How to build a Context-aware Architecture for Ubiquitous VR

Yoosoo Oh and Woontack Woo

Abstract— **There are no general context-aware architectures to be useful which could be applied to full life cycle in ubiquitous VR, even though many context-aware architectures or middleware are proposed. In this paper, we propose a general approach of a context-aware architecture for ubiquitous virtual reality. We would discuss how to improve the intelligence ability in a context-aware architecture.**

Index Terms—**Context Interpretation, Context Learning, Context Reasoning, Social Relationship, Ubiquitous VR**

I. Introduction

Ubiquitous VR is a concept of creating ubiquitous VR(virtual reality) environments which make VR pervasive into our daily lives and ubiquitous by allowing VR to meet UbiComp(ubiquitous computing) [1]. To make seamless connection between human beings and VR and UbiComp environments, we need to have an infrastructure which is specified by sensors and services for the collaboration of those environments. In different domains like human, real and virtual environment, a context-aware architecture for Ubiquitous VR is necessary to be established.

There are many previous works related to a context-aware architecture. CAMidO is a Context-Aware Middleware based on an Ontology meta-model [2]. CAMidO Compiler generates Inference rule file and controller source code. However, CAMidO has the sophisticated path in architecture and has the weak relationship among context sources. JCAF is the Java Context Awareness Framework [3]. JCAF cooperates with context services and supports for adding, deleting, modifying, or varying functionality, capacity, or platform at runtime. JCAF should solve synchronizing data used in many systems. SOCAM is a Service-Oriented Context-Aware Middleware [4]. SOCAM supports context reasoning and knowledge sharing process by using common context model. SOCAM depends on OWL, and so it's not easy to be used widely. Besides these architectures, there are many related works.

However there is still a need for a generic solution for seamless connection among users and real/virtual environments. Previous context-aware architectures are not proper to use and apply to real and virtual environments. Those should be considered more about connection among domain-free entities. Thus, more general approach to context-aware architecture should be developed. In this paper, we propose a general approach of a context-aware architecture for ubiquitous virtual reality. Also, we want to discuss some issues which are how to improve the intelligence ability in a context-aware architecture.

There are no general context-aware architectures to be useful which could be applied to full life cycle in ubiquitous VR, even though many context-aware architectures or middleware are proposed. Also, it's hard to apply the working architectures or middleware to a new domain that is not mentioned in their original works. Thus, we should discuss how to build standardized and generalized context-aware architectures for various environments.

II. Context-aware Architecture: UCAM

A. UCAM architecture

UCAM is a Unified Context-aware Application Model for seamless human-content-environment interaction in ubiquitous computing environment [5]. Figure 1 shows UCAM architecture. For the general approach, we modified previous UCAM. Our proposed UCAM particularly supports intelligence like context reasoning, learning, interpretation, and social relationship management. The proposed UCAM architecture used context model which describes context as a form of 5W1H (Who, What, Where, When, How, and Why) [5]. The context model also maintains context history as context repository.

The proposed architecture can improve the awareness ability of context because the architecture can interpret, reason out, and learn contexts. The architecture can connect any tool for the awareness ability as an add-on feature. Also, the architecture manages social relationship among human, real objects, and virtual objects. For these aspects, our architecture is generalized as an open source code which is a library for basic features. An application developer can simply use our

This work was supported by the IT R&D program of MIC/IITA. [2005-S-069-02, Development of Technologies for Unifying and Fusing Context].

Yoosoo Oh is with the Gwangju Institute of Science and Technology, Gwangju 500-712, S.Korea (e-mail: yoh@gist.ac.kr).

Woontack Woo is with the Gwangju Institute of Science and Technology, Gwangju 500-712, S.Korea (corresponding author to provide phone: +82-62-970-2226; fax: +92-62-970-2204; e-mail: wwoo@gist.ac.kr).

architecture by inheriting the library code.

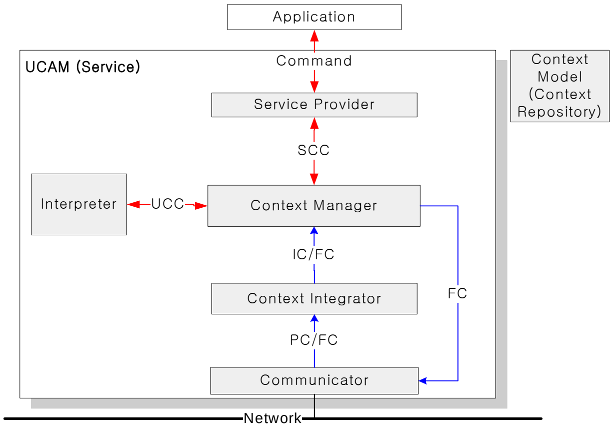

Figure 1. UCAM service part. (PC: Preliminary Context, IC: Integrated Context, FC: Final Context, UCC: User Conditional Context, SCC: Service Conditional Context)

B. *Core technology*

1) Context Model

Context Model in UCAM is defined a set of user contexts that supports ontological structure and reasoning operations. It has to know what kinds of operations are needed in each component in UCAM. It is related to all components in UCAM and generates and saves contexts.

2) Context Reasoning

Context Reasoning in UCAM is a logical process which extracts high-level(implicit) context from low-level(explicit) context and maintains context consistency in processing context modeling. For context reasoning, it is necessary to various kinds of sensors and information sources. Context reasoning process reuses context (context history) and adapts reconfigurable rules to the diverse environment. It is related to sensor part of UCAM and Context Integrator in UAM service.

3) Context Learning

Whether the context-aware application obtains its context information from environmental sensors, user input, VR applications, or some combination of these, it must perform a good deal of processing on the long-term context in order to be able to accurately evaluate the state of the environment, the intensions of the user, and avatar's intelligent reaction. Thus, Context Learning in UCAM is required to context-aware applications. Context learning is related to Context Manager in UCAM, and uses classified learning mechanisms according to the characteristics of them.

4) Context Interpretation

Context Interpretation in UCAM is the way the entity generates conditional context for performing certain task, and the way the entity presents current context to the entity-readable way. It generates conditional context from user's direct input (Feedback) and do context-to-service parameter mapping (Domain Change). Context interpreter maps elements in context format to each application domain, and one application to another application. It requires knowledge about both context format and application parameters for interpretation.

5) Social Relationship Management

Social Relationship Management in UCAM determines relationships among sensors and services which participating in ubiquitous VR environment (human, real/virtual env.). To manage social relationship among entities in ubiquitous VR environment, those entities (users, services and sensors) should collaborate each another in the created community. This process is done by Communicator in UCAM which takes charge of network configuration and sends messages to community members.

III. Discussion

The proposed architecture can improve the awareness ability and extend to diverse application domains. Also, the architecture, UCAM, is generalized as a library in order to be used easily by application developers. However we still have many issues to be discussed. First issue is how to predict user's behavior and connect to the other objects in real/virtual environment. Basically, UCAM can support the reasoning aspect, not prediction for user's behavior. For more generalization, this issue should be considered. Second issue is full life cycle support for context-aware applications. Adding, deleting, replacing, modifying, and debugging entities should be supported. These functionalities are already supported by UCAM, but need to be improved for ubiquitous VR environment. Last issue is how to evaluate a context-aware architecture. There are good evaluation methods in HCI field. Thus, those methods should be applied to our UCAM with some evaluation factors, such as architecture structure, scalability, accountability, reconfiguration, reusability, debugging, openness, flexibility, etc. These evaluation factors could be useful to verify the general approach and intelligence ability of our UCAM.

References

[1] S.Kim, Y.Suh, Y.Lee, W.Woo, "Toward ubiquitous VR: When VR Meets ubiComp," *ISUVR 2006*, pp. 1-4, 2006.

[2] Nabiha Belhanafi Behlouli, Chantal Taconet and Guy Bernard, "An Architecture for supporting Development and Execution of Context-Aware Component applications," *In IEEE International Conference on Pervasive Services, ICPS 2006*, Lyon, France, 26-29 June, 2006.

[3] Jakob E. Bardram, "The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications," *In Proceedings of the 3rd International Conference on Pervasive Computing (Pervasive 2005)*, 2005.

[4] Tao Gu, Hung Keng Pung, Da Qing Zhang, "Toward an OSGi-Based Infrastructure for Context-Aware Applications," *IEEE Pervasive Computing*, vol.03, no.4, pp.66-74, Oct-Dec, 2004.

[5] D.Hong, C.Shin, Sj.Oh, W.Woo, "A New Paradigm for User Interaction in Ubiquitous Computing Environment," *ISUVR 2006*, pp. 41-44, 2006.