Constructing a SOAP-based Distributed Mobile First-Aid System

Hsing Mei, Chih-Chien Chang, Shou-Jin Chou and Chia-Bin Tsai
Web Computing Laboratory
Computer Science and Information Engineering Department
Fu Jen Catholic University
Taiwan, R.O.C.
http://weco.csie.fju.edu.tw

Abstract

In the first aid operation, time plays the critical role. We designed and implemented a distributed mobile first-aid system based on SOAP. We hope the system could save the time throughout the first-aid procedures. With the help from this system, first-aid personnel could get more useful information to rescue people. In this paper, we introduced the motivation of our mobile first-aid system; background of Web Service; and the system itself. The system consists of three parts: Hospital Server, Core Server and Client Devices. We implemented Hospital Server and Core Server based on Java programming language, and two types of Client Devices (Win CE and Palm OS). Since this system is constructing using SOAP, it inherits the benefits of Web Services.

1. Introduction

Web Service starts a new age of Internet, it is a web-based application provides not only contents but also interactive services to meet the user needs, and most importantly, it has highly flexibility and scalability [1]. Simple Object Access Protocol (SOAP) is used as the packaging protocol of Web Services. Web Service is built using standard Internet technologies and combines with HTTP and XML protocols. We designed and implemented a mobile first-aid system based on SOAP thus the system has highly flexibility and scalability.

In the operations of first aid, time plays the critical role. Time could be saved significantly with the new technologies. Here we use mobile device and Web Services to save time. We assume that all first-aid personnel equip with mobile devices (PDAs with communication capability). When an accident occurred, the first-aid personnel can receive the instruction from the device and go to rescue the patient immediately. First-aid personnel can go to the scene right away, and get all the medical records about the patient on his way to the scene. When the first-aid personnel arrive in the scene, he/she could use this information to deal with the patient without any delay.

The distributed mobile first-aid system consists of a network of hospitals, and each of them has a legacy system. It is difficult to integrate all hospitals to form the distributed system by using traditional technologies, but it is possible to deal with this problem by Web Services. Web Services is built using the standard Internet technology, so it can cross different platform easily and requires less modification for the individual legacy systems.

The remained section of this paper is organized as follows. Section 2 provides the background of Web Services. Section 3 presents the design and implementation of our mobile first-aid system. Finally, we discuss the conclusion and the future work.

2. Background

In this section, we introduce Web Services, Simple Object Access Protocol (SOAP), and the benefits of Web Services. SOAP is used as the packaging protocol of Web Services. We describe what Web Services and SOAP are, show the SOAP message format and take the SOAP message used in our system as an example. The benefits of Web Services are illustrated.

Web Services Overview

Web Service is a new technology built on standard Internet technologies. It provides a network accessible interface between the application and the user. Through the interface, a Web service can be composed of other Web services. Therefore, various Web services can form a web of services. Fig. 1 shows the basic concept of Web Services.

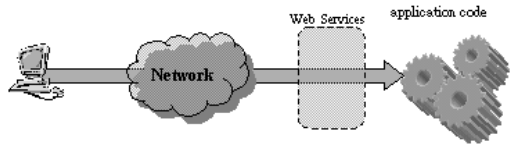


Fig. 1: A Web service allows access to application code using standard Internet technologies.

Simple Object Access Protocol

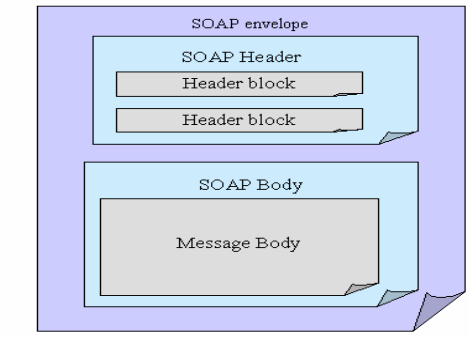
Simple Object Access Protocol (SOAP) is a lightweight protocol for exchanging information in a decentralized, distributed environment. SOAP consists of three parts. The first part is the SOAP message format. The second part is a set of encoding rules for expressing instances of application-defined data types. The last part is a convention for representing remote procedure calls and responses. (All of what we described here is based on the SOAP 1.1 Specification [2].)

A SOAP message is based on XML format and is called a SOAP *envelope*, which contains an optional SOAP *header* and a required SOAP *body*, as shown in Fig. 2(a). The header usually consists of the information of SOAP context, SOAP data type, and SOAP encoding-style. SOAP body contains the actual information to be exchanged. Anything that can be expressed in the XML format can be contained in the body.

In the SOAP specification, the encoding style is defined as "based on a simple type system that is a generalization of the common features found in type systems in programming languages, databases and semi-structured data [2]." Briefly, the encoding style is meant to be the data type or the data model of the information contained in the SOAP envelope.

The data type can be a simple data type or a compound data type. You can describe your own compound data type in XML format. Fig. 2(b) shows the SOAP message of our system used to let the mobile client to send the emergency call to the hospital server, and the hospital server can get the information of the client's identity and location. It consists of a SOAP envelope and a SOAP body, and the SOAP body contains the information of identity and location. As specified in the encodingStyle attribute of the SOAP envelope, we use the encoding rules recommended in the SOAP specification. We also defined our own compound data type as specified in the "http://csie.fju.edu.tw" namespace.

The specification does not constrain the SOAP message exchange model. That is, a SOAP message is fundamentally a one-way transmission from a sender to a receiver. But SOAP messages are often implemented as request/response model. Also, the specification does not constrain the transport protocol to exchange messages. But most SOAP implementations exchange messages over HTTP today. Because of the pervasiveness on the Internet,



(a) the structure of SOAP message

```
<S:Envelope
   S:encodingStyle='http://schemas.xmlsoap.org/soap/encoding/'
   xmlns:S='http://schemas.xmlsoap.org/soap/envelope'
   xmlns:S='http://schemas.xmlsoap.org/soap/encoding/'
   xmlns:S='http://csie.fju.edu.tw'
   xmlns:S='http://www.w3.org/2001/XMLSchema-instance'
   xmlns:S='http://www.w3.org/2001/XMLSchema'>
   <S:Body>
        <a:EmergencyCall>
                <PID b:type='c:string'>110110</PID>
                <Location b:type='c:string'>123 123</Location>
        </a:EmergencyCall>
   </S:Body>
</S:Envelope>
```

(b) the emergency call SOAP message

Fig. 2: The SOAP message

SOAP over HTTP can take the advantage of that most firewalls always pass the traffic of SOAP message straight through.

The Benefit of Web Services

We use Web Services to facilitate our system to achieve certain functions, for example, the Mobile Client sends an emergency call to the Hospital Server, and then the Hospital Server can get the ID and location of the Client. The benefits using Web Services [15] are as follows:

- **Using standard protocols:** The various services can be easily implemented by following the universally agreed specification (the standard Internet protocols). Programmers need not to care about the details for low-level functionalities.
- **Ease of integration with traditional web application:** Web Services can be easily integrated with the legacy systems (traditional web application). Our system takes advantage of Web Services, so a hospital can join easily and only requires least modification.
- **Interoperability:** Developers using the Web Services do not have to care about which programming language or operating system the services are hosted on. Inside our designed system, the hospital can use different operating system, UNIX, Microsoft Windows, Linux, etc.
- **Software component as a Service:** Web Services allows some components of the services to be explored for others to build bigger services, instead of reinventing the existing services.
- **Efficiencies and Extensibility:** A new Web service can be composed of other existing Web services, so incremental development using Web Services is very easy and efficient. Instead of reinventing existing services, the developers can concentrate on critical tasks, and the business can focus the value added service.

Here, we illustrated the benefit of Web Services. In fact, there are some interoperability problems between different SOAP implementation and other issues are discussed more in [5], [6] and [7].

3. The Distributed Mobile First Aid System

Our system is divided into three parts – Hospital Server, Core Server, and Client Devices. Each Hospital Server is appended to the legacy hospital system, which is responsible for communicating with the Core Server or clients.

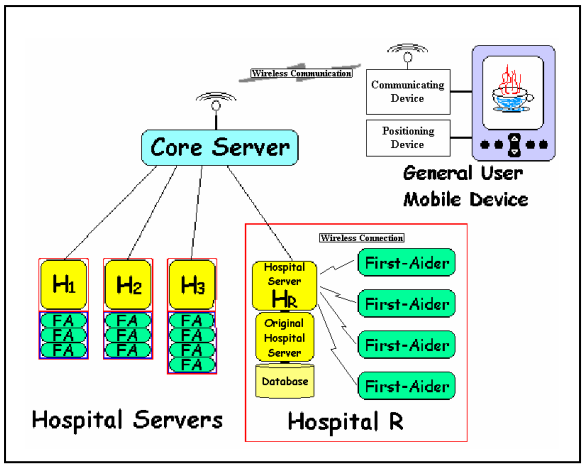


Fig. 3: System Structure

Core Server is designed and implemented with Web Services. There is a trend to use Electronic Medical Record (EMR) [9] in XML format. The record collection could be implemented easily and should be realized as quickly as possible. If SOAP is used for XML documents delivering, then the developing time will be shortened. Note that a redesigned and simpler EMR is used in our implementation.

Client Devices, or clients for short, are mobile devices such as cell phones or PDAs. There are two types of clients, *General User device* (GU device) and *First*-Aid *Personnel device* (FA device), which serves general users and first-aid personnel respectively. The way to communicate between mobile clients and servers is mainly by SOAP. Due to the complexity of parsing XML documents for mobile devices, HTTP is used.

3.1 System components

Hospital Server

Hospital Servers support many functions. The primary functions are listed as follows:

- Hospital Server will inform Core Server when a new user registering to this hospital.
- In an emergency, Hospital Server would receive user's call from Core Server, request integrated EMR and notify the first-aid personnel. Essential information, such as location, name, sex, blood type, birthday, age, EMR will be sent to first-aid personnel.
- Hospital Server will assist Core Server to collect the users' EMR when needed.
- Hospital Servers are the intermediaries between Core Server and origin servers of hospital. Hospital Server might be seen as a "Web Services Component" of origin server. They get patients' data and EMR from origin server and use Web Services technology to connect to Core Server.

Core Server

Core Server is the center of the whole system. We design Core Server as a centralized environment, so emergency calls could be dealt with collectively. Primary functions of Core Server are listed below:

- Hospitals need to register to Core Server because that Core Server needs to know their services URL for transmitting data and location. Also Core Server could prevent unauthorized data gathering in some aspects.
- After an emergency call was received, Core Server should know caller's data including ID and location. According to these data, Core Server would make an emergency request to the hospital nearby the caller and begin to collect this user's EMR and status.
- By registration data, Core Server should know which hospitals the user has registered. Core Server will collect EMR from these hospitals and send the integrated EMR to requesting hospital. Note that the user's EMR may not be necessarily available.
- Record the time when the event happened for further use.

Client Devices

GU Devices and FA Devices are designed for users and first-aid personnel respectively.

The primary functions of GU device are:

- Users can make an emergency call from this device. The device will send the user's ID and location to Core Server.
- User can send his/her condition to specific hospital so the doctors of the hospital can monitor this user.
- The GU device can register on line.

The primary functions of FA device are:

- First-aid personnel are assigned to the mission from the Hospital Server. After receiving the mission, he/she can get the EMR and the condition of the user.
- After the emergency action is done, the first-aid personnel can report the condition of the user. With this information, doctors in the hospital can make necessary preparation before the patient arrives.
- When the emergency mission is finished, the first-aid personnel notifies the Hospital Server that the mission is completed, and then the information will be passed to Core Server and be stored in database.

3.2 System Operations

Registration

A Hospital Server needs to register to Core Server before using the system because Core Server needs to know where the hospital is. A Hospital Server should inform Core Server of users' registration because Core Server needs to know what EMR the hospital has. Registration mechanism is shown in Fig. 4. Hospital Servers register to Core Server, users register to Hospital Servers and users' information should be passed to Core Server.

Emergency Call

Suppose that General User Device and First-Aid Personnel Device both have communicating and positioning equipments. In emergency, a user makes an emergency call to Core Server by using his mobile device. Core Server would accept his call, inform the nearby Hospital Server, and begin to collect the user's EMR. After the emergency call has been passed to the hospital, the Hospital Server would assign first-aid personnel a rescuing mission, and then sends the user's EMR to first-aid personnel and doctors for proper treatment.

There are several ways for positioning the location of patient, such as using Global Positioning System (GPS) or Global System for Mobile Communications (GSM).

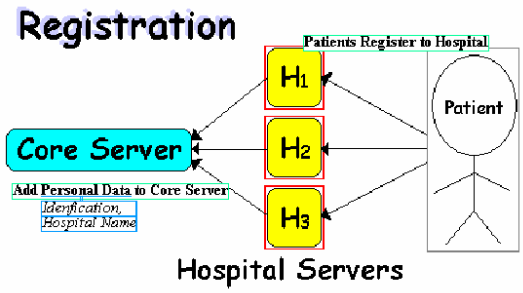


Fig. 4: Registration.

Security Issues

Although security is an important requirement in distributed system, we did not integrate secure design in current implementation. For example, users' EMR should not be transported over Internet in plaintext, Core Server and Hospital Server should authenticate each other to prevent illegal data access, and user location would not be revealed to everyone. Security issues could be solved after the system is fully evaluated.

4. System Implementation

We implemented the system with Java programming language on server side, and two

types of clients, Win CE and Palm OS. On Win CE [10] and Palm OS [11], Microsoft eMbedded Visual Tools [12] with pocketSOAP [13] and J2ME [14] with kSOAP are used as developing tools respectively.

Hospital Implementation

Because large amount of data (EMR) needs to be transferred, Hospital Servers choose SOAP to deliver this kind of data.

For user registration, Hospital Server provides functions for users. The functions inform user's registration to Core Server by sending user "ID" and "Hospital Name".

In emergency, Hospital Server should alert FA by sending a message containing the caller's ID and location. Hospital Server sends another message contains necessary information to FA when available. This message includes EMR and personal data.

In Information Collecting phase of Core Server, the Hospital Server remote procedures enable Core Server to collect EMR from or send necessary information to this server.

Core Server Implementation

Same as Hospital Server, all large files (EMR) should be transferred by SOAP. There are two primary services – Registration service and Information Collecting service. Two services almost operate independently. A service could be added, modified, or removed easily. The primary concerns are management and security.

Registration: This service provides functions for hospitals and users registration. Any hospital can join or leave the system at any time. Any mobile, new, or old devices could use SOAP to register on the system.

There are two types of registration data being stored in this server. User registration data is constructed during users registering to hospital; hospital registration data is constructed while the hospital is joining the system.

Information Collecting: This service provides functions to collecting EMR, receiving emergency call, and event recording. When activated, Core Server begins to request the user's EMR from the hospitals holding the EMR and then integrates them in one EMR. At last the integrated EMR would be transferred to the hospital nearby the caller.

Clients Implementation

Clients communicate with servers via HTTP and SOAP. To save parsing time of XML document when small amount of data needs to be transfer, HTTP is used. Otherwise SOAP is chosen for delivering. For example, GU makes an emergency call by sending ID and location via HTTP, and FA gets user's EMR from Hospital Server via SOAP.

GU software is an application on mobile device. If combining with mobile device hardware, GU device would be more useful to user. Some mobile devices, such as Hewlett Packard Jornada 548 running with Microsoft Win CE, can personalize hardware buttons. This significant design accelerates the time to launch the application, and also the time to make an emergency call.

If a user is seriously ill, this user can send his condition to his doctor every day. Conditions could be recorded by condition monitoring device and would be sent via SOAP. Condition monitoring devices might be some like the devices in Digital Angle [8].

There are three steps for online register. First, GU gets the XML document from the Core Server, which contains each Hospital Server's name and service URL. Then, after selection of proper hospital name, GU would get the second XML document from the Hospital Server. The document contains the doctor's service and time. Finally, the user makes an appointment with hospital by sending register data.

FA devices must login Hospital Server before carrying out their missions, because not everyone has the right to read user's EMR. Most FA functions are using SOAP. FA receives mission via SOAP because some basic information – name, location, and status would be sent. EMR collecting is using SOAP. After emergency treatment, FA reports user's status via SOAP.

5. Conclusion and Future work

In this paper, we introduce the motivation, background, design and implementation of our distributed mobile first aid system. We believe that this system is really useful for first-aid procedure. Although we have not implemented all components of our distributed mobile first aid system with Web Services. By using SOAP, our system inherits the benefits of Web Services and has highly flexibility and scalability. Until now, we have implemented the designed system with two types of clients with different OS. It is possible to develop various clients easily based on SOAP.

By enabling communication between mobile device and tradition medical equipments, we could get more applications to enhance the medical service and the first aid operation. In the future, we would try to combine various medical equipments and other new technologies to enhance the capability of the distributed mobile first-aid system.

Universal Description, Discovery and Integration (UDDI) and Web Service Definition Language (WSDL) are also the main components of Web Services. Our next step is to integrate UDDI and WSDL into this system. It would make our system more comprehensive and more close to the practical environment.

6. Reference

[1] James Snell, Doug Tidwell, Pavel Kulchenko, "Programming Web Services with SOAP," O'Reilly, First Edition December 2001

[2] Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Frystyk Nielsen, Satish Thatte, Dave Winer, *Simple Object Access Protocol (SOAP) 1.1,* W3C Note 08 May 2000 http://www.w3.org/TR/SOAP/

[3] Anne Thomas Manes, "Enabling Open, Interoperable, and Smart Web Services - The Need for Shared Context," position paper at W3C workshop on Web Services, 12 March 2001. http://www.w3.org/2001/03/WSWS-popa/paper29

[4] Apache SOAP v2.2 documentation. http://xml.apache.org/soap/docs/index.html

[5] James Snell, "Web Services Interoperability," January 30, 2002. http://www.xml.com/pub/a/2002/01/30/soap.html

[6] "Interoperability with Other SOAP Implementations." http://xml.apache.org/soap/docs/index.html

[7] Keith Ballinger, "Web Services Interoperability and SOAP," Microsoft Corporation, May 1, 2001. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsoap/html/soapinteropbkgnd.asp

[8] "Digital Angel" http://www.digitalangel.net/

[9] Health Level 7: http://www.hl7.org/

[10] Microsoft Windows CE: http://www.microsoft.com/windowsce/

[11] Palm OS: http://www.palm.com/

[12] Microsoft eMbedded Visual Tools. http://www.microsoft.com/mobile/downloads/emvt30.asp/

[13] PocketSOAP.com http://www.pocketsoap.com/

[14] Java 2 Platform, Microsoft Edition. http://java.sun.com/j2me/

[15] WebServices.Org, "Why Web Services," 16 February, 2002. http://www.webservices.org/index.php/article/articlestatic/75