

An Adaptive Concurrency Control QOS Agent for Ubiquitous Computing Environments

Eung Nam Ko

Department of Information & Communication, Baekseok University,
115, Anseo-Dong, Cheonan, ChungNam, 330-704, Korea
ssken@bu.ac.kr

Abstract. This paper presents the design of the adaptive concurrency control QOS agent, which is running on RCSM(Reconfigurable Context Sensitive Middleware) for ubiquitous networks. RCSM provides standardized communication protocols to interoperate an application with others under dynamically changing situations. It describes a hybrid software architecture that is running on situation-aware middleware for a web based distance education system which has an object with a various information for each session and it also supports multicasting with this information. There are two approaches to software architecture on which distributed, collaborative applications are based. Those include CACV(Centralized-Abstraction and Centralized-View) and RARV(Replicated-Abstraction and Replicated-View). We propose an adaptive concurrency control QOS agent based on a hybrid software architecture which is adopting the advantage of CACV and RARV for situation-aware middleware.

1. Introduction

A variety of collaborative applications have been developed ranging from video conferencing tools to shared whiteboard and text editors to distributed virtual environments to video recording on-demand systems[1,2]. The multimedia distance education is concentrated an interest about new education methods by join an education engineering and an information communication technology[3]. A general web-based distance system uses video data and audio data to provide synchronize between teacher and student. In a ubiquitous computing environment, *computing anytime, anywhere, any devices*, the concept of situation-aware middleware has played very important roles in matching user needs with available computing resources in transparent manner in dynamic environments [4]. It is difficult to avoid a problem of the seam in the ubiquitous computing environment for seamless services. Thus, there is a great need for concurrency control algorithm in situation-aware middleware to provide dependable services in ubiquitous computing. The system for a web based multimedia distance education includes several features such as audio,

video, whiteboard, etc, running on situation-aware middleware for internet environment which is able to share HTML format. This paper describes a hybrid software architecture that is running on situation-aware middleware for a web based distance education system which has an object with a various information for each session and it also supports multicasting with this information. There are two approaches to software architecture on which distributed, collaborative applications are based. Those include CACV(Centralized-Abstraction and Centralized-View) and RARV(Replicated-Abstraction and Replicated-View). We propose an adaptive concurrency control QOS agent based on a hybrid software architecture which is adopting the advantage of CACV and RARV for situation-aware.

2. QoS Layered Model

Traditional QoS (ISO standards) was provided by the network layer of the communication system. An enhancement of QoS was achieved through inducing QoS transport services. For multimedia communication system, the QoS notion must be extended because many other services contribute to the end-to-end service quality. The multimedia communication system consists of three layers: application, system(including communication services and operating system services), and devices(network and multimedia devices). As shown in Figure 3, the organization of QoS-layered model for the multimedia communication system include 4 layers. The four layers consist of a user QoS layer, an application QoS layer, a system QoS layer and a network QoS layer[5].

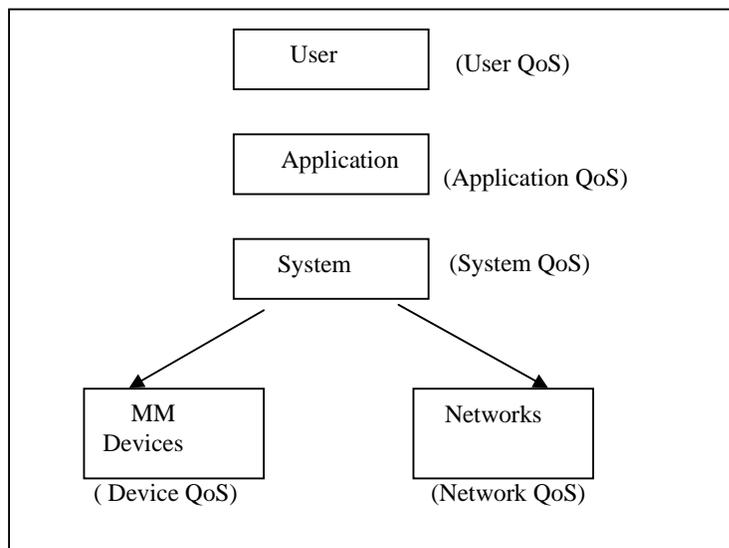


Fig.1 QoS Layering

3. Our Approach

3.1 RCSM(Reconfigurable Context-Sensitive Middleware)

In the Context Toolkit, a predefined context is acquired and processed in context widgets and then reported to the application through application-initiated queries and callback functions. In this Reconfigurable Context-Sensitive Middleware(RCSM), Stephen S. Yau et al.[6] proposed a new approach in designing their middleware to directly trigger the appropriate actions in an application rather than have the application itself decide which method(or action) to activate based on context. RCSM provides an Object-based framework for supporting context-sensitive applications. Figure 2 shows how all of RCSM's components are layered inside a device.

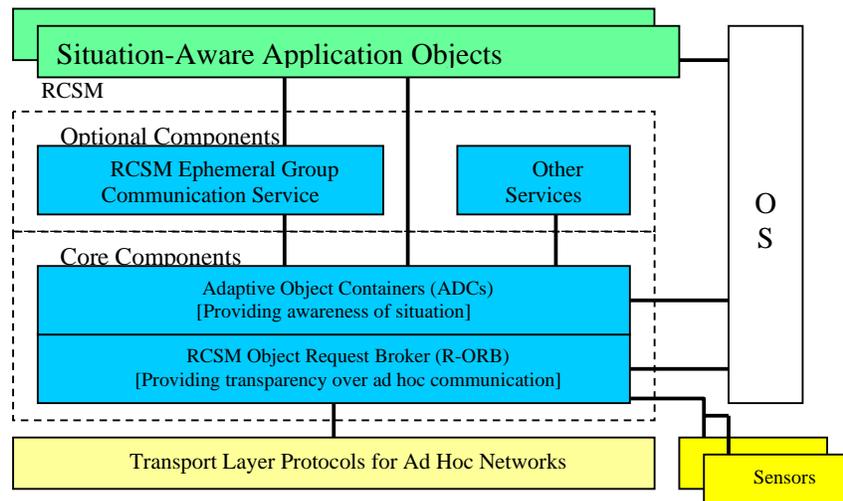


Fig. 2. RCSM's integrated components

All of RCSM's components are layered inside a device. The Object Request Broker of RCSM (R-ORB) assumes the availability of reliable transport protocols; one R-ORB per device is sufficient. The number of Adaptive object Containers (ADC)s depends on the number of context-sensitive objects in the device. ADCs periodically collect the necessary "raw context data" through the R-ORB, which in turn collects the data from sensors and the operating system. Initially, each ADC registers with the R-ORB to express its needs for contexts and to publish the corresponding context-sensitive interface. RCSM is called reconfigurable because it allows addition or deletion of individual ADCs during runtime (to manage new or existing context-sensitive application objects) without affecting other runtime operations inside RCSM.

Ubiquitous applications require use of various contexts to adaptively communicate with each other across multiple network environments, such as mobile ad hoc networks, Internet, and mobile phone networks. An example of *SmartClassroom* is illustrated in [7]. However, it did not include concurrency control QoS support in the architecture.

3.2 QOS Layered Model for Multimedia Distance Education System

Our proposed model aims at supporting concurrency control mechanism running RCSM in order to provide ubiquitous, seamless services. An example of situation-aware applications is a multimedia distance education system. As shown in Figure 3, multimedia distance education systems include advances services, coordination services, cooperation services, and media services. Advances services consist of various subclass modules. This subclass module provides the basic services, while advances services layer supports mixture of various basic services. Advances services include creation/deletion of shared video window and of creation/deletion of shared window. Shared window object provides free hand line, straight line, box, text to collaboration work participant and the participants can use such as the same file in this shared windows. Coordination services include session control module, and floor control module. Session control module controls the access to the whole session. This session can be meeting, distance learning, game and development of any software. Session control also facilitates the access and limits the access to the whole session. Session control module monitors the session starts, terminates, joins and invites and it also permit another sub-sessions. Session control module has an object with an various information for each session and it also supports multicasting with this information. Floor control controls the person who can talk, and person who can change the information. Mechanism of the floor control consists of braining storming, priority, mediated, token-passing and time-out, In floor control module, it provides explicit floor and braining storming.

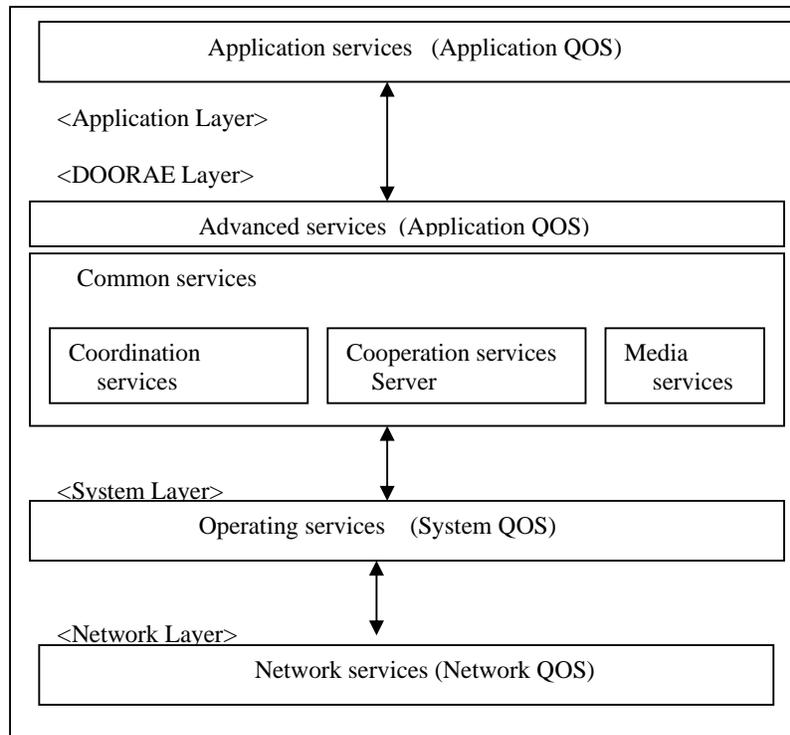


Fig.3 QoS Layered Model for Multimedia Distance Education System

Cooperation services include window overlays module, and window sharing module. Window overlays module is laid a simple sketching tool over a copied window. It provides all users with transparent background and tele-pointers. So, all users can point and gesture. Window sharing module is a combination of window copying, window overlays, floor control and session control. All users are able to interact through application shared by them. One user is running a single user application. The other users get to see exactly what this user sees. The application can allow different users to interact with the application by selecting one of the user's keyboard and mouse the source of input. Media services support convenient services for application using DOORAE environment. Supplied services are the creation and deletion of the service object for media use, media share between remote user. Media services modules limit the service by hardware constraint.

We assumed throughout this paper the model shown in Figure 3. This model consists 3 QoS layer: Application QoS(including application layer and DOORAE layer), System QoS(including system layer), Network QoS(including communication layer). In this paper, we concentrate in the application QoS layer. There are several constraints which must be satisfied to provide guarantees during multimedia transmission. They are time, space, device, frequency, and reliability constraints.

Time constraints include delays. Space constraints are such as system buffers. Device constraints are such as frame grabbers allocation. Frequency constraints include network bandwidth and system bandwidth for data transmission. In this paper, we discussed concurrency control constraints.

3.3 Web Based Multimedia Distance Education System

This paper proposes an URL synchronization function used in WebNote with remote collaborative education system based on situation-aware middleware for CBM(Computer Based Multimedia). It retrieves the common characteristics of these tools and designs an integrated model including all these methods for supporting concurrent collaborative workspace. As shown in Figure 4, this paper describes an integrated model which supports object drawing, application sharing, and web synchronization methods of sharing information through a common view between concurrently collaborating users. This proposed model consists of multiple view layout and each layout control, a unified user interface, and defines the attributes of a shared object.

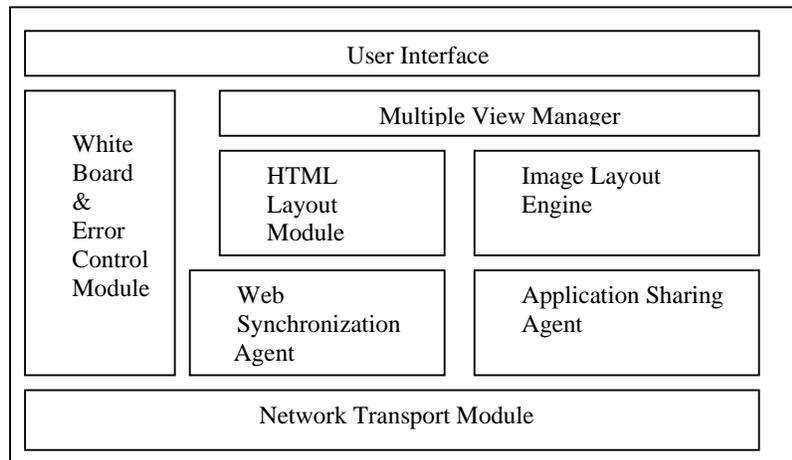


Fig.4 An Integrated Model with Web Synchronization

3.4 URL synchronization

As shown in Figure 5, you can see the relationship between WebNote Instance and WebNote SM. This system is used to be one of services that are implemented on Remote Education System. This Remote Education System includes several features such as Audio, Video, Whiteboard, WebNote running on Internet environment which is able to share HTML(Hyper Text Mark-up Language). We have implemented WebNote function to do so either. While session is ongoing, almost all participants

are able to exchange HTML documents. For this reason, we need the URL synchronization. To win over such dilemma for centralized or replicated architecture, a combined approach, CARV(the centralized abstraction and replicated view) architecture is used to realize the application sharing agent. This system is used to be one of services that are implemented on Remote Education System. This Remote Education System includes several features such as Audio, Video, Whiteboard, WebNote running on Internet environment which is able to share HTML(Hyper Text Mark-up Language). We have implemented WebNote function to do so either. While session is ongoing, almost all participants are able to exchange HTML documents. For this reason, we need the URL synchronization. To win over such dilemma for centralized or replicated architecture, a combined approach, CARV(the centralized abstraction and replicated view) architecture is used to realize the application sharing agent.

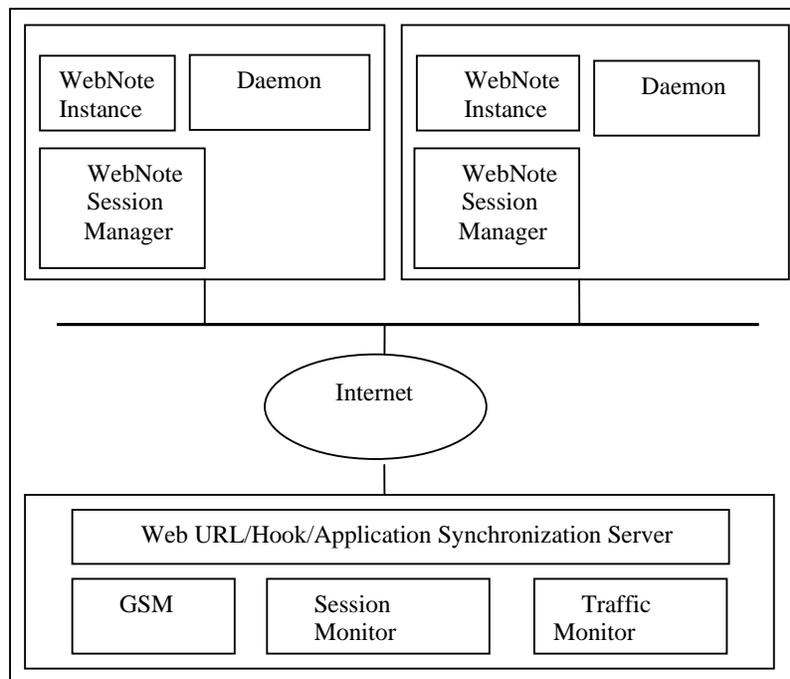


Fig.5. The relationship between WebNote Instance & WebNote SM

3.5 Hybrid Software Architecture for Concurrency Control

As shown in Figure 6, to win over such dilemma for centralized or replicated architecture, a combined approach, CARV(the centralized abstraction and replicated view) architecture is used to realize the application sharing agent. The shared window is a window shared by all the participants, and the modification carried out by the

speaker is notified to every other participants. The local window is not shared except initial file. The tool box provides various tools for editing contents of both the shared window and the local window. Figure 6 shows that teacher and students use their local windows and shared window individually. The local window has the lecture plans which is distributed at the beginning, and enables participants to memo and browsing other parts in the lesson plans, and has functions as a whiteboard.

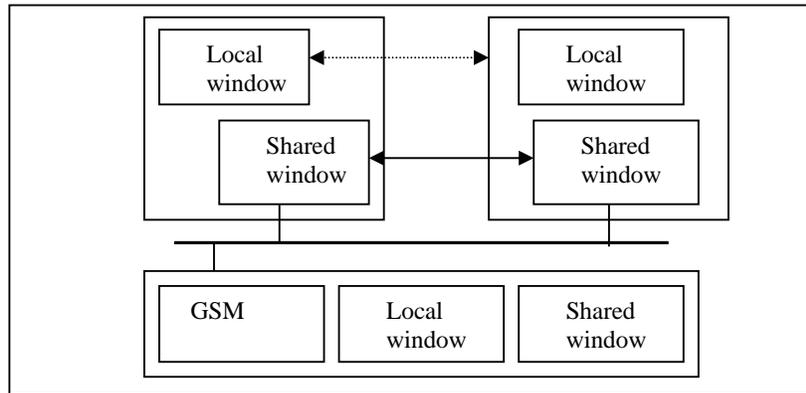


Fig. 6 Web Based CARV Architecture for Concurrency Control

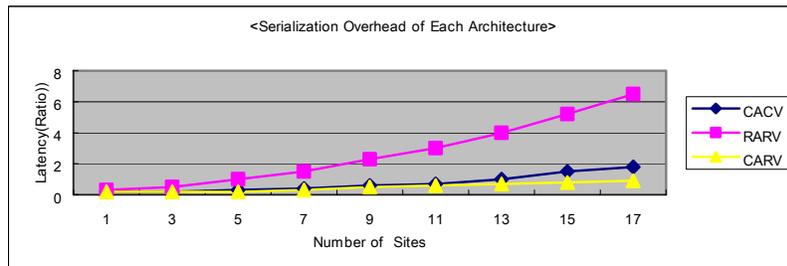
4. Simulation Results

As shown in Table 1, conventional multimedia distance education systems are Shastra, MERMAID, MMconf, and CECED. You can see the characteristic function of each system function for multimedia distance education. A proposed main structure is distributed architecture but for application program sharing, centralized architecture is used. The problem of rapid increase in communication load due to growth in number of participants was solved by letting only one transmission even with presence of many users, using simultaneous broadcasting. Basically, there are two architectures to implement such collaborative applications; the centralized architecture and replicated architecture, which are in the opposite side of performance spectrum. Because the centralized architecture has to transmit huge amount of view traffic over network medium, its performance is reduced to contaminate the benefits of its simple architecture to share a copy of conventional application program. On the other hand, the replicated architecture guarantees better performance in virtue of its reduced communication costs. However, because the replicated architecture is based on the replication of a copy of application program, it is not suit to use for application sharing realization.

Table 1 Analysis of Conventional Multimedia Distance Education System

Function	Sha- Stra	MER- MAID	MM- conf	CE- CED
OS	UNIX	UNIX	UNIX	UNIX
Development Location	Purdue Univ. USA	NEC, JAPAN	CamBridge USA	SRI, International
Development Year	1994	1990	1990	1993
Structure	Server /client	Server /client	Centralized or Replicated	Repli- cated
protocol	TCP/IP	TCP/IP	TCP/IP	TCP/IP multicast
Concurrency Control running on RCSM	No	No	No	No
Web Based running on RCSM	No	No	No	No

The figure on the vertical line means relative overheads compared with maximum value on RARV.

**Fig. 7 Serialization Overhead of Each Architecture**

5. Conclusions

The focus of situation-aware ubiquitous computing has increased lately. An example of situation-aware applications is a multimedia education system. The development of multimedia computers and communication techniques has made it possible for a mind to be transmitted from a teacher to a student in distance environment. The roles of application program sharing are divided into two main

parts; abstraction and sharing of view generation. Application program sharing must take different from each other according to number of replicated application program and an event command. There are two different structures. Those are CACV and RARV. In this paper, we discuss a hybrid software architecture which is adopting the advantage of CACV and RARV. CACV is centralized architecture where only one application program exists among entire sharing agents and centralized server takes care of input processing and abstraction is the same. RARV is replicated architecture at which application data input(event) that is generated when sharing takes place is transferred and executed. This means that only event information is separately performed. We proposed an adaptive concurrency control QoS agent based on a hybrid software architecture which is adopting the advantage of CACV and RARV for situation-aware middleware. It described a hybrid software architecture that is running on situation-aware ubiquitous computing for a web based distance education system which has an object with a various information for each session and it also supports multicasting with this information. This paper proposed a new model of concurrency control by analyzing the window and attributes of the attributes of the object, and based on this, a mechanism that offers a seamless view without interfering with concurrency control is also suggested. We remain these QoS resolution strategies as future work.

References

- [1] Holfelder W, "Interactive remote recording and playback of multicast videoconferences", Proc. of IDMS'97, Darmstadt, Germany, Sep. 1997.
- [2] G.Fortino and L. Nigro, "A Cooperative Playback System for On-Demand Multimedia Sessions over Internet", 2000 IEEE International Conference on Multimedia and Expo ICME2000 Proceedings Volume I, July 30 – August 2, NY USA, pp.41-44.
- [3] T. Boyle, "Design for Multimedia Learning", Prentice Hall Europe, 1997.
- [4]S. S. Yau and F. Karim, "Contention-Sensitive Middleware for Real-time Software in Ubiquitous Computing Environments", *Proc. 4th IEEE Int'l Symp. on Object-Oriented Real-time Distributed Computing (ISORC 2001)*, May 2001, pp. 163-170.
- [5]Ralf Steinmetz and Klara Nahrstedt, "Multimedia: computing, communications & Applications", Prentice Hall P T R.
- [6]S. S. Yau, Y. Wang, D. Huang, and H. In "A Middleware Situation-Aware Contract Specification Language for Ubiquitous Computing", *FTDCS 2003*.
- [7]Saha, D.; Mukherjee, A.; "Pervasive computing: a paradigm for the 21st century", IEEE Computer, Volume: 36, Issue:3, March 2003, Page(s): 25-31.