interaction. A sensor uses metadata broadcasting for service discovery and generates TCP-based multicasting group for reliable multicasting.

```
<vr-UCAM>
.....
        <vrService>
                <port>4446</port>
                <meta>door</meta>
                <context>
                        <who>user_A</who>
                </context>
                <name>NavigationManager</name>
        </vrService>
.....
        </vrSensor>
                <name>MyLocation</name>
                <port>4445</port>
                <meta>door</meta>
        </vrSensor>
</vr-UCAM>
```

Fig. 6. vr-UCAM XML Scripting. VE developers can easily install without programming.

vr-UCAM is implemented as a module in NAVERLib. Therefore, each vrSensor and vrService can be created by using XML scripting as shown in Fig. 6 [7]. The vrService in Fig. 6 provides personalized navigation service in virtual place_A, by using port number 4446. The ubiSensors interested in VE_1 can use this service. Also, the unified context includes "user_A" and "move" to trigger this service. The vrSensor in Fig. 6 provides information on the virtual camera position to the ubiServices interested in VE_1.

vrUCAM is developed by using object orientation programming concept. Therefore VE Developers can make their own vrSensors and vrServices easily. Fig. 7 shows class structure of vrService. vrService includes the five modules mentioned above. All virtual services which inherit vrService class can use the four modules such as Network module, Context Integrator, Context Manager, and Interpreter. The VE developers can make their own vrServices conveniently if they redefine Service Provider offered as virtual function.

```
class vrService{
....
    void NetworkModule():
    void ContextIntegrator(){
            ContextManager()
    }
    void ContextManager(){
            If (Integrator() == true)
                    ServiceProvider()
    }

    bool Interpreter():
    virtual void ServiceProvider():
    void Run(){
            NetworkModule():
            ContextIntegrator():
    }
....
}

- User define vrService
class NavigationManager : vrService{
....
    void ServiceProvider(){
            //User define source code
    }
...
}
```

Fig. 7. vrService class. VE developer can conveniently define vrSensor and vrService using Object Orientation Programming concept.

4. Experiment and Evaluation

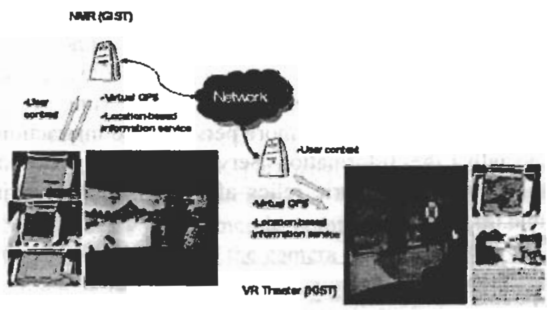

Fig. 8. Virtual heritage tour system. There are two nodes: Tourist1 (GIST), Tourist2 (KIST).

In order to demonstrate usefulness of the vr-UCAM, we applied it to virtual heritage tour system, as shown in Fig. 8. Twelve tourists navigated the virtual heritage, ancient Kyungju, by using their own PDA. At this point, the PDA plays a role in ubiSensor or ubiService. If the PDA is regarded as ubiSensor, it provided the context information such as age, nationality, sex, intellectual level, and location information to vrServices including navigation manager and object manager. The navigation manager helped the tourists navigate with more personalized viewpoint and speed. If the PDA is regarded as ubiService, it provided several services such as information service and virtual GPS based on the context received from location tracking vrSensor. The information service furnished personalized information on historic relics based on tourist's age, intellectual level, and nationality. Also, virtual GPS provided them with location information.

For qualitative evaluation of the personalized context-aware interaction, we have looked at commonly used two methods, called Mean opinion score (MOS) and the Degradation mean opinion score (DMOS) [10].

Fig. 9. The Qualitative evaluation of personalized services.

Table. 2. Comparison of memorable historic relics

	MEAN	STD
With IS	3.167	0.686
Without IS	1.667	0.471

MOS is a rating on a scale from 1-5 where 1 is the lowest or worst and 5 the highest or best. It is a subjective test of the quality of the interaction and feedback. The MOS value is extracted from the results of test Fig. 9 performed on twelve untrained persons. The MOS value of VE with the proposed module is 4.2. It is higher than the MOS value 3.5 of Heritage Alive [8]. It means that the proposed module can improve the sense of real by providing more personalized interaction. Especially, the Information Service (IS) helped them remember more historic relics after tour, as shown in Table. 2.

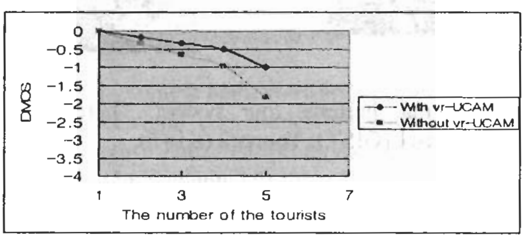

Fig. 10 Characteristics against the number of tourists.

The DMOS is a method for obtaining subjective ratings of perceived quality degradation compared to an original. DMOS uses a five-grade scale ranging from 1, Very annoying, to 5, Imperceptible. Characteristics of the proposed and conventional virtual heritage system against the number of tourists are shown in Fig. 10. The result means that the personalized services provided by vr-UCAM can prevent from decreasing the sense of the real under the condition of the multi-users. However, the transfer delay increases about 20ms with addition of another service to the multicasting group, as shown in Fig. 11. Therefore, a number of services more than 15 cause unnatural interaction.

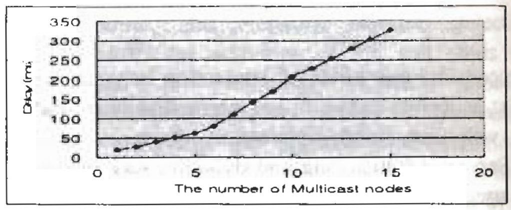

Fig. 11 The networking delay to transfer a context from a sensor to all services

5. Conclusion and Future Work

In this paper, we implemented vr-UCAM in NAVERLib, based on ubi-UCAM and applied it to the virtual heritage tour system. In this application, the tourists could use their own personal device as input device and receive various personalized services. The remaining challenges include reducing the overhead of vr-UCAM, and finding out reasonable method for usability test. Moreover, we will develop various applications to demonstrate usefulness of the proposed module.

References

1. S. Jang, W. Woo, "ubi-UCAM: A Unified Context - Aware Application Model" *LNAI,* vol. 2680, pp. 178 189, 2003.
2. M. Mozer, "The Neural Network house: An environment that adapts to its inhabitants," In *Proc, the American Association for Artificial Intelligence Spring Symposium on Intelligent Environments,* AAAI Press. pp. 110-114, 1998.
3. S. Shafer, B. Brumitt, and B. Meyers, "The EasyLiving Intelligent Environment System," *CHI Workshop on Research Directions in Situated Computing*, 2000.
4. A. Dey, D. Salber, and G. Abowd, "A Context-aware Infrastructure for Smart Environments," In *Proc, MANSE,* 1999.
5. A. Bierbaum, "VR Juggler: A Virtual Platform For Virtual Reality Application Development", in *Proc, IEEE Virtual Reality*, pp. 89-96, 2001
6. H. Tramberend, "Avocado: A Distributed Virtual Reality Framework," in *Proc, IEEE Virtual Reality,* 1999.
7. C.Park, H.Ko, H.Ahn, J.Kim "NAVER : design and implementation of XML-based VR Framework on a PC cluster", *VSMM,* pp. 967 - 975, 2002
8. H.Ko, M.Park, and H.Lee, "Conceptual Framework of Tangible Space Initiative and its Application Scenario to Heritage Alive!," *VSMM*, pp. 1001-1007, 2002.
9. A.K. Dey and G.D. Abowd, "Toward a better understanding of context and context-awareness," *GVU Technical Report.*
10. ITU-R Recommendations, "Recommendation 500-10; Methodology for the subjective assessment of the quality of television pictures," *ITU-R Rec.* BT.500-10, 2000.