http://dx.doi.org/10.5392/IJoC.2015.11.2.037

Design of an Integrated Monitoring System for Constructional Structures Based on Mobile Cloud in Traditional Towns with Local Heritage

Byung-Won Min, Sang-Hoon Oh, Yong-Sun Oh*
Division of Information and Communication Convergence Engineering
Mokwon University, Daejeon, 302-729, Korea

Yasuhisa Okazaki
Graduate School of Science and Engineering
Saga University, Saga, 840-8502, Japan

Jae-Soo Yoo
School of Information and Communication Engineering
Chungbuk National University, Cheongju, 362-763, Korea

Sun-Gyu Park
Division of Architecture
Mokwon University, Daejeon, 302-729, Korea

Hwang-Woo Noh
Department of Visual Design
Hanbat National University, Daejeon, 305-509, Korea

ABSTRACT

Sensors, equipment, ICT facilities and their corresponding software have a relatively short lifetime relative to that of constructional structure, so these devices have to be continuously fixed or exchanged during maintenance and management. Furthermore, software or analysis tools should be periodically upgraded according to advances in ICT and analysis technology. Conventional monitoring systems have serious problems in that it is difficult for site engineers to modify or upgrade hardware and analysis algorithms. Moreover, we depend on the original system developer when we want to modify or upgrade inner program structures. In this paper, we propose a novel design for integrated maintenance and management of a monitoring system by applying the mobile cloud concept. The system is intended for use in disaster prevention of constructional structures, including bridges, tunnels, and in traditional buildings in a local heritage village, we analyze the status of these structures over a long term or a short-term period as well as in disaster situations. Data are collected over a mobile cloud and future expectations are analyzed according to probabilistic and statistical techniques. We implement our integrated monitoring system to solve the existing problems mentioned above. The final goal of this study is to design and implement a monitoring system for more than 10,000 structures spread within Korea. Furthermore, we can specifically apply the monitoring system presented here to a bridge made from timber in Asan Oeam Village and a traditional house in Andong Hahoe Village to monitor for possible disasters. The entire system design and implementation can be developed on the LinkSaaS platform and the monitoring services can also be implemented on the platform. We prove that the proposed system has good performance by performing a TTA authentication test, web accommodation test, and operation test using emulated data.

***Key words**: Constructional Structure, Integrated Monitoring System, Mobile Cloud, Traditional Town, Local Heritage.*

1. INTRODUCTION

Comparing lifetime of constructional structure with its attached ICT HW and/or SW including measuring tools, sensors, and other peripheral facilities, we can easily recognize

** Corresponding author, Email: ysunoh@mokwon.ac.kr*

Manuscript received Mar. 31, 2015; revised Apr. 22, 2015; accepted Apr. 29, 2015

much shorter lifetime in measuring ICT facilities so that we should periodically exchange or modify these attached facilities during the lifetime of structure itself. In addition, periodical upgrade in the monitoring system should be essential because we need to apply advanced software or analyzing technology to these maintenance or management processes. On the other hand, conventional monitoring systems for constructional structure generally have their blind characteristics in software so that any maintenance personnel should depend upon the original developer when they need to modify or upgrade software and analyzing algorithm because it is difficult to understand the inner contents of that programs [1]-[3]. This problem has been one of the most serious ones in the world of monitoring system of constructional structure.

Currently, almost all monitoring systems for constructional structure in Korea are using Client/Server or ASP environment for their software supply. However, such systems are difficult to manage and require high operating costs because of the problem related to high costs of HW, SW, and their installation, distribution, customization, upgrade, fault & tolerance management, and expensive license royalties [4]-[7].

Analyzing the movement of constructional structure in the status of common and/or special, we need to expect the future movement of the structure based on this analysis mentioned above. It will be more important these days to expect the movement of structure in the near future. We should apply probability theory and statistical technique to the problem and collect their analytical results in a real-time manner [8]-[10]. For the sake of these purposes, we should stack a lot of believable data for a long time, nevertheless it has been known as a really difficult work in the environment of domestic maintenance field of constructional structure [11].

In this paper, we present a novel concept of collecting data from maintenance and management of constructional structure based on mobile cloud concept. We design and develop an integrated monitoring system for measuring the movement of constructional structure [12], [13]. Collecting useful data and evaluating these large data on confidence levels by statistical analysis methods, we propose a novel concept of integrated monitoring system for constructional structure and their safety and/or security. We design and implement a monitoring system based on mobile cloud or web version which can replace the conventional system of the type C/S or ASP. We can, of course, obtain a lot of benefits and strong points basically given by web-based or mobile cloud systems.

The rest of this paper consists of the followings; In Section 2, we present the concept of our integrated monitoring system for constructional structure based on mobile cloud. We also compare the differences between C/S environment and mobile cloud when applying to the monitoring system for traditional town with local heritage. Next we describe the procedure of developing the system with step by step manner in Section 3. Developing environment, LinkSaaS platform and components are briefly explained. In addition, the framework of our system including server/clients, services, and their considerations is presented. In Section 4, we test our system on the basis of TTA authentication. Finally in Section 5, we put concluding remarks and introduce some future works expected.

2. DESIGN OF INTEGRATED MONITORING SYSTEM MEASURING STRUCTURAL MOVEMENTS

2.1 Monitoring System Concepts

There has been a serious problem in dealing with conventional monitoring system for constructional structure for a long time. As the inner characteristics of SW and/or HW are not apparently open, it is generally difficult to modify or upgrade the analytical algorithm and program on the side of management or maintenance consideration. We frequently depend on the original developer to do this work even when elementary or trivial modifications are concerned. In addition, all the software of this conventional monitoring system are offered by ASP(application software provider) or even C/S(client/server) type environment so that costs of HW, SW, installation, distribution, customization, upgrade, management of crack problem, and cost of license etc. can be continuously increased. Especially there are a lot of problems in software management including high cost and effort in their monitoring affairs [14], [15].

To overcome these existing problems, we propose a new concept of integrated monitoring system for constructional structure based on mobile cloud as shown in Fig.1. The only difference from conventional monitoring system using C/S or ASP concept is that new motoring software is supplied as a service through communication network or mobile cloud.

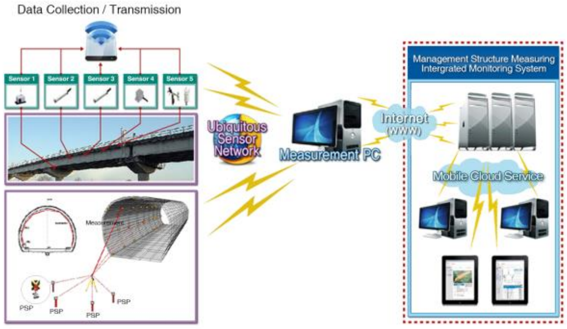

Data Collection / Transmission
Sensor 1
Sensor 2
Sensor 3
Sensor 4
Sensor 5
Measurement
PSP
PSP
PSP
PSP
Ubiquitous Sensor Network
Measurement PC
Internet (WWW)
Management Structure Measuring Intergrated Monitoring System
Mobile Cloud Service

Fig. 1. Concept of Integrated Monitoring System for Constructional Structure Based on Mobile Cloud

According to the benefit of level-3 multi-tenant SaaS system concept, standardizing automated measuring equipment, cost of referring start is very low, easy and simple so that two kinds of mobile cloud service models like ‘hosted application’ style and ‘software on demand’ style can be applied [16], [17]. Based on these concepts of mobile cloud, we design the integrated monitoring system Hardware including measuring sensor, fixed & dynamic data collectors and Software including analyzer and expectation so that modification and upgrade indicated above can be easily performed by management personnel. We focus on monitoring of structures like bridge and tunnel especially in traditional town with local heritage.

Our new concept of the integrated monitoring system depicted in Fig. 1 consists of 3 main parts. Measuring machinery part includes a lot of sensors and automatic measuring tools which spread over each part of a bridge or

PSPs(point of standard prism) on a tunnel. Communication machinery part consists of transmission equipment, exclusive communication cables, and USN(ubiquitous sensor network) with its peripherals. Analysis and Storage part consists of computerized memory and monitoring software for any infinitesimal variations or movements of the target structure in a long or short term period. Automatic measuring system should always perform essential measurements and transmit basic data to the central measuring computer through USN. In addition, all the measuring computers are connected with each other through internet and they are all served by mobile cloud. Triggering system occasionally starts to store abnormal variation data of the target structure. If a strange symptom occurs, it reports and interprets the results as a dynamic variation analysis. Total system supports to recognize any kind of durability variation of the target structure such as bridge, tunnel, and other constructions by means of comparison analyses.

2.2 System Architecture

Fig. 2 shows two different architectures of monitoring system for constructional structure. Left one indicates conventional Client/Server type architecture and right one is our Mobile Cloud type architecture of monitoring system. Considering upgrading easiness, user-friendly interface, and other cost benefits indicated above, we convert all the design of C/S type integrated monitoring system into mobile cloud type.

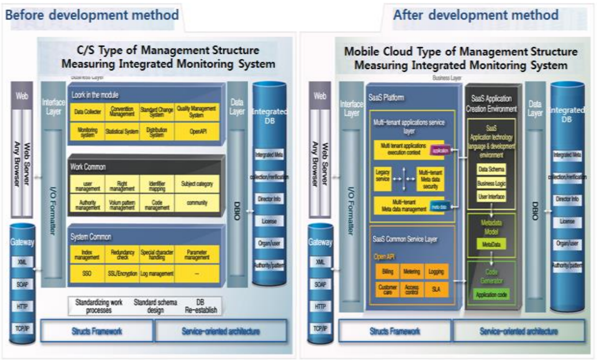


Fig. 2. Comparison of System Architectures between C/S Type and Mobile Cloud Type for Structure Monitoring System

Management structure measuring integrated monitoring system designed in this paper to support a SaaS-based multi-tenant environment is composed of a multi-tenant application service layer, a SaaS(software as a service) common service layer, and a SaaS application generating environment layer as shown at the right hand side in Fig. 2.

First of all, multi-tenant application service layer performs multi-tenant metadata management function, legacy service interface function, multi-tenant metadata security function, and multi-tenant application execution environment management function; On this layer, metadata interface single-tenant applied execution technology, metadata interface multi-tenant applied execution technology, and multi-tenant load balancing technology are to be developed.

Second, SaaS common service layer is an open API (application programming interface) which performs customer management, billing management, access management, SLA (service level agreement) management, metering, and logging management functions. It actually applies SaaS platform customer management and billing, SaaS service access control and SLA monitoring, SaaS service metering, and SaaS logging technologies.

Finally, SaaS application generation environment layer performs data schema, business logic management, SaaS application language and development environment function for user interface, metadata model management, and code generation functions. Applied technologies of this layer include SaaS application description language, SaaS metadata modeling, SaaS code generator, SaaS application development environment, and SaaS platform interface testing environment technologies etc.

3. DEVELOPMENT OF PLATFORM AND SERVICES

3.1 Development Environment

We develop management structure measuring integrated monitoring system on the LinkSaaS platform depicted in Fig. 3. The lowest basement is a SaaS platform named Nereus^{TM} which provides with API; the second basement is Spring framework which consists of AOP, ORM, DAO, MVC, Web, Context etc.; and the top of this environment is designed for application development which consists of Web(JSP, JSTL, HTML, CSS, JavaScript, and XML), Business Logic(web manager, central manager, field manager, and general user service etc. iBatis and Oracle are attached side by side on the top layer.

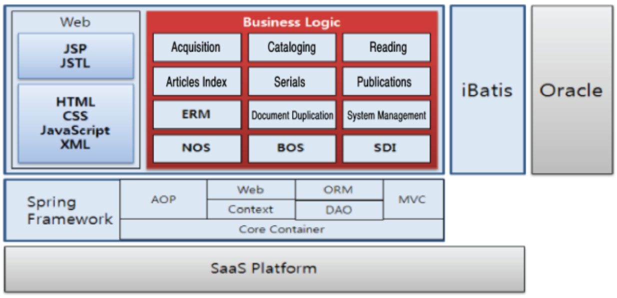


Fig. 3. Development Environment

3.2 Platform

LinkSaaS application platform is a software infrastructure designed to accept large scale data with cost-effective save, supercomputer-like capacity to process, free transform, and acting like a channel for our benefits.

This platform performs these excellent behaviors by leveraging spare capacity of computing resources on the “edge” of the Internet, which generally use only a small fraction of their capacity even when they are busy for marginal operating cost with environmental impact. It can be allowing users and developers to combine and coordinate with each other in novel and dynamic ways. In order to assure the safety of these edge resources, LinkSaaS application platform is built upon Java applets. For the best established security context for running untrusted or third-party code on the Internet, the use of this technology also seamlessly extends the reach of LinkSaaS platform across different platforms, operating systems, and even enterprises. In principle, the LinkSaaS platform network can tentatively encompass all the world's network-attached computers, providing an aggregate capacity 100 times greater than 500 most powerful supercomputers combined.

Purpose of LinkSaaS architecture is twofold: First, LinkSaaS is designed to increase productivity of installed base of computing devices. As noted above, most of computers in the world are commonly using in the range of extremely small part of their computing capacity even when they are "busy" status like editing a letter, working on spreadsheet, or browsing the Web. LinkSaaS aims to harness this spare capacity for general computing situation. However, there are significant technical obstacles to realize this purpose. For this reason, LinkSaaS is designed to be simple to join, leave, and administer with the resource owners remaining firmly in control of their computers. Moreover, it must satisfy reasonable security expectations for running untrusted, third-party code. Second, LinkSaaS is intended to increase the productivity of developers' creating new applications for this large network. Distributed programming has been historically problematic fraught with tedious details. It has not been usually trust worthy of the effort except in extreme cases. The easiest situation is where the developer only has a single computer cluster under his influence, but this cluster may not be large enough. Once administrative boundaries are crossed even within the same organization, the developer must keep track of a variety of configurations, accounts, and security mechanisms. In addition, the code usually has to be built and installed on platforms which all differ, even if only slightly interconnected with the tedious details. The code then has to be activated and the network configuration including the presence of firewalls must be considered.

LinkSaaS addresses the problem mentioned above by specifying a streamlined environment for programming and deploying code across the network based on Java applets. This environment will be described in greater detail below. With these considerations, platform developing, deploying, and provisioning a large-scale network applications become realistic and practical.

3.2.1 Description

In this subsection, we describe the LinkSaaS architecture in somewhat detail. Its components are described first, followed by descriptions of how developers, users, and administrators see the resulting system. Every attempt has been made to keep LinkSaaS simple and straightforward to join, leave, program, and administer.

LinkSaaS network itself provides a basic infrastructure which makes the network programmable like a peer-to-peer network while still respecting administrative constraints. It is not intended to make a distributed system like a single computer but the authors suggest that this level of abstraction is best approached through software built on the LinkSaaS infrastructure. Moreover, the single computer abstraction is neither necessary nor appropriate in many circumstances; instead, programmers should be free to use the network creatively, and to construct network configurations appropriate to the nature and scale of the problems they are tackling.

To the extent that the LinkSaaS architecture is based on services which can be combined to form other services, it will be seen to bear some relation to a service-oriented architecture (SOA). However, LinkSaaS services do not have to reside on big and expensive servers; instead, they are provisioned by resources on the edge of the network. Also unlike a typical SOA, LinkSaaS specifies a streamlined remote service deployment model, facilitating dynamic highly distributed service provision. Thus it relieves programmer to be some of the most tedious aspects of practical distributed programming.

On the other hand, LinkSaaS relies on Java virtual machine (JVM) with its built-in security mechanisms to accomplish remote deployments safely; it should be noted that Java applets have been running untrusted and as a third-party code in web browsers for over ten years. JVM allows compiled code to run under Windows, Linux, Mac OS, and a number of other operating systems as well as on hardware ranging from typical x86-based PCs to ARM-based mobile phones. The speed penalty for running code in the JVM has also largely vanished due to HotSpot runtime compilation into native machine code. By building on Java technology, LinkSaaS can offer secure, high-performance, and cross-platform computing.

3.2.2 Components

Fig. 4 shows a block diagram of LinkSaaS network. It is basically divided into administrative domains, each of which contains one or more LinkSaaS servers. And each server is in turn connected to the Internet, through which it communicates with other servers as well as generic web servers. LinkSaaS clients then are connected to one or more LinkSaaS servers; in principle, they could connect to any server in a different domain if their network access allows it.

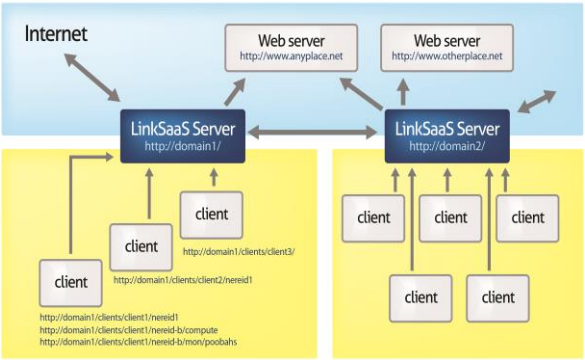

Internet
Web server
http://www.anyplace.net
Web server
http://www.otherplace.net
LinkSaaS Server
http://domain1/
LinkSaaS Server
http://domain2/
client
http://domain1/clients/client3/
client
http://domain1/clients/client2/nereid1
client
http://domain1/clients/client1/nereid1
http://domain1/clients/client1/nereid-b/compute
http://domain1/clients/client1/nereid-b/mon/poobahs
client
client
client
client
client

Fig. 4. Block Diagram of LinkSaaS Network

It can be seen that LinkSaaS network is an overlay network built on the Internet itself. LinkSaaS servers are identified in public by their Internet addresses. Since LinkSaaS servers and domains are fundamentally independent of one another, the network is scalable in the same sense that the Internet itself is scalable. The network within a LinkSaaS domain may be different, however, and clients may or may not have their direct access to the Internet because of a firewall or its implementation on an appropriate network for instance.

Even so, LinkSaaS clients will remain accessible to the network operation.

3.2.3 LinkSaaS Server

LinkSaaS server is the lynchpin of all communications throughout the network: any client can join LinkSaaS network by connecting to a server, and the server acts as a proxy for the clients in communication with the wider network as well as with other clients as shown in Fig. 5. All the clients in the network can communicate with wider networks even if they are protected behind a firewall. Because what is only exposed to the other wider networks should be a server, clients can just be connected to wider world including other servers or clients within a safety zone of networking.

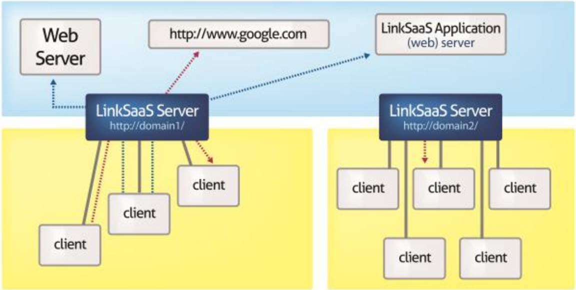

Web Server
http://www.google.com
LinkSaaS Application (web) server
LinkSaaS Server http://domain1/
LinkSaaS Server http://domain2/
client
client
client
client
client
client
client
client

Fig. 5. An Example of Communication Showing the Central Role of LinkSaaS Server

Within an administrative domain, a LinkSaaS server should be maintained and monitored as any central service such as an e-mail service or an institutional web service. In turn, because of its central role in the network, the server can act as an administrative control point for its clients, monitoring their bandwidth usage and limiting it if necessary.

Thus, even though clients can communicate with one another as if they were participating in a peer-to-peer network, the administrator can maintain a useful degree of control over their resource usage. LinkSaaS architecture, therefore, addresses the common shortcomings in peer-to-peer network operation. LinkSaaS server manages the namespace of its domain, in other words, it assigns names to all the clients connected to it. These names are globally unique by construction as long as LinkSaaS servers maintain unique IP addresses, which is an assumption of the Internet itself.

3.2.4 LinkSaaS Clients

LinkSaaS clients can be seen as workhorses of the network. They are programmable elements out of which developers create applications. A client itself is a program which runs on a host computer: it can be started from a web browser (applet client), run as a user program (stand-alone client), or installed as a persistent service (enterprise client). It is possible to have several clients running on a single computer. In this case, each client will have a unique identifier with which it identifies itself to the corresponding LinkSaaS server.

A LinkSaaS client downloads its first service descriptor from the corresponding LinkSaaS server. This service descriptor tells the LinkSaaS client what to install in its root container. As mentioned above, the client can connect to more than one server simultaneously by initializing a root container with a different service descriptor for each server in the case of stand-alone or enterprise client, since each server of which potentially in different administrative domains can implement different domain policies. It is just a responsibility of the corresponding server to decide whether or not to accept a connection from a particular client.

Security context of an applet client is shown in Fig. 6. In this case, the entire LinkSaaS client is encapsulated within a Java applet and its security manager. So the service contained in a LinkSaaS client is prevented from reading local files or making arbitrary Internet access without interaction of the corresponding server

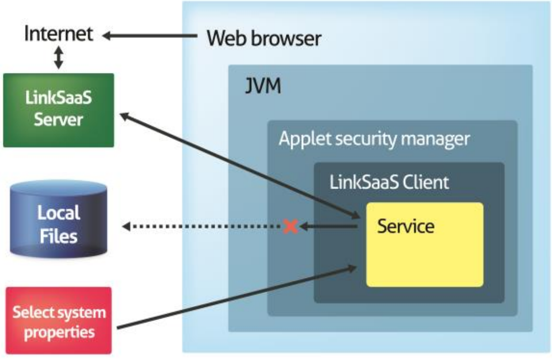


Fig. 6. Security Context of a LinkSaaS Applet Client

3.2.5 Application Server

LinkSaaS clients are able to access normal web resources via the corresponding LinkSaaS server. This feature allows them to access not only the whole World Wide Web for information or other functionality but it also allows the developer a convenient method to control his applications.

As noted above, LinkSaaS does not specify a particular application model, but a particularly simple model used already for several applications consists of a web server or application server which provides a work queue and a user interface through which a user submits to it. Clients are set up to poll the application server with simple HTTP GET requests for work to do. Results can be sent back via HTTP POST operations or stored until fetched and cleared by the application server.

Application server is not a part of LinkSaaS network and it can be indeed written without any reference to LinkSaaS code; its interaction with LinkSaaS clients can be entirely managed through normal web and socket connections can be made without LinkSaaS server. Application server, therefore, can be implemented using simple CGI scripts on an existing web server, though more flexibility may be obtained using streamlined web server infrastructure software such as Grizzly or LinkSaaS’s own NIONetworkHandler class. It should also be noted that a LinkSaaS service could also be a function of an application server.

3.2.6 LinkSaaS Service

A LinkSaaS service is a program that can be installed and run on a LinkSaaS client. Services can perform data processing, serve web content, and communicate with other services or external web sites. In addition, a LinkSaaS service can create child containers into which other LinkSaaS services can be installed. A LinkSaaS service runs on a LinkSaaS client, not on a server. However, it can be said that a LinkSaaS server which provides the root of the service's public URL is what offers the service after all, even though the actual execution of the service is on another computer.

A LinkSaaS service is initialized by pointing the client to a web-accessible service descriptor in the form of a markup language document named NML. This descriptor can be located on any web server, not necessarily the application server or any LinkSaaS server, though NML document describing the root container resides on the corresponding LinkSaaS server by the design.

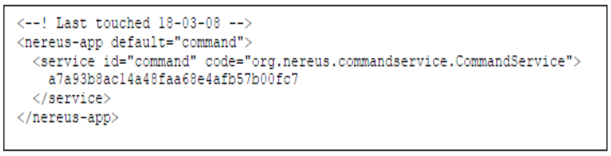
LinkSaaS Markup Language(NML) has an XML-like syntax with three tags defined as follows:

- <LinkSaaS-app> defines the application's name and encompasses all the class loaders and services it uses.
- <classloader> defines the location of class hierarchies and JAR files for services defined within its scope.
- <service> defines an actual service with a name and a main class.

Data between the opening and closing tags are passed to the service's initialization method as a character sequence (CharSequence).

<classloader> tag is optional; any services defined outside a <classloader> scope is assumed to find classes starting from the service descriptor's URL directory. It is often convenient for any necessary class or JAR files to reside alongside the service descriptor, but the <classloader> tag allows applications to draw upon classes from a variety of sources. An example of an NML document describing a (CommandService) and an often-used root container are shown below.

In this example, the application is called "command" and its Java class file is found relative to the server's default web directory. The data between <service> tags are passed to the service on initialization. They are primitive authorization strings in this case. An NML document can describe multiple services with different code sources. Services may be combined into larger services by including references in the initialization data whether the other service names are within the same NML document or using URL's of external services. An auto-configuration of LinkSaaS service may be employed to generate further service descriptions as needed.



```
<--! Last touched 18-03-08 -->
<nereus-app default="command">
  <service id="command" code="org.nereus.commandservice.CommandService">
    a7a93b8ac14a48faa68e4afb57b00fc7
  </service>
</nereus-app>
```

Fig. 7. NML Document for CommandService Descriptor

3.2.7 LinkSaaS API

The API(Application Programming Interface) of a LinkSaaS service is much like that of a Java applet: if it is deployed in a standard JRE then it can use any of that JRE's libraries. However, the usual applet security restrictions apply. In other words, it cannot modify system properties, access local file system directly, or open network connections except to the server from which the applet was originally downloaded. Actions beyond these cases must be authorized by client and implication.

Most basic LinkSaaS service interface (LinkSaaSService) specifies only an init() entry point, which is called when the service is created. The name of the service is determined by the service descriptor; the client delegates URL's with this path element to the service.

An AbstractService class has been provided to make it easier to write new LinkSaaS services. AbstractService provides two abstract entry points like prepareToListen() and handleConnection(). The former is invoked when the service is created. It can be invoked by AbstractService.init() method. The later, on the other hand, is called whenever a connection is made to the service from outside including the times when a user clicks into the service through the LinkSaaS server.

Because of LinkSaaS client's applet-like constraints, several methods have been provided in AbstractService to facilitate communication with outside world. First, println() and printErr() methods have been provided to print messages to output and error panels in LinkSaaS client's user interface. Web access must be redirected through the LinkSaaS server, which serves as a web proxy for its clients: a createRedirectedURI() method is provided to create the proxy URL from an original URL. A socket proxy is also provided.

The common application pattern described above is implemented on the service side using prepareToListen() to create a new thread which periodically polls the application server whose web address has been transformed by createRedirectedURI()) for work to do. In this case, handleConnection() can be used by the application server to fetch stored results, or by other LinkSaaS services to fetch intermediate results. It can also respond to requests for status information, for instance by users' clicking through the LinkSaaS server web page.

3.2.8 Network and Namespace

One of the key benefits of LinkSaaS network to perform a network programming is to define a straightforward namespace by which elements of the network can be addressed. LinkSaaS namespace consists of URL's, all of which are valid for public web access.

Each LinkSaaS server has a public URL such as http://domain1/, where "domain1" refers to the LinkSaaS server host. Since a LinkSaaS domain typically have one or a few LinkSaaS servers and policies are determined on a domain level and implemented by the servers, it is appropriate (though not necessary) to designate LinkSaaS servers with domain names. Since LinkSaaS servers are publicly accessible Internet nodes, their names must be registered in the normal way.

LinkSaaS server provides several web pages, such as http://domain1/clients, which lists the clients connected to it. When clients connect to a LinkSaaS server, it creates a root container into which the default service enters. In addition, this container described by DomainRoot.nml is also installed. This service is addressed by appending the client aliasing to the clients' subdirectory of the corresponding server URL, e.g., http://domain1/clients/client5. The client then delegates this interpretation of any further path elements to the service. Thus, if the service has created child containers in which it has installed services, those services are referred to in further path elements.

In principle, the entire LinkSaaS namespace is accessible via the Web. If a service needs to address another service, it creates a redirected URL out of the other service's public URL, just as if it was a normal web resource.

3.2.9 Applications

Software delivered as a service offers distinct advantages over software delivered by more traditional means including C/S and ASP styles. SaaS frequently supports mobile, web-based, centrally managed, and nearly free of troublesome installations and patches. The architecture of this new style implied by these features consists of many clients attached to a large server infrastructure in its simplest form. Increasing number of users as well as improved functionality which generally impact the server load therefore derives server requirements with increasingly expensive outcomes. Reducing server load, therefore, decreases the overall cost of the system, but risks are substantially increased for operational costs.

Fig. 8 indicates methods and processes of interactions within a LinkSaaS application. Once a user requires the server an application through a click on the link, a command is transferred to the client to make a browser active. Client loads a container named Nereid to locate an application code and its configuration. Then server accesses LinkSaaS cloud to obtain the application to meet the code and configuration in Nereid.

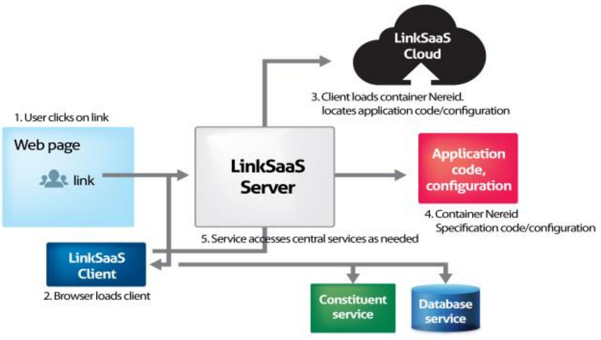

LinkSaaS Cloud
3. Client loads container Nereid. locates application code/configuration
1. User clicks on link
Web page
link
LinkSaaS Server
Application code, configuration
4. Container Nereid Specification code/configuration
5. Service accesses central services as needed
LinkSaaS Client
2. Browser loads client
Constituent service
Database service

Fig. 8. Interactions within a LinkSaaS Application

3.3 Possible Monitoring Application and its Folk Village Considerations

Fig. 9 shows an application of SaaS and mobile cloud on structure measuring and management. An integrated monitoring system based on mobile cloud in traditional town with local heritage should be presented. Houses, bridges, tunnels, and/or other constructional structures in folk villages can be monitored using the method proposed in this paper

Implementation of data structure focuses on stable collection of data, processes of mal-state of structures, and easy checking. Design of integrated manager focuses on checking total state of affairs for the target structure, analysis of each data from USN for structures, and convenience of report management.

There are totally 3 parts of main managers in our monitoring program as shown in Fig. 9. After detecting mal-function data from sensor network installed on various points of structures, we apply virtualization on CM(central manager) and FM(field manager) of the monitoring system to overcome problems in operation because the system is physically distributed far apart. We can easily collect these correct data from site information at the center, and it will be possible to respond these real-time occurring and extend our effectiveness of operation.

Of course, these functions essentially used in monitoring system can be implemented by C/S type or ASP type that conventional maintenance systems are taking. But serious problems have been occurred when maintenance or management affairs for novel software or others are needed.

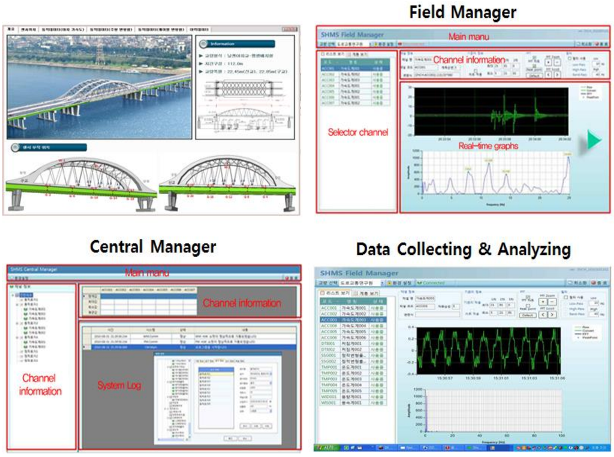


Fig. 9. Applications of Mobile Cloud to the Integrated Monitoring System for Constructional Structure (Managers)

Mobile cloud or 3-level SaaS mechanism can get over these problems and provide users and field engineers a novel concept of convenient service that emphasizes Web service search, accessibility, usability, contents creation and recombination, interoperability and a personalized service.

To develop our integrated monitoring system for constructional structures using the concept of mobile cloud, an open Web conforming to the W3C standard is applied as follows;

- Web page supporting all browser versions of Netscape 7.x, MS IE 6.x, and Firefox 1.x or higher (cross browsing supported)
- Various operating environments: Web accessibility and usability
- Web service environment with consideration to generality and scalability
- Standardization: mutual interfaces of the Web service and integrated usage
- Operation standard guidelines are applied for administrative agency homepage development as an operation standard guidelines
- W3C(World Wide Web Consortium) presents Web standard technology recommendations
- Web service standard architecture: XML, UDDI, WSDL, and REST

- Information System Development and Operation Technical Guidelines 2.0 are applied

In addition, various mash-up services using open API must be considered. And finally, user authentication and access privilege management must be provided using the homepage login linking (SSO).

3.3.1 Asan 'Oeam' Village

For the sake of possible applications of the monitoring system proposed here to traditional towns including local heritage, we investigate two representative folk villages in Korea. One of them is Asan 'Oeam' Village shown in Photo 1. What we can be focusing on this folk village is that there are a lot of objective structures to be monitored like bridges, houses, and other facilities.

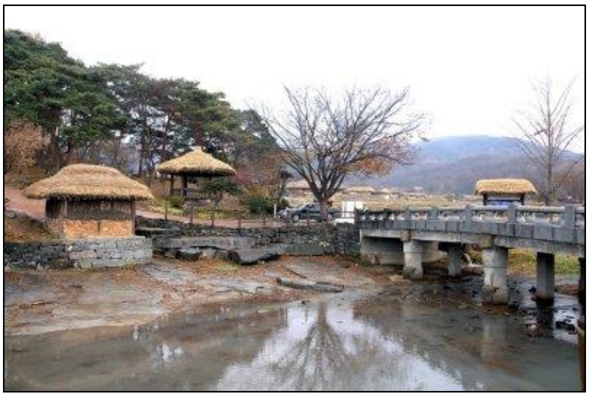


Photo 1. A Folk Village Asan 'Oeam' Having Lots of Traditional Structures Needed Maintenance Monitoring

Geographically, Asan 'Oeam' Village is located in ChungcheongNam-Do. It was established about 500 years ago, of which total area is about 198,194m^2 and more than 150 residents are living in the village [18]. There are a lot of inheritances like 'BangA', 'JangSeung', and other traditional folk antiquities. 74 old style buildings constructed by timber structure, 2 stone bridges, and 3 wooden bridges are preserved in the village [19]. But they do not have any countermeasure of modern style against natural or artificial disasters like heavy rain, land slide, and earthquake etc. so that they can be disclosed to the possibilities that crack, collapse, or even vanish occur.

We propose the monitoring system designed throughout this paper applying to these traditional constructional facilities as a good counterplan against the disaster. Measuring, maintenance, and management of any constructional structures include cleaning, safety diagnosis, repairing, reinforcement, and restructuring etc. that we can obtain by applying the integrated monitoring system. Fig.10 shows an example of application that the monitoring system is established at the critical spots of a timber bridge located in Asan 'Oeam' Village.

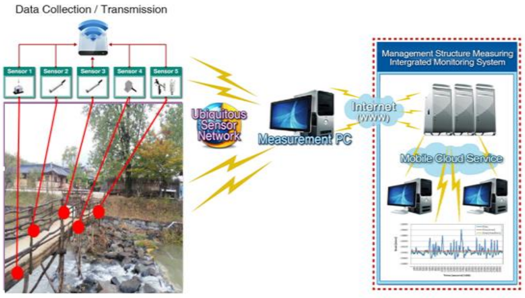

Data Collection / Transmission
Sensor 1
Sensor 2
Sensor 3
Sensor 4
Sensor 5
Ubiquitous Sensor Network
Measurement PC
Internet (WWW)
Management Structure Measuring Intergrated Monitoring System
Mobile Cloud Service

Fig. 10. An Application of the Monitoring System to a Structure in Asan 'Oeam' Village

Sensors like elasticity, fluctuation, luminance, virtual, and accelerometer are installed and connected throughout the USN so that we can collect various sensing data and transmit them to both static and dynamic measuring tools to stack. We can automatically lookout beforehand its possible variation of safety, progress of damage, and/or even small crack by processing the accumulated data by the algorithm of component extraction. In addition, we expect that managers of the village can collect safety data of facilities, investigate status, and counterplan against any abnormal situations using the integrated monitoring system of this paper.

3.3.2 Andong 'Hahoe' Village

Photo 2 shows a bird's eye view of Andong 'Hahoe' Village as a whole. The name of the village 'Hahoe(河回)' means a river turns around it like a ring as shown in the Photo. Andong 'Hahoe' Village was registered in UNESCO World Cultural Heritage in 2010. It had been established about 600 years ago [20]. There are 125 traditional houses and 232 residents are living in the village. They say that more than 1 million tourists are visiting the village per year these days [21].

Most of the houses in Andong 'Hahoe' Village are mainly made by timber structure and separated by truly narrow alleys between them like any other folk villages in Korea.

Photo 2. An Overview of Andong 'Hahoe' Village

All the roofs of the houses in the village are 'Choga' or 'Kiwa'. Backbone of 'Choga' roof is basically made by timber structure and covered with straw thatch. 'Kiwa' roof, on the other hand, has its own fire-resistance so that danger of damages from external source can be relatively lower than

‘Choga’ roof. But what we should be concerned about it is that there are a lot of small wood sticks inside the ‘Kiwa’ roof in the village [22]. As a result, we can easily recognize they have a lot of weak points against fire disaster. Moreover, once a fire occurs, it should be difficult to extinguish because of their structural properties as well as their close positions with each other.

Although there has been very small possibility of natural disasters and they are saying that the area of ‘Hohoe’ Village is naturally firm and lucky geometrical environment, we can expect serious fire and/or artificial disasters to threat its safe conservation. Especially, ‘Ankang’ elevated bridge located at the entrance of the village, ‘Water Tunnel’ connected between houses in the village, and a lot of old houses having narrow alleys are apparent objectives of our monitoring system applications. Fire-fighting facilities installed in the village are hydrants, flash suppressors, and chemical fire extinguishers. No modern style integrated monitoring system for disaster like fire is not established.

Fig. 11. shows an application of our integrated monitoring system to detect symptoms of fire disasters in the structures of ‘Choga’ and/or ‘Kiwa’ houses.

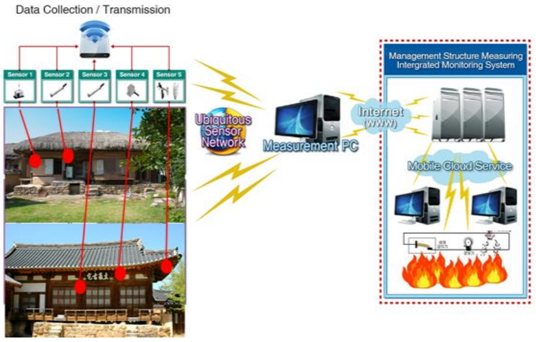

Data Collection / Transmission
Sensor 1
Sensor 2
Sensor 3
Sensor 4
Sensor 5
Ubiquitous Sensor Network
Measurement PC
Internet (WWW)
Management Structure Measuring Intergrated Monitoring System
Mobile Cloud Service

Fig. 11. An Application of the Monitoring System to Detect Fire Symptoms for Traditional Houses in Andong ‘Hahoe’ Village

It is apparently possible to apply our integrated monitoring system for traditional houses in Andong ‘Hahoe’ Village against fire disasters using the formation and process depicted in Fig.11. Heat, smoke, gas, flame, and/or temperature sensors can be established and connected throughout the USN so that we can collect fire alarm sensing data and transmit them to automatic measuring equipment. Using these accumulated sensing data to be processed by component extraction algorithm, we can automatically detect beforehand variations of safety, progress of damage, and/or symptoms of fire occurrence.

4. PERFORMANCE ANALYSIS

In this section, we present procedures, items, and results of performance analysis of the integrated monitoring system proposed in this paper. We conduct our performance test as 3 separated steps as follows;

1) TTA Authentication Test for LinkSaaS platform

2) Web Accessibility, Compatibility, and Openness Test for Application Programs

3) and Operation Test Using Real Measuring Data

4.1 TTA Authentication Test for LinkSaaS Platform

As shown in Table 1, items of TTA authentication test for our LinkSaaS platform consist of SaaS maturity and performance items. Application program functions are not for the platform itself but for applications installed on the platform. Therefore, we present the test results of these applications in the next subsection which deals with the 2nd step of our performance analysis.

SaaS maturity items indicate whether the platform reaches to level-3 maturity of SaaS or not so that we can submit ‘multi-tenant support’, ‘same code support’, ‘customizing support’, and ‘data sharing support’. In addition we can extend the most important ‘multi-tenant support’ item into several detailed items as show in Table 2. On the other hand, we can investigate performance items as basic operating speed of the platform so that we can submit ‘CPU utilization rate’, ‘memory utilization’, ‘response time’, and ‘portability’ etc. We should prove our platform satisfies the minimum TTA Authentication criteria for the first 3 quantitative items and 1 qualitative item.

Table 1 Items of TTA Authentication Test

Categories	Detailed Test Items
SaaS Maturity	-. Multi-tenant Support -. Same Code Support -. Customizing Support -. Data Sharing Support
Performance	-. CPU Utilization Rate (%) -. Memory Utilization (Mbytes) -. Response Time (Sec.) -. Portability (Yes/No)
Application Program Functions	-. Web Accessibility ▪Easiness of Recognition ▪Easiness of Operation ▪Easiness of Understanding ▪Robustness -. Web Compatibility ▪Observance of Web Standard Grammar ▪Guarantee of Web Compatibility -. Web Openness ▪Robots.txt Blocking ▪No Index, No Follow Blocking ▪Redirecting to Install ActiveX and Java ▪Java Script Errors

Each item is evaluated following the test methods and criteria officially regulated by TTA. We could obtain results of the authentication test for our LinkSaaS platform as shown in Table 2.

Table 2. Results of TTA Authentication Test for the Platform

Performance Test Items	Unit	Measurement Results
1. SaaS Maturity Model	**Level**	**Level-3**
① Connection	Instance	N(customer) : 1(instance)
② Provided program	Code	Same code
③ Customizing	Configuration	Configurable by the customer for each tenant
④ Economy of scale	Yes/No	Yes(instance sharing)
⑤ Scaling	Yes/No	Yes(Replacement to better performing system)
⑥ Data	Format	Data sharing

2. Performance Test	Class	Class-A
① CPU utilization rate	%	8% or less
② Memory usage	Byte	500MB or less
③ Response time	Sec	1 Sec
④ Portability	Yes/No	Yes

After they certify essential criteria for Multi-tenant SaaS system such as N-to-1 connection in the cloud, same code property, customer configurability, positive economy of scale due to instance sharing, performance scaling due to system replacement, and data sharing format, our LinkSaaS platform could be evaluated as a Level-3 SaaS platform for mobile cloud. Moreover, total 4 performance test items are all satisfied with the TTA criteria so that the performance of our platform could be evaluated as Class-A. At the busiest running condition, the CPU utilization rate is measured less than or equal to 8%. Memory usage reveals its maximum amount of less than or equal to 500Mbytes at the worst case. Any kind of data can be through in less than 1 Sec. so that the response time of the platform could also be certificated. Finally our platform is evaluated portable to any kind of mobile device or server because it apparently obey Web standard.

These results indicate that LinkSaaS can be applied to an integrated monitoring system for constructional structure as a mobile cloud platform if we additionally prove the application reveals web accommodations after porting on the platform. Test for application program functions and their web accommodation properties is the 2nd step of this performance analysis.

4.2 Test for Application Programs

Application program that must be ported on the platform discussed in the previous subsection is the integrated monitoring system for constructional structure of maintenance, management, and measurement. In this subsection, we investigate web accommodate performance of this application program ported on the platform described above. Self-test applied software are called 'Web Manager', 'Field Manager', and 'Central Manager'. On the other hand, 'Web accessibility', 'Web Compatibility', and 'Web Openness' are the main categories of test items of our application program.

Tabel 3 shows the results and their corresponding detailed items of the test. Evaluation of web accessibility can be performed by measuring conveniences or easiness of recognition, operation, understanding, and robustness using an automatic diagnostic tool K-WAH4.0. Web compatibility can be evaluated whether the program obeys standard grammar and guarantees or not using the tool KW3C2.0. Finally web openness can be tested using self-diagnostic process on Robots.txt blocking, Noindex/nofollow blocking, Redirecting to install ActiveX, and JavaScript errors.

Table 3 Test Results of Application Program Functions (Web Accommodations)

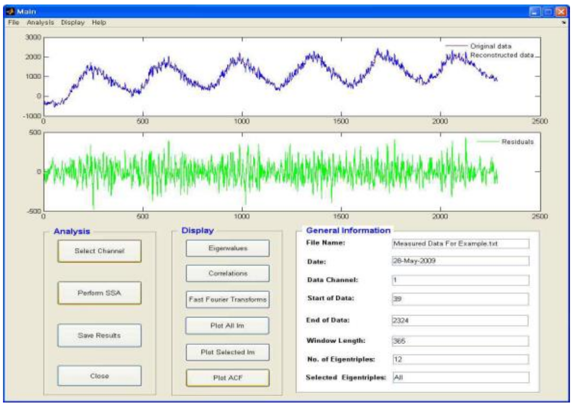
Categories	Detailed Test Items	Results
Web Accessibility	Easiness of Recognition	Pass
	Easiness of Operation	Pass
	Easiness of Understanding	Pass
	Robustness	Pass
Web Compatibility	Observance of Web Standard Grammar	Pass
	Guarantee of Web Compatibility	Pass
Web Openness	Robots.txt Blocking	Pass
	No Index, No Follow Blocking	Pass
	Redirecting to Install ActiveX and Java	Pass
	Java Script Errors	Pass

As shown in the right-hand side of Table 3, all the detailed test items of Web Accommodations for our integrated monitoring system are properly passed.

4.3 Operation Test Using Real Measuring Data

In this subsection, we investigate an essential feature of our monitoring system for maintenance of constructional structure that can be confirmed by real operation of the system applying emulated data. First of all, we obtain emulated data from a simulator fabricated as a self-diagnostic tool. Thus we apply them to our monitoring program for a performance analysis of operation.

Main features of the program tested here are 'Extraction of components from real measuring data', 'Statistical analysis of extracted components', and 'Comparison analysis between emulated data'. Fig. 12 shows a particular result of the monitoring program operation using a bunch of emulated data. Applying the method of spectral analysis, we can extract 'Trend Component', 'Period Component', and 'Irregular Component' etc. using real measuring data obtained from the target constructional structure. Implemented features of the analyzer can be summarized as SSA(singular spectrum analysis) which includes both time-domain and frequency-domain analyses. After selection of channel, we can receive data from the channel and analyze them to extract various kinds of components and perform a statistical processing using these components. We can derive eigenvalues of a specific sensing waveform, correlation functions between waveforms, and spectral functions of waveforms by FFT(fast Fourier transform) algorithm. Finally plotting ACF(autocorrelation function image) and display functions of images(or selected images) in the monitoring program can apparently show these derived values or functions as well as characteristic images.



File Name: Measured Data For Example.txt
Date: 28-May-2009
Data Channel: 1
Start of Data: 39
End of Data: 2324
Window Length: 365
No. of Eigentriples: 12
Selected Eigentriples: All

Fig. 12. Examples of Component Extraction and Statistical Analysis Using Emulated Data

Main screen shows some general information of data being processed with the results of statistical analyses at the same time so that we can easily detect any possible malfunction or uncommon symptom from the comparison analysis between emulated data on the display. We confirm these required features of the monitoring program are well implemented in the system.

5. CONCLUDING REMARKS

We design and implement an integrated monitoring system based on mobile cloud for maintenance, management, and measuring of constructional structures estimated more than 10,000 spread over the national land in Korea.

Throughout this paper, we have tried to overcome conventional problems of ASP or C/S environment which makes them difficult in system tolerance by SaaS and mobile cloud applications. Based on these concepts, we have presented integrated monitoring Hardware(measuring sensor, fixed and dynamic data collectors) and Software(analyzer and monitoring program) so that modification and upgrade can be easily performed by management personnel.

System offers a lot of useful features for the sake of disaster prevention for constructions such as bridge, tunnel, and other traditional structures in folk villages of local heritage. We can obtain rate of variations, cracks, and/or vibrations of target structures and analyze the physical status of them in the long term or short term period as well as disaster situations. Bridge of timber structure in Asan Oeam Village and traditional houses in Andong Hahoe Village are proved to be good applications of the monitoring system presented in this paper. Folk villages, in general, have traditional houses and other structures of timber and they are very close with each other so that there can be a lot of weak points against fire disasters. Our monitoring system can provide with a good solution of automatic detector or watching equipment for possible disasters in that area.

Finally, we present a performance analysis of the monitoring system including TTA authentication test for the platform, web accommodation test for the applications, and operation test using real measuring data. Thus, we prove our system can perform good monitoring functions of maintenance, management, and measuring for constructional structure based on SaaS mobile cloud concept of level-3.

In the near future, we expect a continuous research topic to prevent traditional structures from any kind of disaster using the monitoring system studied in this paper, additionally associated with big data solution.

ACKNOWLEDGMENTS

This work was supported by the National Research Foundation of Korea Grant Funded by the Korean Government (NRF-2014K2A2A4001442)

REFERENCES

[1] Korea MOLIT(Ministry of Land, Infrastructure, and Transport), *Maintenance and Rehabilitation Manual of Management Structure Measuring Integrated Monitoring System*, 2012.

[2] Korea MOLIT(Ministry of Land, Infrastructure, and Transport), *Improvement of Management Structure Measuring Integrated Monitoring System*, 2012.

[3] Yong-Sun Oh, *Maintenance, Management, and Measuring Structure Integrated Monitoring System Based on Mobile Cloud*, the Proposal of cooperation between company and university technology development business, Small and Medium Business Administration, 2014.

[4] H. S. Shin, *Understanding MS Cloud Computing and the Edger Service Platform*, ZDNet Korea, Dec. 2008.

[5] Y. M. Kim, *The Web-Based SaaS Platform*, Dec. 17, 2008, http://www.software.or.kr/ICSfile/afieldfile/2008/12/18/5.pdf

[6] Digital Times, *SaaS and the Future of Software*, May 2006.

[7] S. H. Han, "Study of Thesaurus Updating Using Collective Intelligence," Journal of the Korea Society for Information Management, vol. 26, no. 4, Sep. 2009, pp. 25-43.

[8] http://www.naver.com

[9] S. J. Kwak, *Study of the National Library Operation Plan*, The National Library of Korea, Mar. 2011.

[10] B. H. Yang, "Web 2.0-Based Library Information Service," Information Management Studies, vol. 39, no. 1, 2008, pp. 199-220.

[11] Sasan Adibi, "Data Mining - A Captured Wired Traffic Approach," IJAST, vol. 21, Aug. 2010, pp. 11-30.

[12] P. Intapong, S. Settapat, B. Kaewkamnerdpong, and T. Achalakul, "Modular Web-Based Collaboration Platform," IJAST, vol. 22, Sep. 2010, pp. 37-48.

[13] Dinesh C. S. Bisht and Ashok Jangid, "Discharge Modeling Using Adaptive Neuro - Fuzzy Inference System," IJAST, vol. 31, Jun. 2011, pp. 99-114.

[14] Byung-Won Min and Yong-Sun Oh, "Design of the SaaS Platform-Based Integrated Digital Library System," Proceedings of Korea Contents Association 2010 Spring Conference, 2010, pp. 447-449.

[15] Byung-Won Min, Hyung-Yong Oh, and Yong-Sun Oh, "Improvement of the SaaS-Based Digital Library Integrated Management Service System," Proceedings of Korea Contents Association 2011 Spring International Digital Design Exhibition, 2011, pp. 3-4.

[16] Byung-Won Min and Yong-Sun Oh, "Implementation of Integrated Management System for Digital Library Supporting Multi-tenant Environment Based on SaaS," Journal of Korea Contents Association, vol. 11, no. 5, 2011, pp. 93-103.

[17] Hyung-Yong Oh, Byung-Won Min, and Yong-Sun Oh, "User Customized Web Interface Design Optimized for SaaS-based Digital Library System," Journal of Korea Contents Association, vol. 11, no. 5, 2011, pp. 148-156.

[18] W. K. Lee, H. K. Choi, and K. B. Lee, "A Study on the Transformation of the Oe-am Folk Village -From 1988 to

2001 (after Designated as a Traditional Preservation Area)," Journal of Architectural History, vol. 7, no. 1, 2001, pp. 77-91.

[19] J. S. Lee, W. K. Lee, K. H. Oh, K. Y. Shin, and H. S. Kwon, "A Study on the Basic Ideas for Fire Fighting Prevention System in Traditional Folk Village - Focused on Oeam Village in Asan," Journal of Architectural History, vol. 19, no. 1, 2010, pp. 71-90.

[20] M. S. Kim, "A Study on the Fire Safety of Hahoe Village in Andong," Proceedings of KAAH 2013 Autumn Annual Conference, 2013, pp. 147-154.

[21] Y. S. Jung, "A Study on Locations of Houses and Geographical Features of Hahoe Village," Journal of Architectural Institute of Korea, vol. 26, no. 7, 2010, pp. 135-144.

[22] Sun-Gyu Park, Nobuo Mishima, Hwang-Woo Noh, Jae-Soo Yoo, Sang-Hoon Oh, Byung-Won Min, and Yong-Sun Oh, "Countermeasure against Fire Disaster in Regional Heritage Village on the Concept of ICT-Based Disaster Prevention Design," International Journal of Contents, vol.11, no.1, Mar. 2015, pp. 62-68.

Byung-Won Min
He received M.S degree in computer software from Chungang University, Seoul, Korea in 2005. He worked as a professor in the department of computer engineering, Youngdong University, Youngdong, Chungbuk, Korea, from 2005 to 2008. He received Ph.D. degree in Information and Communication Engineering from Mokwon University, Daejeon, Korea., in 2010. His research interests include SaaS & Mobile Cloud, Database, and Software Engineering. Recently he is interested in Big Data processing and its applications.

Sang-Hoon Oh
He received his B.S. and M.S degrees in Electronics Engineering from Busan National University in 1986 and 1988, respectively. He received his Ph.D. degree in Electrical Engineering from Korea Advanced Institute of Science and Technology in 1999. From 1988 to 1989, he worked for LG semiconductor, Ltd., Korea. From 1990 to 1998, he was a senior research staff in Electronics and Telecommunication Research Institute (ETRI), Korea. From 1999 to 2000, he was with Brain Science Research Center, KAIST. In 2000, he was with Brain Science Institute, RIKEN, Japan, as a research scientist. In 2001, he was an R&D manager of Extell Technology Corporation, Korea. Since 2002, he has been with the Department of Information Communication Engineering, Mokwon University, Daejon, Korea, and is now professor. Also, he was with the Division of Computational Science and Engineering, Georgia Institute of Technology, Atlanta, USA, as a visiting scholar from August 2008 to August 2009. His research interests are machine learning, speech and music signal processing, pattern recognition, and bioinformatics.

Yong-Sun Oh
He received B.S., M.S., and Ph.D. degrees in electronic engineering from Yonsei University, Seoul, Korea, in 1983, 1985, and 1992, respectively. He worked as an R&D engineer at the System Development Division of Samsung Electronics Co. Ltd., Kiheung, Kyungki-Do, Korea, from 1984 to 1986. He joined the Dept. of Information & Communication Engineering, Mokwon University in 1988. During 1998-1999 he served as a visiting professor of Korea Maritime University, Busan, Korea, where he was nominated as a Head of Academic Committee of KIMICS an Institute. He returned to Mokwon University in 1999, and served as a Dean of Central Library and Information Center from 2000 to 2002, as a Director of Corporation of Industrial & Educational Programs from 2003 to 2005, as a Dean of Engineering College and as a Dean of Management Strategic Affairs from 2010 to 2013, respectively. He had been the President of KoCon from 2006 to 2012. During his sabbatical years, he worked as an Invited Researcher at ETRI from 2007 to 2008, and as a Visiting Scholar at KISTI from 2014 to 2015. His research interests include Digital Communication Systems, Information Theory and their applications. Recently he is interested in Multimedia Content and Personalized e-Learning.

Yasuhisa Okazaki
He received the B.S. and the M.S. degrees in Physics from Kyushu University, Fukuoka, Japan in 1988 and 1990, respectively. And also received Ph.D. degrees in Educational Technology from Saga University, Saga, Japan in 2004. Since 2005, he has been an Associate Professor at Graduate School of Science and Engineering, Saga University, Japan. His research interests focus on information and communication systems in learning.

Jae-Soo Yoo
He received his M.S. and Ph.D. degrees in Computer Science from the Korean Advanced Institute of Science and Technology, Korea in 1991 and 1995, respectively. He is now a professor in Information and Communication Engineering, Chungbuk National University, Korea. He has also been the president of Korea Contents Association since 2013. His main research interests include sensor data management, big data, and mobile social networks.

Sun-Gyu Park

He received the Ph.D. degree in the department of architecture from Tokyo University, Japan in September, 2004. He was employed by Mokwon University, Korea as a professor from March, 2009. His main research fields are an initial crack prediction of concrete technology development of the blast furnace slag concrete using alkali activator and a performance enhancement of the recycled aggregate concrete.

Hwang-Woo Noh

He received the B.S. and M.S. degrees from Department of Industrial Design, Hanbat National University, Daejeon, Korea, in 1996 and 2003 respectively. He has also completed his Ph.D. degree course at the Department of Industrial Design, Chungnam National University, Daejeon, Korea, in 2013. From 1997 to 2008, he worked as a representative of Design-Comphics Inc. He joined the faculty of Department of Visual Design, Hanbat National University, Daejeon, Korea, in 2009. During 2008-2009 he served as an executive director of Korea Design Industrial Association where he was nominated Head of the Daejeon-Chungcheong Branch. He is currently a professor of Hanbat National University, a Secretary-General of Daejeon Design Development Forum, and a Vice-president of Korea Contents Association. His main research interests include Visual Communication Design and its Fundamentals, Packaging Design, and Disaster Prevention Design.