Healthcare Web Services for Rural People

[1]V Dinesh, [2] J Jeyasudha, [3]G Senthil Kumar
[1]MTech Student, [2]Assistant Professor, [3Assistant] Professor
Dept of Software Engineering, SRM University, Kattankulathur, Chennai, India-603203
[1]dinesh_v19@yahoo.com , [2]jjeyasudha@gmail.com, [3]senthilkumar.g@ktr.srmuniv.ac.in

***Abstract*- The use of information technology and management systems for the betterment of health care is more and more important. However, current efforts mainly focus on informatization of hospitals or medical institutions within the organizations, and very few are directly oriented to the patients, their families, and other ordinary people. The crude demand for various medical and public health care services from customer's calls for the creation of powerful individual-oriented personalized health care service systems. In this paper, we present and implement a software modelled healthcare Information Service Platform, which is based on such technologies. It can support numerous health care tasks and provide individuals with many intelligent and personalized services, and support basic remote healthcare. In order to realize the personalized customization and active recommendation of intelligent services, techniques such as decision making are integrated.**

Keywords ——Health care service systems, Access oriented services, Web services.

1. INTRODUCTION

As an emerging form of enabling technology, Web-Services based Applications provide patients easier accesses to their healthcare information and services. Service-Oriented Architecture provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations. In This paper, we implement such a Public-Oriented Personalized Health Care which can integrate many backend medical services effectively. A major challenge in implementing such a system is to meet critical security requirements, such as the confidentiality of patient data, the integrity of diagnosis results, and the availability of healthcare services additionally also the prospect of offline services. This Thesis addresses the issue from the implementation of the software perspective.

Users of a public-oriented personalized health care can create and save a personalized page including only the content they would like to access [3]. For example, a nurse or a doctor may prefer seeing only the newsfeed in cardiology or a relative of the patient would be able to request generation of reports from the GUI. A medical service (Web Service) can run in real time, automation or store-and forward mode, and the web App should support all of above. Further, there is a strong demand for medical and health care service systems for the public under the new computing model. They can provide remote health consultation, remote real-time monitoring, remote diagnosis, personal health record network-based health care education, and other personalized services for the public through personalized medical information management and services configuration and integration.

The essential idea of enabling web services is to move beyond HTML and to support arbitrary program to program interaction across the internet. Web services allow a broad range of internet applications to be constructed from web components which offer web service interfaces. In this paper, we are able to adapt web services to overcome patient data relay problems by activating the alerts based on a decision making library [13] pre-defined with SOAP and encryption protocols, and thereby achieve business integration across the web. The adaptation is achieved using a platform independent architecture for decision making library, database management and the web services deployed and written within the web application... An XML script is used to drive multiple XSL (extensible style sheet language)[HREF1] transformations of web services; this enables the interoperability of different web services thereby making the feature of adding more services available to the healthcare or in this case primarily a ICU department of a hospital.

2. RELATED WORK

Creating a public-oriented health care system has triggered considerable research interest in both academia and industry in recent years, and the rapid advancement of information and communication technologies brings more opportunities for innovations in the health care field. Some systems are now available to provide various solutions, and there are also some research projects that aim at providing Web-based personalized health care support. Thus, in this section, we provide an overview of some of these systems and projects, and related service composition work.

A. WEB SERVICES

Web services provide a standard means of interoperating between different software applications running on a variety of platforms and/or frameworks [10]. Web services are applications that expose their business logic, data and processes through programmatic interface. Unlike traditional Client/Server models, Web services generally use HTTP as the underlying communication protocol which allows messages to go through most of the firewalls (most of the firewalls' setting allow the access to HTTP port) [3]. Web services do not provide users with GUI (Graphic User Interface). However, developers can add Web services into their Web page or applications and other users with GUI which contains the functionalities of the Web services. The major advantages of implementing an SOA using Web services are that Web services are very simple and programming language and platform-independent. Moreover, there are many extended Web services specifications ranging from transactions,

business process, messaging, transport, security, metadata to performance, which are on their way toward standardization. These provide enterprise-level integration in a standard way for services to interoperate with each other without any incompatibility and at the same time ensures the QoS and other requirements. As the three core specifications of the Web services protocol stack, SOAP, WSDL and UDDI form the initial specifications for Web services. Figure 1 describes the architecture of Web services. In the following, each of them will be introduced in details.

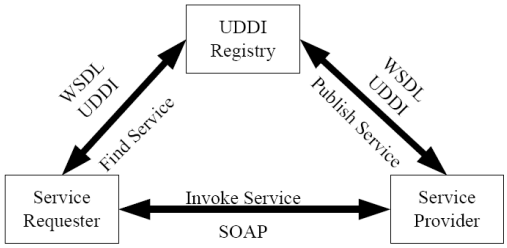

Figure 1: Web Services Architecture

B. SOAP

SOAP is a protocol for exchanging XML-based messages between peers over computer networks. It provides a standard way to access Web services. SOAP is the foundation of the Web services protocol stack and provides the basis upon which other abstract layers are built. It generally uses the HTTP/HTTPS (other protocols like SMTP and XMPP can also be used) as its underlying protocol. The major advantage of using SOAP over other distributed protocols like GIOP/IIOP or DCOM is that SOAP with HTTP works well with network firewalls, whereas other protocols are normally blocked by firewalls. SOAP 1.1 was proposed to W3C in May 2000 [10]. Currently, W3C is working on SOAP 1.2. A SOAP message is an XML document which contains the following XML elements: Envelop Header, Body and Fault. The SOAP Envelop element is the root element of a SOAP message. It is used to identify that the XML document is a SOAP message and encapsulates all other XML elements. The SOAP Header element is optional and may contain application specific information like security features. The SOAP Body element is required and contains call and response information. The SOAP Fault element is optional and it represents error information when processing the message. Each Fault element must contain a fault Code element and a fault String element. The former one represents the code of the error and the latter one is used to provide the detailed information of the error. Figure 2 depicts how SOAP messages exchanged between SOAP client and server. A SOAP client (also termed as service requester) is an application that creates a SOAP message containing the information needed to invoke remote methods in heterogeneous environments. It could be a traditional program, Web services or any other server-based applications (such as servlet, port let etc.). A SOAP server (also termed as service provider) is a program generally resides in the Web server that listens, distributes and interprets the received SOAP messages. In Figure 2, the SOAP client sends a request (a SOAP message) to the SOAP server via HTTP/HTTPs. After receiving this request, the SOAP server sends it to the SOAP processor which resides in the server side and acts as an evaluator to validate this request against the XML schema. If it is valid, the SOAP processor invokes the Web service. Then the Web service processes this request and Creates a response. After that, the SOAP processor wraps this response with SOAP message format and sends the response SOAP message back to the SOAP client over HTTP/HTTPs. Like the SOAP server, the SOAP processor in the SOAP client will parse and validate this response message and present the result to the SOAP client user interface.

Figure 2: Exchanging Message with SOAP

C. WSDL

The WSDL (Web Services Description Language) is a specification recommended by W3C [10]. The current draft version is 2.0. It is an XML-based language used for describing Web services. Each WSDL document contains have major elements which describe three aspects of Web services. The types, message and port Type elements are used to describe what tasks the service provides. The type's element defines the data type used by Web services (Generally, for maximum platform independent purpose, WSDL uses XML Schema syntax to define data types). The message element defines the data elements within each operation. These data elements can be compared to the parameters of a function call in traditional programming languages. The port Type element describes the operations that a Web service can perform. It can be compared to a function in traditional programming languages. The binding element defines the message format and underlying transport protocol details for each port and the service element represents the location of the service. The main purpose of the WSDL is to provide human readable and machine interpretable information about the service.

D. UDDI

UDDI (Universal Description, Discovery and Integration), sponsored by OASIS [7], is a platform-independent, XML-based registry, which enables Web services implementations to be published and discovered either within or between enterprises. A UDDI business registration contains three components: White Pages (describes basic information about the service provider, such

as address, contact, and other identifiers), Yellow Pages (describes services by industry categorizations, service type or geography information) and Green Pages (describes technical information about services, such as interfaces and URL locations etc.) UDDI eases enterprises to create a platform-independent and open architecture for quickly discovering and publishing businesses and services via Internet. Enterprises can describe and publish their services to the UDDI registry. Once an enterprise finds a potential business partner, they can quickly and easily begin trading. As shown in Figure 1, the service provider may publish its WSDL document in the UDDI registry and the service requester may retrieve it by searching the UDDI registry according to the keywords described in the above pages. When the service requester obtains the WSDL document for a service, it may invoke it according to the elements described in the WSDL document.

3. PROPOSED METHODOLOGY

The proposed Method is tractable and admissible as the number of action nodes may be exponential in the number of service parameters in the worst case. Furthermore; there lies the issue of security and storage'sprototype version tested upon heart stroke patients till date, Narrowed research. Data transfer process Alert process for a specific doctor or a group of doctors working in shifts Needs of the users are often uncertain,diverse.sequential process models cannot meet the existing needs of the users. Thus, the existing method fails to address this issue.

A. FUNCTIONAL REQUIREMENTS:

The design of healthcare system aims to meet a collection of functional requirements as follows [1].

1. User Personalization. Users of an e-Health portal can create and save a personalized page including only the content they would like to access. For example, a patient may prefer seeing only the newsfeed in cardiology.
2. Content Aggregation. Users of the portal can access related services on a single page, regardless how many service providers are involved or how different those services are implemented. Navigation elements should be provided such that users can easily switch to a different page when necessary.
3. Ease of Use. This requirement is particularly relevant to e-Health[7] systems where many users are seniors or have limited knowledge of computer technology and even the installation of client-side software may go beyond their capability.[2]
4. Backend Customization. Administrators (Top mgmt.) can customize the source of services provided by the portal using a content management system, and such modifications should be transparent to normal users.
5. Interoperability. The portal must be able to seamlessly integrate heterogeneous medical services implemented on different platforms and with different technologies. Such implementation details should be transparent to users of the portal.
6. Extensibility. The integration of new services should imply minimal impact on the normal operation of other services provided by the same portal. Downtime should be minimized because the continuous availability of medical services may have direct impact on patients' health or life.

B. PROBLEM ADDRESSED

The proposed algorithm is tractable and admissible as the number of action nodes may be exponential in the number of service parameters in the worst case. Furthermore; there lies the issue of security [5] and storage's prototype version tested upon heart stroke patients till date, Narrowed research. Data transfer process Alert process for a specific doctor or a group of doctors working in shifts Needs of the users are often uncertain,diverse.sequential process models cannot meet the existing needs of the users. Thus, the existing method fails to address this issue.

C. ALGORITHM

1. Service Composition Planner:

In the algorithm, each action node N in the generated activity diagram represents an available service from community C. Moreover, an information state S is associated with N to denote the current status after executing the service that the node represents. It contains the currently available input and output parameters and such information as what conditions are presently met. This is the basis for the next step reasoning. In service composition based on input and output parameters only, state S can be just a collection of parameters.

The associated state of the initial node N_0 is $S_0 = I_A$, which is the set of available inputs that the service requester provides. At S, we may have a set of services $\Omega(S)$, each of which can be invoked. After every step of reasoning, we check the set of newly obtained executable services to see whether any concerned service appears or not. If yes, we have to introduce a branch structure into the activity diagram under certain conditions. At the end of the algorithm, the set R_S records all of the action nodes whose corresponding states can satisfy the user's requirement. According to these nodes, we can find out all of the feasible solutions from the constructed diagram. $R_S = \emptyset$ indicates that there are no solutions to this WSC problem

2. Processing Preference with Multiple Cases (PPMC):

Algorithm **PPMC** is a sub procedure used to introduce multiple branch structures into the activity diagram when some conditions are satisfied. In the reasoning process, once the services mentioned in the user's preference (i.e., w_1, w_2, w_n in PPMC) appear, we will examine whether the truth value of condition formula P can be acquired and will prepare to introduce a branch structure into the diagram.

First, P can be evaluated under the current state S. In this case, we can introduce a decision node into the diagram directly, whose input comes from the action N (which is being processed in **SCP**).

Second, P cannot be evaluated under S. However, in the set of services $\Omega(S)$ that can be executed currently, there is a service such that the state obtained after executing it can result in the truth value of P. Under this circumstance, after the action corresponding to the service, we introduce a decision node.

Third, if both aforementioned cases are not satisfied, we continue the reasoning process based on the action set N *S*temp which is corresponding to the invocable service set $\Omega(S)$, until there exists an action whose state can result in the truth value of P. Here, the reasoning process is similar to the one in **SCP**.

After the decision node is introduced, no matter which case is satisfied, we draw connectors from it to Np_1, Np_2, and Np_n (corresponding to the services designated), respectively. Meanwhile, set the corresponding guard (C_1, C_2, and C_n) for these control flows. Then, we introduce a merge node into the diagram, whose inputs originate from Np_1, Np_2, and Np_n. After that, we compute the corresponding state S_M of merge node N_M and add N_M into N_S to complete the connection with **SCP**. The merge node is also with a state for reasoning here for the simplicity of our algorithm. If none of these three cases is met, the multiple branching structures cannot be introduced into the diagram at present.

5. EXPERIMENTAL SETUP:

This section describes the implementation details of a prototype of the software module for this project. The system is designed to integrate existing medical systems, application, and services.

1. ECG Monitoring Service(Case 1):

The ECG monitoring service is in real-time mode. For the automated mode, sophisticated algorithms will be needed to analyse the ECG signals, which is out of the scope of this research. The ECG monitoring service contains three components: a patient-side applet, a medical staff-side applet, and an ECG monitoring server. This patient-side applet is based on the toolkit ECGSYN which is a realistic ECG waveform generator in the PhysioNet toolkits. It can generate a synthesized ECG signal with user-settable mean heart rate, number of beats, etc. PhysioNet is an Internet resource for biomedical research and development sponsored by the NIH's National Center for Research Resources. In real world, the ECG signals should be gathered from the external medical devices which are wired (such as USB, RS-232 etc.) or wireless (Wi-Fi, Bluetooth, IR etc.), and are connected to a computer. For demonstration purposes, we use an ECG signal generator to simulate this procedure. We also simplify the implementation of the toolkit and add our own features, such as signal transmission, Kerberos functions, etc. The ECG monitoring server is a java-based application that acts as a repeater to forward signals to the medical staff side and stores the ECG signal data into a database. When a patient wants to use the ECG monitoring services/he needs to simply click a desired appointment in the ECG appointment request panel, and the applet will be downloaded and run in the patient's local machine. The patient can then perform the monitoring via the applet interface.

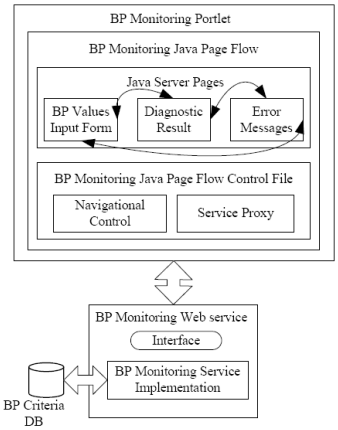

Figure. 3

6. RESULTS:

Interoperability and extensibility are both inherent to the architecture because most of the backend ser- vices (run in automation or store-and-forward mode, e.g. the BP monitoring service is running in automation mode) can be integrated in a Web services manner. Service provider (backend services) and service requester (port let) can publish and find desired medical services via UDDI, respectively. Once a new medical service is integrated into the system, it can be described by a WSDL file and this file will be sent to the UDDI. After discovering the binding information of the services via UDDI, service requester will communicate with backend services through standard SOAP regardless of what kinds of platforms are used in each side. To access a service, a user connects to the portal server via a standard Web browser.[9] According to the user's personalized settings and the entitlement, a portal interface is displayed with a collection of port lets inside it. When the user clicks on a button encapsulated in

a port let, the corresponding action of the service will be performed at backend service providers (which may be governed by a remote portal). As an exception, services run in real-time mode (such as the ECG Monitoring service and Teleconsultation service) are allowed to bypass the portal since they usually demand better performance than what SOAP can provide. Thus, an applet or activeX control downloaded to the browser will perform the required actions on behalf of the users. From the above descriptions, the proposed architecture clearly meets all requirements stated.

Figure.4 Patient Side ECG Monitoring Interface

7. CONCLUSION AND FUTURE WORK:

7.1. Conclusion:

In our prototype system, real-time ECG monitoring service, BP monitoring service, have been implemented and integrated into our e-Health portal. Our solution was based on a two-tier access control architecture that integrated existing modules with a rule-based access control extension.

7.2. Future Work:

The proposed architecture may have been redesigned as a multi-tier architecture supporting more services. Future researches are listed below:

7.2.1. Extend rule-based access control engine to enforce authorization of workflow-based services.

7.2.2. Quality of Services (QoS) is an important factor to be considered. QoS includes the scalability and performance of Web service-based services, the performance of the access control engine, the delay, jitter, response time etc., of real-time services etc. In the future work, we will investigate those QoS issues.

7.2.3. The WSRP standard does not address any security standard currently. Thus, for the authentication of consumer, the authentication of end user to the producer, message integrity and confidentiality should be enforced in a standard way

REFERENCES

[1] G. Eysenbach, “What Is e-Health? “Journal of Medical Internet Research, vol. 3, no. 2, p. e20, 2001.

[2] G. Bisson, “e-Health Portal and SNOMED for a More Personalized Integrated EHR,” Proc. UM2005 Workshop on Personalization for e-Health, 2005.

[3] S. Lu, Y. Hong, Q. Liu, L. Wang, and R. Dssouli, “Access control for e-health system portal,”Proc. 4th International Conference on Innovations in Information Technology (Innovations 2007), IEEE, 2007.

[4] Y. Hong, S. Lu, Q. Liu, L. Wang, and R. Dssouli, “A hierarchical approach to the specification of privacy preferences,”Proc. 4th International Conference on Innovations in Information Technology (Innovations 2007), IEEE, 2007.

[5] Q. Liu, S. Lu, Y. Hong, L. Wang, and R. Dssouli, “Securing telehealth applications in a web-based e-health portal,” Proc. 3rd International Conference on Availability, Reliability and Security (ARES 2008), IEEE, 2008.

[6] Y. Hong, S. Lu, Q. Liu, L. Wang, and R. Dssouli,“Preserving privacy in e-health systems using Hippocratic databases, “in Proceedings of the 32nd Annual International Computer Software and Applications Conference, 2008.

[7] “http://www.oasis-open.org/,”

[8] C. Koch, “A new blueprint for the enterprise,”CIO Magazine, 2005.

[9] A. Gokhale, B. Kumar, and A. Sahuguet, “Reinventing the wheel? Corba vs. web services,” The 11th International World Wide Web Conference, 2002.

[10] “http://www.w3c.org/,”

[11] “http://edocs.bea.com/wlp/docs81/index.html,”

[12] “http://jcp.org/en/jsr/detail?id=168,”

[13] “http://portals.apache.org/pluto/,”

[14] J. Fayn, C. Ghedira, D. Telisson, H. Atoui, J. Placide, L. Simon-Chautemps, P. Chevalier, and P. Rubel, “Towards new integrated information and communication infrastructures in e-health: Examples from cardiology,”Computers in Cardiology, 2003.