Framework Based on Web Service for Efficient Management of Athletes in the Ubiquitous Environment

*Kyung-Sik Kim, **Lubin Lee, *Jin-Sung Kim
*Dept. of Information and Computer Science Dankook University,
**Lubin Lee (Western Reserve Academy, USA)
kyungsik.kim@gmail.com, leeofunited@daum.net, ckjs001@gmail.com

Abstract

In this paper, an efficient framework based on Web Services has been proposed. The framework has various functions, which are needed in Web service environment. The necessary functions of Web service architecture are analyzed and the frameworks using CBD(Component Based Development) method is implemented. Efficient techniques are applied to manage athlete information. For the experiment, athlete management system is implemented and client programs that work with athlete management system are also implemented in various platforms (Windows, Solaris, Linux) using different languages. As a result of experiment, the proposed framework saves costs by decreasing production time and is more efficient to manage since core function are made of component.

Keywords: Web Services, Ubiquitous Computing, CBD, Athlete Management System

1. Introduction

Web Services is emerging to provide integration and interoperability in various computing environment. It is provides independent environment based on standard interfaces regardless of hardware, platform and program languages. Now, many research that is related to Web Services have been processing like XML[1],SOAP[2],WSDL[3],UDDI[4]. The AXIS 2[5] and Web Services toolkit 2.01[6] are generally used to construct Web Services.

However, most toolkits of Web Services have basic functions, so we spend much time and costs to develop application, which are supporting Web Services functions. The system based on Web Services is very complexity because many processing logics are necessary in order to integrate with legacy system. Consequently, it is hard to manage and maintain of the system as well [7.8].

In this paper, we propose a framework that supports Web Services. The framework is designed based on necessary elements to supports Web Services efficiently in the ubiquitous environment and implemented by CBD(Component Based Development) method. We implement components of the framework having interface to be used respectively. The Athlete Management System is implemented with the framework to prove efficiency of the proposed methods in the framework. We construct experiment environment by using the Athlete Management System and clients program and experiment with them.

The remainder of this work is structured as follows: the next section designs and implements a framework based on Web Services with CBD methods. Section 3 constructs athlete management system by using the proposed framework. Section 4 constructs experiment environment and performs experiments with athlete management system and client programs. The experiments also are performed with them. Finally, section 5 summarizes the study with some conclusion and points to some future works.

2. Design and Implementation of Framework

In this section, we propose a framework and then design and implement the framework applying the UML(Unified Modeling language) method.

2.1. Propose Framework

We propose a framework as figure 1. The framework is comprised of 11 parts to be satisfied with Web Services features in the Ubiquitous environment. We consider various conditions such as database connection, security, errors and events, resource access and frame-

work management. The detail functions of each element in the framework will be explained section 2.2.

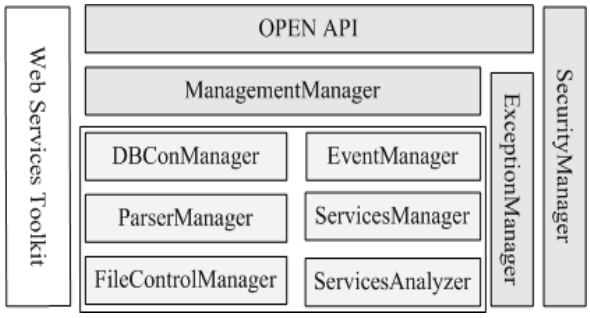

Figure 1. Proposed framework

2.2. Design Framework

We design the framework with the UML(Unified Modeling Language)[9] that is one of CBD(Component Based Development) methods. In this paper, we study about component diagram and batch diagram of the framework.

1) Component Diagram

We define 10 components based on necessary elements, which are defined section 2.1. Not only Components can be used directly access of each components through interface, but also COPENAPI. COPENAPI has interface to access the components, so we can use all components freely. CDBConManager is used for connecting to the database. CParserManager parses client's requests, which generally is consisted of client information and services. CFileControlManager handles the Files such as setting files and template files. CExceptionManger controls errors that are occurred in the framework. CSecurityManager protects exchanging information between clients and system, which is built by the framework. CServiceAnalyzer analyzes client's services requests. CEventManager handles events, which are happened in the framework. CServicesManager manages service lists that might be provided by implemented system with the framework. CManagementManager manages the framework.

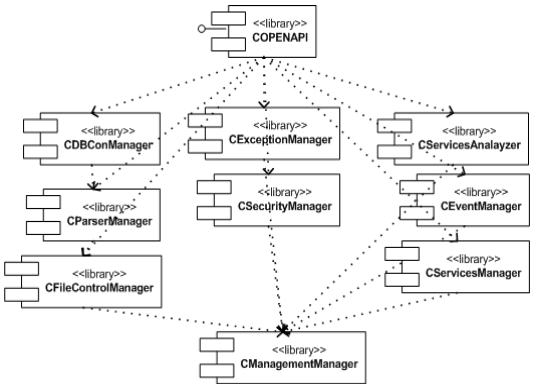

Figure 2. Component Diagram of the Framework

2) Batch Diagram

The batch diagram of the framework is comprised of Web Service Provider System, Web Services Client and Database Server. The Web Services Provider System has Internet Information Server 6.0, Web Services Framework and Microsoft .NET Framework. The Web Services Framework is the proposed in this paper. The Database Server has raw data, which are needed to construct a system.

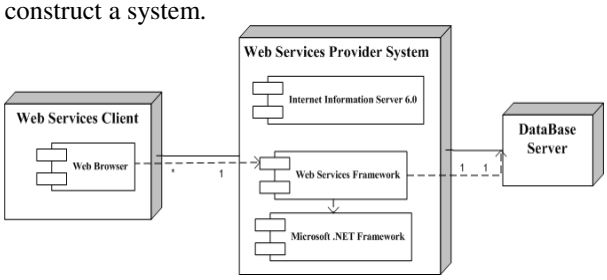

Figure 3. Batch diagram of the framework

2.3. Algorithms

In this section, we study the core algorithms applied in the framework. The Framework has 4 algorithms such as SERVICE_PROOCESSING algorithm, CLIE NT_AUTHENTICATION algorithm, RESOUR CE_ ACCESS algorithm and SERVICES_RUNNING _CONDITIION algorithm, but in this paper, we just examine into SERVICE_PROOCESSING algorithm and CLIENT_AUTHENTICATION algorithm.

1) SERVICE_PROCESSING Algorithm

This algorithm is used to process services in the ServiceManager Component of the framework. It is working when clients request the services. The clients request for services with arguments such as client information and services information.

Algorithm 1 SERVICE_PROCESSING
Input : client Information (type :xml), Services (type:xml)
Ouput : requested service(type:xml)

BEGIN
1. **IF** verifyClient(clientInfo) **THEN**
2. **IF** existServices(requestServices) **AND** verifyAccessResource(requestServices) **THEH**
3. **RETURN** ServiceConnection(requestServices)
4. **ELSE**
5. throw Exception()
6. **END**
7. **ELSE**
8. throw Exception()
9. **END**

Figure 4. SEVICE_PROCESSING algorithm

In line 1, a client authentication is executed with client information. If the authentication is success, move to next line or else move to line 8(when errors are occurred). In line 2, running services in the framework is checked with requested services of the client and then access authentication is checked. If both conditions are satisfied, requested service is processing and provides it to the client or else the framework creates error message and transmits it to client.

2) CLIENT_AUTHENTICATION Algorithm

This algorithm is used in order to verify clients. In line 1, sqlquery is created by using client information. In line 2, client authentication is executed with client information on the database. If client is validate user, return true or else return false.

Algorithm 2 CLIENT_AUTHENTICATION
Input : client Information (type:xml)
Ouput : authentication result(type: boolean)

BEGIN
1. sqlquery=create query with client info
2. **IF** dbcon.executequery(sqlquery) **THEN**
3. **RETURN** true
4. **ELSE**
5. **RETURN** false
6. **END**

END

Figure 5. CLIENT_AUTHENTICATION algorithm

3. Athlete Management System

We implement the Athlete Management System with the proposed framework to prove efficiency of the proposed framework. The Athlete Management System is implemented Web-based by using C#. The Athlete Management System, which is implemented in this paper, now is used to manage our university's athletes.

3.1. Database schema

A database for storing athlete information is designed as shown figure 6. The database schema is comprised of 11 tables. The Athlete_Info table, which is core parts of database, controls all information about athletes. An athletic club consists of schedule information, manager information, training information and athlete information. Each athlete has personal information, game information, family information, training information, department information, consult information and graduation information.

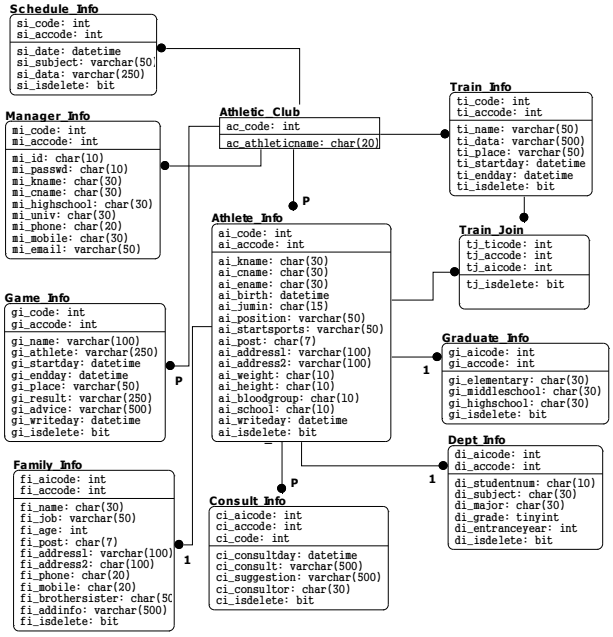

Figure 6. ERD(Entity Relationship Diagram) of athlete management system

3.1. Service List

The Athlete Management System provides all functions through Web Services. Table 1 shows services lists in the Athlete Management System. To access services, clients must be authenticated and have an access level. If access level is 9, clients can access limited services. Increasing value of access level, security is reinforced.

Table 1. Web service list in the athlete management system

	Service Name	Access Level

Athletic Club Information		Athletic_ClubInfo	1
Game Schedule Information		Game_ScheduleInfo	1
Game Information		GameInfo	1
Training Information		TrainingInfo	2
Athlete Related Information	**Athlete Information**	AthleteInfo	1
	Game-grade Information	Athlete_GameInfo	5
	Department Information	Athlete_DeptInfo	5
	Family Information	Athlete_FamilyInfo	9
	Consulting Information	Athlete_ConsultingInfo	9
	Physical Information	Athlete_PhysicalInfo	5

3.3. Athlete Management System

We implement the Athlete Management System web based by using the proposed framework. The Athletic Management System is divided into web and Web Services. The web provides management functions through the web interface as shown figure 7. The Services also provides same function as web but it provides Web Services. Figure 8 shows Web Services lists which is used to access athlete information.

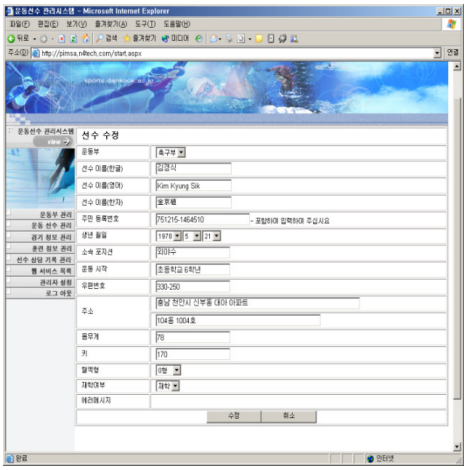

Figure 7. Athlete management system

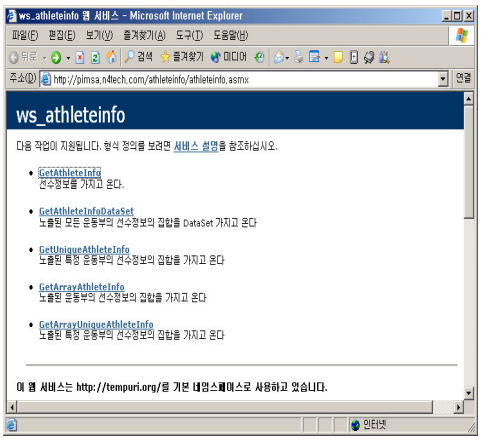

Figure 8. Athlete information in the web service

4. Experiment

In this section, we perform an experiment to test operating together between the athlete management system and client programs. To do this, we construct experiment environment with servers and clients.

4.1. Construct Experiment Environment

Figure 9 shows experiment environment. It is comprised of Database Server, Athlete Management Server, Proxy Server, and three Clients. The Database Server has raw data of athlete information. Athlete Management Server manages and process athlete information. Three clients request and use athlete information from Athlete Management Server. Table 3 shows hardware and software configuration of client programs for the experiment.

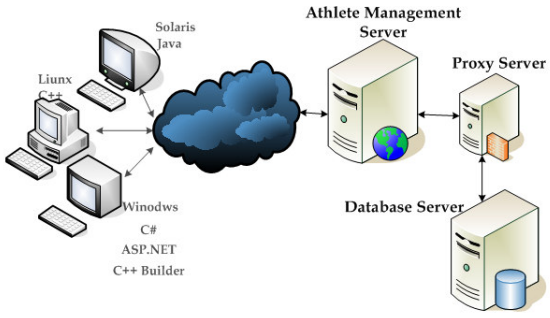

Figure 9. Experiment environment

Table 2. Hardware and software conflgurati-on of client programs for the experiment

	Hardware	Software
Client 1	CPU: Intel P4 2.4GHz RAM: 2G HDD:300GB NIC: 100Mbps	Solaris 10, Java, AXIS

Client 2	CPU: Intel P4 2.0GHz RAM: 1G HDD:200GB NIC: 100Mbps	Windows 2003 Server, IIS 6.0, C#, SOAP tool-kit3
Client 3	CPU: Intel P4 1.8GHz RAM: 2G HDD:200GB NIC: 100Mbps	Linux 2.6, PHP

4.2. Experiments with Client Programs

We implement the client programs, which are differently implemented according to programs languages (Java, C#, PHP) in various platforms (Windows, Linux, Solaris). We perform the experiment about the interoperability among the Athlete Management System and the client programs. The client programs are comprised of basic functions in order to perform the experiment.

1) Solaris Platform with java

The Figure 10 shows client program, which is implemented with Java in the Solaris. We create and use WSDL stub code with JDK 1.4 SDK and AXIS of apache.

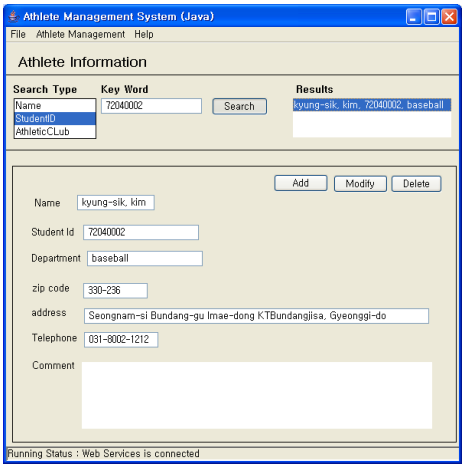

Figure 10. Client program that is implemented by Java

2) Windows 2003 server Platform with c#

The Figure 11 shows client program, which is implemented with C# in the windows 2003. We use .NET framework and Web Services Toolkit 2.01.

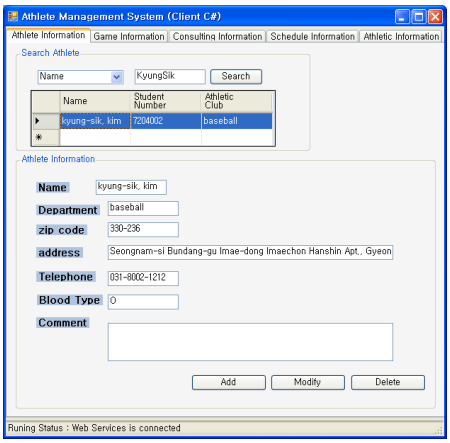

Figure 11. Client program that is implemented by C#

3) Linux Platform with PHP

We implement the client programs with PHP:SOAP in the Linux. We could use most functions of the Athlete Management System. Figure 12 shows all athletic club information.

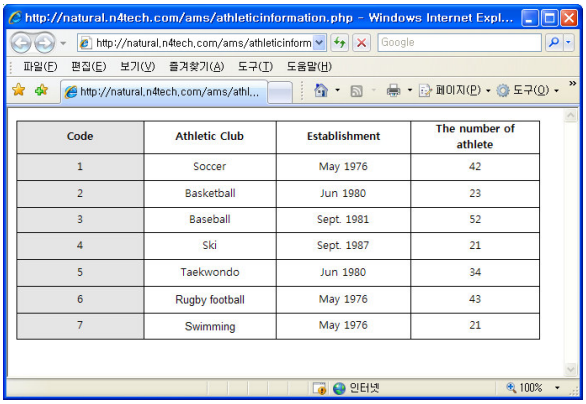

Figure 12. Client program that is implemented by PHP

4.3 Experiment Result

The experiment result, we could get efficiencies as follows. First, we could easily implement the athlete management system with the core functions of the framework. Consequently, we could production time and costs. Second, the framework provides independent environment regardless of program languages and platforms. Finally, core parts of the framework is implemented applying CBD(Component Based Development) methods, so it is impossible to modify and update according to components. Therefore, it is efficient to manage and maintain the framework.

5. Conclusion and Future Works

In this paper, we propose the framework of athlete management system based on the Web Services in order to use in ubiquitous environment. The framework has various functions which are needed for Web Services environment. The components of the framework are implemented by CBD methods. The athlete management system is implemented with the proposed framework. The client programs also are implemented by different language(C#, Java, C++, PHP) to test interoperability in various platforms. Efficient techniques are applied to manage athlete information in the system. As a result of experiment, we could decrease costs and time with the proposed framework during system constructing and the constructed system provides independent environment regardless of programs and platforms. In the future, we will add athlete recommendation function, which recommends each athlete according to game situation. We also will extend some functions of the framework to be adapted in various fields.

ACKNOWLEDGEMENT
The present research was conducted by the research fund of KOCCA as the result of 2008 culture technology development.

References

[1] Extensible Markup Language, http://www.w3 .org/XML, 2008
[2] SOAP Specification, http://www.w3.org/TR/soap/
[3] Web Services Description Language, http://www.w3.org/TR/wsdl, 2008
[4] Universal Description, Discovery and Integration, http://uddi.xml.org/uddi-org, 2008
[5] Axis2 project, http://ws.apache.org/axis2, 2008
[6] Microsoft Web Services Toolkit 2.01, http://www.microsoft.com, 2008
[7] Christina Catley, Dorina C. Petriu,Monique Frize, "Software Performance Engineering of a Web service-based Clinical Decision Support infrastructure," Proceedings of the fourth international workshop on Software and performance, Redwood Shores. California, pp. 130-138, 2004
[8]K. Hogg, P. Chilcott, M. Nolan,B. Srinivasan, “An evaluation of Web services in the design of a B2B application,” Proceedings of the 27th conference on Australasian computer science, Dunedin. New Zealand, pp. 331 - 340, 2004
[9] Unified Modeling Language, http://www.uml.org, 2008