Mobile Proxy Architecture and Its Practice: Mobile Multimedia Collaboration System

Sangyoon Oh*

모바일 기기를 위한 프록시 구조와 모바일 멀티미디어 협업 시스템 적용예

오 상 윤*

Abstract

The performance and portability of mobile applications can be greatly increased by adopting proxy modules which exists between the conventional system and the device. When mobile devices collaborate with the conventional computers, there are problems to address: a battery life problem, limited input and output methods, and intermittent wireless connection. Those issues are magnified in the multimedia collaboration environment since it works in a real-time condition and the size of the message in the system is big in many cases. Additionally, because multimedia collaboration system softwares are too heavy and complex for mobile devices, it is very hard to integrate them with conventional systems. In this paper, we describe our design and its implementation of a novel approach to map events (i.e. messages) using a proxy for mobile applications. We adopt a proxy to provide a content adaptation (i.e. transcoding) where the message contents are customized. Also, we design a mobile version publish/subscribe system to provide communication service for mobile device in loosely coupled and flexible manner. We present our empirical results which show that our design can be efficiently implemented and integrated with a conventional multimedia collaboration system.

• 제1저자 : 오상윤
• 투고일 : 2009. 06. 25, 심사일 : 2009. 06. 29, 게재확정일 : 2009. 07. 10.
* 아주대학교 정보 및 컴퓨터공학부 조교수
※ 본 연구는 2007학년도 아주대학교 정착연구비 지원에 의하여 연구되었음
※ 본 연구는 2009학년도 아주대학교 일반교내연구비 지원에 의하여 수행되었음
※ 본 연구는 지식경제부 및 정보통신연구진흥원의 대학IT연구센터지원사업의 연구결과로 수행되었음(IITA-2009-C1090-0902-0003)

요 약

모바일 기기들이 기존의 일반 컴퓨터들과 협업을 하는 경우에는 일반 컴퓨터들 간의 협업에서는 나타나지 않던 새로운 문제들, 즉 전원제한 때문에 나타나는 짧은 사용시간, 모바일 기기들이 가지는 제한된 입출력 방식, 그리고 기존의 유선통신에 비해 불안정한 무선 통신 등의 문제들이 발생을 하게 된다. 특별히, 모바일 기기들을 활용한 멀티미디어 협업시스템의 경우는 애플리케이션의 특성상 상대적으로 메시지의 크기가 크고, 실시간성 보장이 필요한 이유로 위에 언급된 문제들의 정도가 심각해지는 현상을 보이게 된다. 또한 멀티미디어 협업시스템을 이용하기 위한 소프트웨어 시스템이 상대적으로 복잡하고 무거워서 모바일기기에 수용되기 매우 어려웠다. 본 논문에서는 모바일 애플리케이션의 성능과 이식성 향상을 위해서, Proxy를 사용한 효과적인 모바일 애플리케이션 구동환경을 제안한다. 제안된 환경은 이벤트 매핑과 메시지 콘텐츠 최적화를 Proxy에서 처리하며, 모바일 메세징 시스템을 이용하여 느슨하고 유연한 구조를 취할 수 있도록 하였다. 제안된 구조 디자인을 이용한 Proxy를 구현하여 모바일 기기가 실제 멀티미디어 협업시스템과 연동이 가능하도록 하였으며, 모바일 협업 애플리케이션들을 구현하여 실제 시스템과 연동 실험한 결과 모바일 애플리케이션이 효과적으로 작동되는 것을 확인할 수 있었다.

▸ Keyword : 트랜스코딩(transcoding), 프록시 구조(proxy architecture), 멀티미디어 협업시스템 (multimedia collaboration system)

I. Introduction

In the past decade, ubiquitous computing environments have been changed significantly and the changes give people freedom to use computing resource without sitting at the desk. Mobile computing is the area we can see the biggest aspectual impact of the changes to the society and economy. Recently, the capability of mobile devices and wireless connection technology has increased dramatically. The performance of the mobile device is significantly enhanced by faster processors, larger installed memory, and enhanced user display. Meanwhile, the connection to a network has become easier through a widely available the third generation (3G) networks and IEEE 802.11. Because of these improvements, more and more researchers are adopting mobile devices into their system [21~23] and examples spread over many areas including social network analysis (SNS), the computational grid, and mobile multimedia collaboration works.

Yet, there are few hurdles which have not been fully solved in mobile computing environments. First, there is a battery life problem. If the device computes more, it consumes more battery by the law of physics unless we have more efficient processing unit. Thus, less computation is necessary to keep the device mobile.

Second, the size of the display is limited compare to the desktop computers. There are some 'state-of-art' cellular phones which are capable of 'full browsing'. However, for the most of devices on the market documents and images on mobile devices should be customized before rendered. Especially the image file type should be checked before rendering because mobile platforms support selective image types. Any given image file type may be supported by 'the A mobile platform' and it isn't supported by 'the B mobile platform'.

Third, the input method is limited. After iPhone [1] and iPod touch [2] hit the market hard, touch-screen has become a de facto standard of the input method for high-end mobile device. However, a touch-screen is still a less capable input method compared to a combination of a keyboard and a mouse. Therefore it is recommended to the application developers to reduce the user inputs in the scenario when they design the application.

Fourth, the wireless connection is slower and intermittent. The reliability of the wireless connection is catching up that of the landline (e.g. Local Area Network), but there is a still gap between wireless and landline. Rather than it like conventional machines, the most of mobile devices use wireless connection. Thus, there have been many investigations to address these issues. But these proposals and solutions tackle small pieces of the problem, rather providing the integrated system level solution.

Like we listed them above, there are issues to be addressed when we integrate mobile devices into the conventional system. Those issues are magnified in the multimedia collaboration system which enables people to communicate and cooperate remotely, since it works in a real-time condition and the size of the message in the system is big in many cases. We claim that an appropriate proxy module between the collaborative system and the mobile devices can resolve this drawback through efficient event mapping and adapting content of the messages.

In this paper, we propose a new architecture design for mobile multimedia applications to increase performance and portability. Our proposal includes solutions for two major issues in mobile computing. First, we propose adopting a proxy module to provide the content adaptation (i.e. transcoding). For the intermittent mobile communication channel, we design and implement a mobile publish/subscribe communication service, the HandHeld Messaging System (HHMS) [3]. HHMS is used to provide the mobile publish/subscribe messaging service between conventional collaboration system and heterogeneous mobile devices.

In order to demonstrate the potential of our proposed design, we implemented the design and it will be discussed in the following sections. We organized this paper as follows. In section 2, we discuss background work including other mobile platforms and multimedia collaboration system. Section 3 reviews the architecture design we use in our approach. We illustrate implementation details of the implemented system in section 4. We conclude in section 5.

II. Related Works

There are some notable research efforts to adopt mobile devices into the conventional systems using a proxy. Focuses of current research works are vary from the simple content adaptation to the sophisticate context-aware mobile platforms and we can easily find researches which cover multiple research topics (e.g. a middleware platform supporting content adaptation and context aware processing. We discuss approaches and solutions for research topics using proxy architecture.

Additionally, we also provide a description and a discussion about Multimedia Collaboration System, Global MultiMedia Collaboration System (Global MMCS) which is used for the target application to verify our proposed design. Global MMCS is design and developed at Community Grids Laboratory of Indiana University [4,5].

1. Content Adaptation

As described in the previous section, one of huddles to tackle is content adaptation for heterogeneous mobile devices. This is still an important issue to solve since the approaches and techniques for content adaptation are also applicable to the mediation between different standard in the distributed environment.

Layaida et al. [6,7] use SMIL to adapt content for embedded devices combined with proxy architecture. SMIL (Synchronized Multimedia Integretion Language) [8] mark up is XML-based and a W3C recommendation for describing a multimedia contents, such as images, audio, and video. Most of SMIL adaptation is done on the client side and this would be a big performance overhead to embedded devices with limited processing capability. Thus, they introduced a proxy between clients (i.e. embedded devices) and servers (i.e.

media content servers) to get request from clients, retrieve the contents, adapt contents and deliver them to clients. In this case, clients and servers can stand still without changes and contents are adapted for heterogeneous hardware devices. Their approach has similarities with our approach in terms of 1) using a proxy to adapt contents and 2) using a metadata, i.e. SMIL to customizing contents with respect to the characteristics. Their experiments shows the proxy and metadata for customizing (i.e. SMIL) can provide good content adaptation architecture when they are mixed. Their approach address an individual issue for content adaptation, however our approach address a communication service issue as well for more complete system level solution.

2. Context-awareness

Context-awareness is one of the key characteristics of ubiquitous computing which provides computational services through everyday objects. Since adopting mobile devices truly provides a requirement of "everyday object" to the ubiquitous system, it has been a hot issue for recent years [9,10,11,14,15,16]. There are many research works which use a proxy to gather context-aware information

MobiPADS [9] is a context aware system for mobile computing. Primarily, MobiPADS provides a mobile platform for active deployment of services over the wireless environment. In addition to its primary purpose, it is also designed to support context-aware processing by providing an executing platform to enable active service deployment and reconfiguration [10] of the service mix in response to an environment where the context varies.

There are research results from Finnish research institutes: ContextPhone [11] is a software platform consisting of four interconnected modules provided as a set of open source C++ libraries and source code components. It runs on off-the-shelf mobile phones with the Symbian [12] and the Nokia Series 60 Smartphone platforms [13]. It gets context data from sensors (location, user interaction, communication behavior, and physical environment) and feed information as a resource to customizable services and application for machine adaptations.

CAPNET is a mobile platform middleware developed by Davidyuk et al. [14] to fulfill the requirements set to the architecture by context-awareness, mobility of the software components, multimedia applications and adaptation. They use collected location data to record stream video.

3. Multimedia Collaboration System

By its definition, the collaboration system provides structured, recursive process where two or more people at geographically distributed locations work together toward a common goal. Specially, Multimedia Collaboration Systems supports richer environment to participants through Audio/video conferencing, shared display, and other multimedia applications/services. Since we can increase a level of accessibility by adding mobile devices as clients, customizing relatively heavy multimedia data is very important research issues.

In this section, we describe the GlobalMMCS which is a Web Services based, heterogeneous collaboration system which supports distance education and collaboration computing. It can integrate multiple audio/video conferencing system with SIP VoIP [17], H.323 and AccessGrid [18] system. It also provides an instant messaging service, shared display, shared export and shared application. The framework has two technical factors for supporting general interoperable capability over heterogeneous network environment: a messaging and Web service interface which defines an interoperable machine-to-machine interactions over the network [19]. Global MMCS is built on the NaradaBrokering [20] which is a general event brokering middleware and supports publish/subscribe messaging model. In the Global MMCS architecture, NaradaBrokering middleware routes events including audio/video data to various communities and collaboration clients. Figure 1 depicts the Global MMCS architecture.

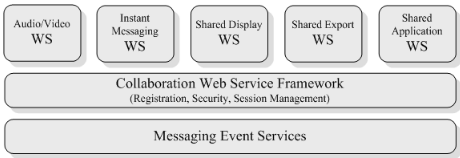

그림 1. Global MMCS 아키텍처 구조
Fig. 1. Global MMCS architecture

Even though there have been lots of research efforts to adopt mobile devices into the conventional system using a proxy, a successful design and implementation to adapt multimedia collaboration system using mobile devices has not been announced yet. In the following section, we will present our design and implementation of mobile multimedia collaboration system using proxy architecture.

III. THE PROXY ARCHITECTURE DESIGN

Our proposed architecture adopts a proxy to transcode message contents and uses an asynchronous mobile publish/subscribe service for its primary communication. We depict our proxy architecture approach in Figure 2.

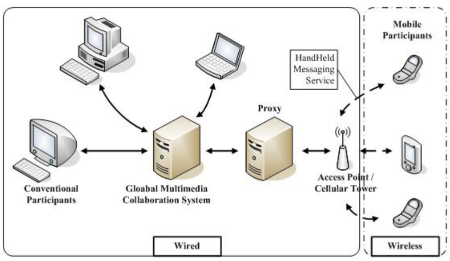

그림 2. 제안 프록시 구조 개념도
Fig. 2. Overview of our proposed proxy architecture

1. Proxy Design Approach

As stated, wireless connectivity and performance of mobile devices has been improved a lot in recent years. However, the computing capability of mobile devices is still bounded to the battery life. The more computation they do and the faster connection they have, the batteries will be consumed more and ultimately the device will lose its mobility. Thus, there is the gap between desktop computers and mobile devices because of the nature of the latter. Thus, if the mobile application off-loads their job to more capable entity, it can be a remedy for mobile application's battery problem. As we discussed in section 2, there have been many research proposals and implementation which adopt a proxy in mobile architecture to reduce the gap by adapting message contents (i.e. transcoding) for mobile devices.

2. Design Characteristics

There are design characteristics which should be considered for mobile application and supporting proxy architecture, since the runtime environments are constraint and consist of heterogeneous networks, devices, and platforms. We describe three design characteristics of our mobile proxy architecture.

The first characteristic is efficiency. As noted, sending and receiving smaller sized messages will reduce the transit time of messages, which is a big gain for high latency and slow wireless connections. Also, by customizing multimedia message content, the prototype system based on our architecture design is able to reduce rendering computation on the client device. Thus, these 'computation offloading' (i.e. server-side computing) to the proxy helps the overall system performance and this is especially true if there is a low powered mobile client in the collaboration system. However, there is a drawback of the approach. Since the proxy connects multiple mobile clients to the conventional system, it could be a performance bottleneck and/or a single point of failure. A more advance architecture design is required to solve this issue such as a cluster of proxy and a load balancing algorithm.

In the architecture design, computation offloading process can be done concurrently with event-mapping. Therefore, we describe the event-mapping procedure instead of computation offloading. In our proxy architecture, the content-adaptor module gets an event message from the Global MMCS server and it is responsible for transcoding (i.e. event mapping). Each content-adaptor, which is application specific, processes the message and delivers a slimmed-down (i.e. in size and complexity) message to the mobile client.

In addition to use proxy design for computational offloading, an optimized application level protocol of HHMS helps to achieve efficiency of the architecture design. GlobalMMCS messages are in XML format and it is heavy for a mobile device which usually runs in constrained environment. We map XML messages into optimized format using the HHMS protocol.

To maximize the efficiency, we define the format of the HHMS message prescriptively. Thus we ship this fixed format message over application level protocol. A message in the HHMS protocol contains five fields: OS type (e.g. Palm, Windows Mobile), application type (e.g. Text Chat, Shared-display, and Shared SVG), event type (e.g. login, logout), length of the event body and event body which is the actual event message. First four fields use one byte per fields and event body has a variable length which is specified in the length field. After mapping, an original XML type message is slimmed down into a prescriptive fixed format message which is smaller in size and complexity.

The next characteristic of our mobile proxy architecture is supporting interoperability. Mobile applications run on various mobile platforms and they are written in various programming languages. As we have seen the problem in many distributed computing environment, the heterogeneity among the collaboration participants cause a lot of customizing issues. Let us list few of them; a direct shipping of in-memory-data maybe not a good idea in many cases. The recipient may not be able to retrieve data correctly because of incompatibility. Also, data format conversion (or transcoding) is required if a mobile platform does not support certain multimedia data format. We need to transform 'A' file format to 'B.' We may face to mixed size or resolution situation among the participants. For example, the sizes of display of mobile devices are different so the shared image should be resized to anticipate the maximum effect. These kind of content adaptation has been explored by many research groups. For collaboration systems, it is a vital design issue to adapt those devices to collaborate together with the conventional PC.

In the architecture, we introduce a XML based user profile on the proxy. When a user log in, a personal server module 'load in' the user profile from a repository. Depending on the user profile and device type of the current machine, the prototype system adapts the content of messages appropriate to the user device. Shared SVG and Shared display applications on the mobile device heavily rely on this idea because those graphic applications on devices with the small display requires event customizing to render images and processing large images costs heavily in computing resources.

The last characteristic is scalability. A proxy can be a bottleneck in the architecture if there is a single point of connection to the conventional collaboration system for entire mobile clients. Thus, the overall architecture of ours must consider clustering multiple proxies to provide load balancing and scalability.

The general clustering for geographically disparate nodes (e.g. clustering of publish/subscribe broker nodes) requires an efficient routing algorithm and a recovery mechanism from failures. Since our architecture is targeted to the collaboration system domain, session and log management are very important to provide seamless multimedia and textual collaboration services. Thus, those features should be considered additionally.

IV. EVALUATION: A PROTOTYPE SYSTEM USING PROXY

To demonstrate the feasibility of our proposed proxy architecture design, we have implemented a prototype with Global MMCS which is design and developed at Community Grids Laboratory of Indiana University. For this prototype, we implement the core part only to make our prototype implementation small enough but not miss any crucial issues for verification. We assume that mobile clients already retrieve their profile form the user profile repository. Thus, we do not implement the user profile repository

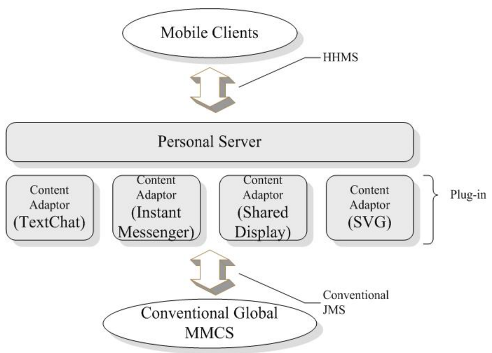

그림 3. 프로토타입 구현의 구조 개념도
Fig. 3. Overview architecture of the prototype

1. Component of the prototype

The prototype consists of a proxy and a lightweight client-side library. A proxy also consists of several sub components: a personal server which manages user state and event routing to the appropriate content adaptor, a content adaptor which does most event mapping (i.e. transcoding), and a connection manager which is a part of HHMS and provides a asynchronous communication channel between mobile application clients and a personal server in a proxy paired. A proxy sits between the mobile clients and the conventional GlobalMMCS server and behaves like a typical GlobalMMCS client to the server. In this case, the GlobalMMCS server does not have to do any special treatment for the mobile clients (i.e. customization/transcoding) and all the necessary customization is done at the proxy level.

A personal server is responsible for the user management in the prototype. It manages the log state of mobile users as well as the user profiles. The typical usage scenario is as follows: 1) a new mobile client log-in to the prototype system. 2) The personal server creates content adaptors and assigns them to the user. 3) Multiple content adaptor modules are created because participants usually use several applications in a single collaboration session. As described in section 3, a user profile is used when a content adaptor creates optimized message for mobile applications. Keeping and loading a user profiles is another function of the personal server related with user management. Another role of the personal server is internal message routing. Since there is a single communication connection (e.g. raw TCP socket or HTTP) per mobile client to the GlobalMMCS server which is shared by multiple content adaptors, the personal server routes event messages internally to proper content adaptors.

Customizing the collaborative messages (i.e. event mapping, transcoding) is done in the content adaptor. It listens to the entire collaborative message on behalf of the mobile participants and maps the event (i.e. message contents) into the optimized format for mobile clients according to the use profile. There should be one content adaptor for each application and it is required to follow HHMS application API to be able to plug-in it into a proxy in the HHMS.

For the communication channel, we implements a TCP socket connection and a HTTP connection. The HTTP without a cookie support or a persistent connection is stateles s and this makes us to add additional works, polling to the Java Servlet. Polling is a good technique to monitor the current state; however it consumes network and processing resource. Thus, we use a TCP socket connection as a default option.

2. Experiments

We experiments four of GlobalMMCS application in our implementation: textchat which uses Jabber API, shared display and shared SVG. Shared SVG application works as follows: 1) the content adaptor component in the HHMS proxy gets the URL of a SVG file and 2) it customizes the image based on the user profile. 3) Next, the personal server sends the customized image to the application on the mobile client over optimized application level protocol (i.e. HHMS) between the proxy and the mobile client device. Shared display application works the same way, except it captures the raw image of a shared display application rather than sharing file URL.

We choose Java as a programming language for prototype (i.e. both a proxy and client library implementation) because it is a platform independent language. So we do not need to port applications and services to make them interoperable. The system environment is summarized in table 2 and the snapshot of the running mobile collaboration application in the prototype is shown in figure 4.

표 4. 구현 환경 요약
Table 2. A summary system environment

PDA (Mobile Client)	Compaq Pocket PC iPAQ H3630 Windows CE 3.0 Strong ARM - 200MHz 32MB 802.11b (PCMCIA)
Server (A Proxy)	Windows XP Intel Centrino duo (1.6GHz) 1GB
Programming Language	Java SE for a proxy prototype Java ME for a client library

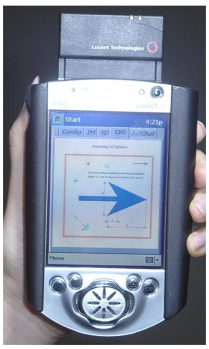

그림 4 제안 구조를 이용해 구현된 모바일 애플리케이션 작동 예 (셰어드 SVG)
Fig. 4. Snapshot of a mobile application in use (Shared SVG) on PDA

3. Discussion

From the experiments with the prototype, we successfully demonstrate the effectiveness of our proposed proxy architecture for mobile multimedia collaboration system. By using the prototype, we can solve problems which we listed in section 1. The comparisons between conventional system and the proposed proxy system are presented in table 2.

표 5. 일반 시스템과 제안 시스템의 비교
Table 2. Comparisons between conventional system and the proposed system

Conventional System	Proposed Proxy System
Computational Burden on Mobile Devices	Computational Offloading to the proxy
incompatible file sizes and formats	transcoded size and formats for heterogeneous mobile devices (interoperable environment)
intermittent wireless communication channel	reliable communication channel by asynchronous application level protocol

V. CONCLUSION

Since using mobile devices increases the level of accessibility, there have been a lot of research proposals and implementations to adapt mobile devices in a collaboration system. However, there are few researches on design and architecture to use mobile devices in the multimedia collaboration system.

In this paper, we investigate the computing environment for mobile collaboration applications and design a simple architecture to integrate mobile devices with the conventional collaboration system by placing a proxy between the mobile devices and the conventional system. A proxy manages user state (i.e. session management), maps the event to customize message content to the given mobile devices, and provide an optimized application level communication channel for intermittent wireless connection.

In order to verify the feasibility of our proposal, we implement the prototype using Global MMCS, the conventional multimedia collaboration system. Our prototype not only performs successfully a session management and delivers multimedia data for mobile devices efficiently, but also shows that the appropriate proxy architecture fits well particularly with multimedia collaboration environments.

We include results of our prototype demonstration instead of scientifical measurements, yet we believe there are enough to show the effectiveness of our design.

REFERENCE

〔1〕 Apple.com, "iPhone Specification," *http://www.apple.com/iphone/specs.html*

〔2〕 Apple.com, "iPod Specification," *http://www.apple.com/ipodtouch/specs.html*

〔3〕 S. Oh, G. C. Fox, S Ko, "GMSME: An Architecture for Heterogeneous Collaboration with Mobile Devices," The Fifth IEEE/IFIP Int'l Conf. on Mobile and Wireless Communications Networks (MWCN 2003), Singapore, pp. 25-28, October, 2003.

〔4〕 W. Wu, H. Bulut, A. Uyar, G. C. Fox, "Adapting H.323 Terminals in a Service-Oriented Collaboration System," Special "Internet Media" Issue on IEEE Computing Magazine, Vol. 9, No. 4, pp. 43-50, July-August, 2005.

〔5〕 A. Uyar, W. Wu, H. Bulut, G. C. Fox, "Service-Oriented Architecture for a Scalable Videoconferencing System," IEEE Int'l Conf. on Pervasive Services (ICPS 2005), July, 2005.

〔6〕 L. Villard, C. Roisin, N. Layaida, "An XML-based multimedia document processing model for content adaptation," DDEP/PODDP 2000, Lecture Notes In Computer Science 2023 Springer 2004, Munich, Germany, pp. 104-119, Sep., 2000.

〔7〕 T. Lemlouma, N. Layaida, "Context-Aware Adaptation for Mobile Devices," The Fifth IEEE Int'l Conf. on Mobile Data Management (MDM 2004), Berkeley, CA, U.S.A., Jan., 2004.

〔8〕 W3C, "Synchronized Multimedia Integration Language (SMIL) 1.0 Specification," *http://www.w3.org/TR/REC-smil/*

〔9〕 A. Chan, S. Chuang, "MobiPADS: A Reflective Middleware Context-Aware Mobile Computing," IEEE Transaction on Software Engineering, Vol. 29, No. 12, pp. 1072-1085, December, 2003.

〔10〕 S. Chuang, A. Chan, J. Cao, R. Cheung, "Dynamic Service Reconfiguration for Wireless Web Access," Int'l World Wide Web Conf. (WWW 2003), Budapest, Hungary, pp. 58-67, May, 2003.

〔11〕 M. Raento, A. Oulasvirta, R. Petit, H. Toivonen, "ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications," IEEE Pervasive Computing Magazine, pp. 51-59, April-June, 2005.

〔12〕 Symbian OS, *http://www.symbian.com/symbianos/index.asp*

〔13〕 Nokia Series 60, *http://www.s60.com/life*

〔14〕 O. Davidyuk, J. Riekki, V. Rautio, J. Sun "Context-Aware Middleware for Mobile Multimedia Applications," The Third ACM Int'l Conf. on Mobile and Ubiquitous Multimedia, (MUM 2004), College Park, MD, U.S.A., pp. 213-220, October, 2004.

〔15〕 M. J. Sinderen, A. T. Halteren, M. Wegdam, H. B. Meeuwissen, "Supporting Context-Aware Mobile Applications: An Infrastructure Approach," IEEE Communication Magazine, pp. 96-104, Spetember, 2006.

〔16〕 G. Chen, D. Kotz, "Solar: An Open Platform for Context-Aware Mobile Applications," The First Int'l Conf. on Pervasive Computing, Dallas, TX, U.S.A., pp. 41-47, March, 2003.

〔17〕 RFC 3261, "SIP: Session Initiation Protocol," *http://tools.ietf.org/html/rfc3261*

〔18〕 AccessGrid, *http://www.accessgrid.org/*

〔19〕 W3C, "Web Services Glossary," *http://www.w3.org/TR/ws-gloss/*

〔20〕 S. Pallickara, G. C. Fox, "NaradaBrokering: A Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids," ACM/IFIP/USENIX Int'l Conf. on Internet Computing, LNCS 2672 Springer 2003, Munich, Germany, Vol. 2, pp. 41-61, Sep., 2000.
〔21〕 박대혁, 홍마리아, 서정만, 이근수, 임영환, "모바일 웹 서비스를 위한 PC 웹에서의 중요 이미지 결정 방법," 한국컴퓨터정보학회 논문지 제11권 제1호, 2006년 3월.
〔22〕 김용태, 정윤수, 박길철, 이상호, "SOAP 메시지 처리 개선과 NBTM을 사용한 모바일 웹 서버의 성능 향상," 한국컴퓨터정보학회 논문지 제13권 제5호, 2008년 9월.
〔23〕 이유리, 박동규, "모바일 환경에 적합한 헬스케어 정보 시스템에서의 역할기반 접근제어," 한국컴퓨터정보학회 논문지 제10권 제3호, 2005년 7월.

저 자 소 개

오상윤 (Sangyoon Oh)
2006년 인디애나 주립대 컴퓨터학과 박사,
2006년 - 2007년 SK 텔레콤 전략 기술부문
2007년 - 현재 아주대학교 정보통신 대학 정보및컴퓨터공학과 조교수
관심분야: 웹/분산 시스템, SOA, Ad-Hoc/P2P 시스템