An SOA-based diseases notification system

Conference Paper · January 2010

DOI: 10.1109/ICICS.2009.5397519 · Source: IEEE Xplore

3 authors:

Chi Po Cheong
PJ
10 PUBLICATIONS **37** CITATIONS
SEE PROFILE

Chris R Chatwin
University of Sussex
537 PUBLICATIONS **4,107** CITATIONS
SEE PROFILE

Rupert C D Young
University of Sussex
311 PUBLICATIONS **2,870** CITATIONS
SEE PROFILE

An SOA-Based Diseases Notification System

Chi Po Cheong
School of Science and Technology
University of Sussex
Brighton, United Kingdom
webster@macau.ctm.net

Chris Chatwin
School of Science and Technology
University of Sussex
Brighton, United Kingdom
C.R.Chatwin@sussex.ac.hk

Rupert Young
School of Science and Technology
University of Sussex
Brighton, United Kingdom
R.C.D.Young@sussex.ac.uk

Abstract— This paper proposes a Diseases Notification System which is designed using a SOA pattern. Disease notification is an import component of the disease control systems. A successful diseases notification system will deliver positive effects for human beings. The proposed system is designed and implemented by use of the Business Process Execution Language for the business process layer and Enterprise Service Bus for the connectivity layer. Therefore, it can cope with a dynamically changing environment or requirements. Moreover, the proposed system can reduce the notification process time and it can provide timely information on diseases notification.

Keywords – SOA, Diseases Notification, BPEL, ESB

I. INTRODUCTION

Many computer-based systems or tools are used for monitoring of disease migration. Some provide record-based information and some provide visual data, examples are given in [1], [2]. A location-based data visualization system can be of great benefit for disease monitoring and tracking. However, a successful disease control system is based on cumulative recording of the occurrence of spreading diseases. Diseases notification and tracking is a vital component in ensuring protection of public health. To prevent and control the spread of infectious disease around the world, health organizations must monitor trends over time not only in human diseases such as chickenpox, tuberculosis, plague, HIV, Severe Acute Respiratory (SARS) especially, the currently spreading disease: Influenza A H1N1, etc. and even animal diseases such as bird flu H5N1.

A. Motivation

The existing notifiable diseases exchange mechanisms rely on traditional means of communication such as postal mail, telephone, facsimile and email. Health care professionals, who make first contact with a patient who has contracted a notifiable disease, are required to report to the local health monitoring authority such as Centers for Disease Control and Prevention (CDC) for both suspected and confirmed cases. The case is then submitted to other foreign health organizations such as the World Health Organization (WHO). However, the reporting system remains largely paper-based and handwritten sources are often difficult to read. Hence there are many hand-written errors in vital information such as diagnosis and demographic data. A consistent and accurate diseases notification system is proposed in this paper.

B. Contribution of this paper

The paper proposes an SOA-Based disease Notification System. It will be implemented by Web Services with an Enterprise Service Bus (ESB) for the connectivity layer and use Business Process Execution Language (BPEL) for the business process layer. The proposed system not only provides timely and accurate data on diseases notification but also complements existing legacy health information systems in preventing emerging infectious diseases which are an increasing threat with the accelerating rate of globalization in the twenty-first Century.

II. WHY SOA

Service Oriented Architecture (SOA) is a design architecture pattern used in the software design phase. Generally, a large complex system is composed of user interfaces, business logics and databases. The business logics are the core of the system. If the business logics are tightly integrated, the system cannot cope with a dynamically changing business environment. The difficulties can be solved by decomposing the business logics into smaller and loosely coupling components. The business logic in SOA is called a service. The requirements of a SOA-based system are loose coupling, open standards, a high degree of standardization and technology independent.

A. Grid and Cloud Computing

Distributed computing works by splitting up the larger task into smaller chunks [3]. Grid computing is a distributed computing concept which is used for sharing and integrating computing resources, it evolved from heterogeneous systems. Grid computing aims to solve the common IT problem of dedicated and underutilized hardware resources [4]. It is also used to divide a large task into many tasks that run in parallel on separate servers. The success of the Internet leads to the concept of cloud computing [5]. A cloud computing infrastructure can do large-scale processing and it is massively scalable. Due to the nature of the Internet, cloud computing can break down the physical barriers and link the hardware and software platforms in different global locations through the Internet. Thousands or millions computers can create an enormous machine pool. A complex and time-consuming task running on a vast amount of computers can obtain an expected result within minutes instead of weeks or months. The differences between cloud computing, grid computing and

distributed systems are shown in [6]. Performance is one of the issues in an SOA-based system. The invention of cloud computing may solve the performance problem.

B. Select the optimal architecure

In an organization, several system architectures are adapted to design a system. It is impossible to use a single architecture to handle all business requirements. The "best of breed" concept [7] in system integration can also be used in selecting a system architecture for a particular application. For instance, a two-tier client-server architecture can be used to deploy a simple On-Line Transaction Processing (OLTP) system or Point of Sale System (POS). The business or application logics are placed in the client side software (fat client). 3-tier or N-tier architecture can be used in Internet-based system. The business logics are placed in an application server such as Oracle WebLogic Application Server and the client side software usually is a Web browser (thin client). Client-server architecture, 3-tier architecture and N-tier architecture are successful widely adopted distributed architectures. However, a large system deployment will leads to "server sprawl" [8].

An SOA-based system is also a kind of distributed system and similar to the Internet-based system. One of the major differences is the level of dependence of business logics in the design phase. A traditional Internet-based system decomposes the business logics into more tightly-coupled components but SOA splits it into loosely coupling components and the SOA relies on components and creation of services. The other difference is that the system architecture of a SOA-based system can be changed after deployment. Therefore, a new service can be added in the system without stopping the whole system. However, there is significant confusion about SOA; such as SOA must be implemented using Web Services. The common misconceptions about SOA are discussed in [9] and the classification of SOA-based systems are discussed in [10]. The potential benefits of SOA are services reuse, integration improvement, leveraging the legacy investment and best of breed integration [11]. SOA is suitable to design a distributed, Internet-based, dynamic change, autonomous and non point to point system. Service-Oriented Architecture (SOA) is adopted as an architectural approach to design services which can be dynamically selected, reused and created by various health organizations such as the World Health Organization (WHO), Centers for Disease Control (CDC), hospitals, etc. In addition, an SOA-based system can be integrated with the existing legacy health information system such as: inpatient system, laboratory system, etc. and can support interoperation with external systems such as an external notifiable diseases reporting system.

III. SOA-BASED DISEASES NOTIFICATION SYSTEM

The proposed system, SOA-based Diseases Notification System (SOADNS) utilizes medical standards to exchange or identify notifiable diseases, including: International Classification of Diseases (ICD) [12] and Health Level Seven (HL7) [13]. A doctor can use the ICD code to classify each disease. For instance, code "487" represents "Influenza" in ICD version 9CM and code "J11" represents "Influenza, virus not identified" in ICD version 10. The SOADNS has two roles, one is service provider and other is service consumer. For instance, it provides services to the disease declarer and consumes service provided from the local health authority. A service consumer can discover the service by the Universal Description and Discovery and Integration (UDDI) [14] or service brokers. Earlier SOA-based systems can be implemented using different connection technologies such as Distributed Component Object Model (DCOM) or Common Object Request Broker (CORBA) [15]. However, the proposed system uses the Web service which will be implemented by the Simple Object Access Protocol (SOAP), Web Service Definition Language (WSDL), etc. to implement the system. Figure 1 shows the high level architecture of the proposed system. A clinic or a private doctor can declare notifiable diseases by use of a Web interface provided by the local health authority. The Hospital acts as a service consumer and submits the notifiable diseases by use of a standard SOA client application via SOAP message format. The hospital can also act as a service provider to provide Web services to other service consumers. For instance, it can provide the details of patient records to the WHO if necessary. The SOADNS is designed as a standard application package. Therefore, it can be deployed to every participant and only requires minor changes in the business process layer.

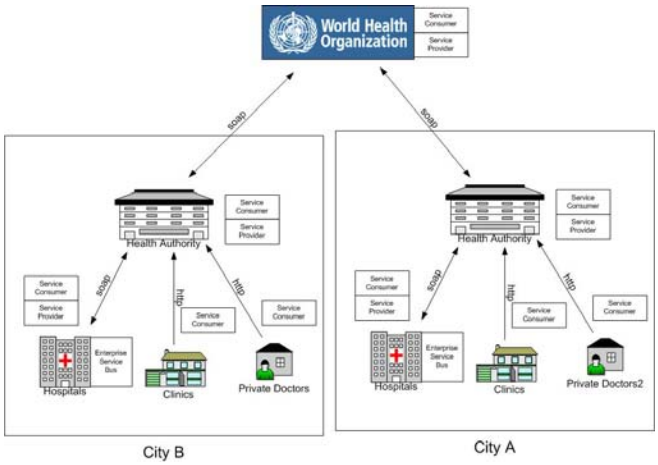

Figure 1. High Level Architecture Diagram of SOADNS

The proposed system is composed of five layers including: presentation layer, service layer, business process layer, connectivity layer and data layer. The presentation layer - the client side uses Web browser to input data, such as patient records and diseases notification data. The service layer contains all the services provided from the proposed system. The service can encapsulate one or more business logics of the system and it has an interface for the service consumer to interact with. Services are orchestrated into business processes in the business process layer. The Business Process Execution Language (BPEL) is used in this layer and it can specify which services should be invoked and the sequence in which they are invoked, which can ensure the business flow is in the correct order. Enterprise Service Bus (ESB) is used in the connectivity layer. It can move, transform and route the clinical data within the inside system and between the outside systems in XML format. The data layer includes the database of diseases

notification data, demographic data and it can be used through an adapter service defined in connectivity layer, for instance, a database adapter service. The details of service layer, business process layer and connectivity layer are discussed in the following sections.

A. Service Layer

Service modeling is one of the vital processes to build a SOA-based System. The process decomposes the business logics into several generic processes or services. There are several core services in SOADNS including: CodeValidationService, NotificationService, TaskService and DatabaseAdapterService. The NotificationService is used for notifying guarders when a serious incident occurred has via email, fax, pager or SMS. The ICD Code is validated by the CodeValidationService. It can handle the different versions of the disease codes. The TaskService is a human workflow task and it can interweave human interactions with connectivity to the SOADNS. The human task is linked to the Business Process Layer through a WSDL. In the SOADNS, the TaskService is used for the human re-approval and re-confirmed for the disease being monitored before notifying the emergency committee. DatabaseAdapterService provides a bridge between the database and the business process layer. All the services can interact with each other at the message level and are described in Web Service Orchestration language.

B. Business Process Layer

The Business Process Execution Language (BPEL) [16] is used in this layer for specifying interactions with Web Services including the business logic and the execution order. The BPEL process of SOADNS is shown in Figure 2.

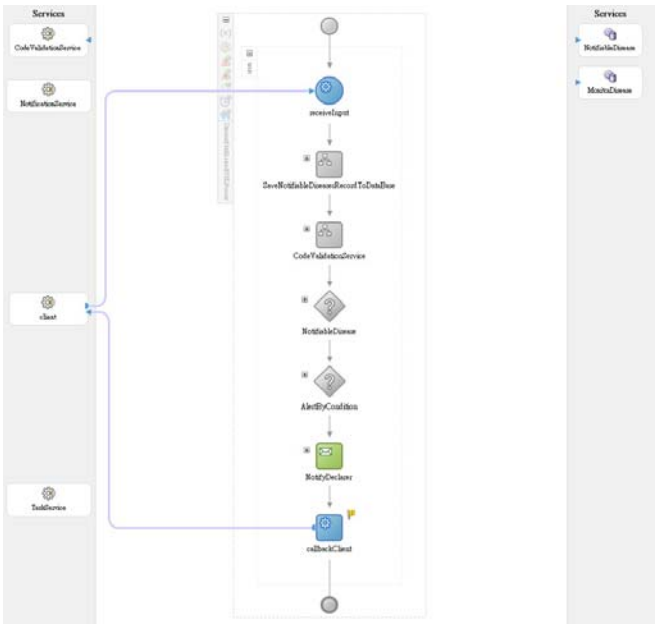

Figure 2. BPEL Process of SOADNS

For instance, a doctor submits a notifiable disease through a web browser. The doctor can input different versions of ICD code. Then, the BPEL process is invoked by a message from the application server or ESB. The BPEL process writes the notifiable disease record to the database. The BPEL process calls the CodeValidationService to valid the disease code. If the disease code is valid, it will check whether it is required to notify a monitor or a committee member by using NotifialeDiseaseDBAdapter. The committee or members of the WHO may receive an alert message under certain circumstances. Finally, an email will be sent back to a declarer for confirmation. Figure 3 shows the details of AlertByCondition switch. If it is a confirmed case, the system will send an alert message through SMS or a pager to the emergency committee. Otherwise, if it is a suspected case, the system will only send an alert message via an email to a general officer for example.

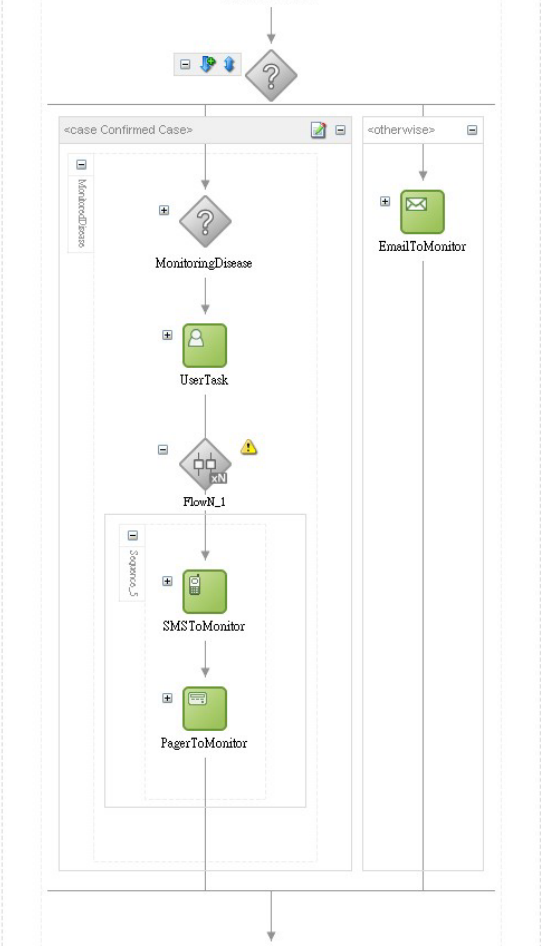

Figure 3. Details of AlertByCondition Switch

C. Connectivity Layer

In a hospital environment, many different architecture systems, which include hardware and software are used for different purposes. For instance, Health or Hospital Information System is used for recording patient demographic and clinical information. The Lab or laboratory management system (LMS) is used in the laboratory for the management of the patient's samples such as blood test samples. It requires an

interfacing program between the LMS and laboratory equipments. Different systems produce outputs in different formats, such as text, XML, HL7, etc. Therefore, the Enterprise Service Bus (ESB) can be used to put everything together. The ESB is suitable for use in the hospital environment. It can collect the output coming from different platforms and in different formats. It means that the disease code not only comes from user input through a Web user interface, it can also be obtained from heterogeneous systems automatically.

D. System Implementation

The Oracle SOA Suite [17] which includes BPEL Process manager, Enterprise Service Bus, Web Service Manager, etc. is adopted to implement the proposed system. Oracle JDeveloper is also used as a development tool to help developers to model, create, test, deploy and maintain a services-based application. All the developed services, web sites and J2EE applications are running on the Oracle Application Server. Web service standards such as XML, SOAP, WSDL are also used within the proposed system. The implementation technologies of SOA are ripe for exploitation. However, a major issue for the deployment of the proposed system is legislation in each jurisdiction. The details of notification mechanisms are discussed in [18].

IV. ADVANTAGES OF SOADNS

The SOADNS is designed in a SOA style. Therefore, it can obtain the advantages of SOA such as business logic reusability, loosely-coupled to other services, etc. The BPEL and ESB are also used to make the system complete. The system can be dynamically changed by modifying the BPEL process in the business process layer. Therefore, the system is easy to spread and to deploy. The ESB is used in the SOADNS. It can accommodate the data of exiting medical systems, such as laboratory management system, X-Ray management system, etc. The SOADNS is composed of services, BPEL and ESB. Therefore, it increases flexibility, it is easy to change the system architecture and it scales from a point to point solution to a distributed solution.

V. CONCLUSIONS

This paper has discussed the importance of diseases notification and the reasons for adopting the SOA to design the proposed system. Moreover, a SOA-based Diseases Notification System (SOADNS) is proposed. The SOADNS can make diseases notification more effective and provide timely information. It uses BPEL as a standards-based way of orchestrating all services. The system architecture can be changed easily by modifying the business process layer through BPEL. Therefore, the SOADNS can be deployed by different participants without modifying the application source code. The ESB is used to collect notifiable diseases data in different formats from heterogeneous systems. The proposed system is a complete solution for diseases notification in and from different locations.

REFERENCES

[1] L. C. Chang, H. K. Chiang, W. Y. Chiang, Smart-GIS: “A SVG-based tool for Visualizing and Monitoring of SARS Movement”, in Proceedings of International Conference on Information Technology: Research and Education, ITRE 2005, 3rd, pp. 282-286, 2005.

[2] Z. Qian, L. Zhang, J. Yang, C. Yang, “Global SARS information WebGIS design and development”, in Proceedings of International Geoscience and Remote Sensing Symposium, IGARSS’04, pp. 2861-2863, 2004.

[3] B. Godfrey, “A primer on distributed computing”, www.bacchae.co.uk/docs/dist.html.

[4] Gird Computing with Oracle, An Oracle Technical White Paper, Oracle Corporation, 2005.

[5] G. Boss, P. Malladi, D. Quan, L. Legregni, H. Hall, “Cloud Computing”, IBM white paper, IBM corporation, 2007

[6] I. Foster, Y. Zhao, I. Raicu, S. Lu, “Cloud Computing and Grid Computing 360-Degree Compared”, in Proceedings of Grid Computing Environments, Workshop, GCE ’08, pp. 1-10, 2008.

[7] P. Wirz, M. Lusti, “Information Technology Strategies in Mergers and Acquisitions – An Empirical Survey”, in Proceedings of the winter international synposium on Information and communication technologies, ACM International Conference Proceeding Series; Vol. 58, 2004.

[8] E. C. Leno, J. He, “IT and business Integration through the Convergence of Virtualization, SOA and Distributed Computing”, in Proceedings of International Conference on e-Business Engineering, ICEBE’08, pp. 615-620, 2008.

[9] G. A. Lewis, E. Morris, S. Simanta, L. Wrage, “Common Misconceptions about Service-Oriented Architecture”, in Proceedings of 6th International Conference on Commerical-off-the-Shelf (COTS)-Based Software Systems, ICCBSS’07), pp. 123-130, 2007.

[10] W.T. Tsai, C. Fan, Y. Chen, R. Paul, J. Y. Chung, “Architecture Classification for SOA-Based Application”, in Proceedings of 9th International Symposium on Object and Component-Oriented Real-Time Distributed Computing, pp. 295-302, 2006.

[11] Thomas Erl, “Service-Oriented Architecture: Concepts, Technology, and Design", Chapter 3 Introduction SOA, Session 3.4 Common tangible benefits of SOA, Prentice Hall PTR, 2005.

[12] World Health Organization, International Classification of Diseases (ICD) , http://www.who.int/classifications/icd/en/

[13] American National Standard HL7 V3-2006, HL7 Version 3 Normative Edition, ANSI, 2006.

[14] L. Clement, A. Hately, C. Riegen, T. Rogers, “UDDI Version 3.0.2 UDDI Spec Technical Committee Draft”, OASIS, 2004, http://www.uddi.org/pubs/uddi_v3.htm

[15] I. Wong-Bushby, R. Egan, C. Isasscon, “A Case Study in SOA and Re-Architecture at Company ABC”, in Proceedings of the 39th Hawaii International Conference on System Sciences, pp. 179b-179n, 2006.

[16] D. Jordan, J. Evdemon, (Chairs), “Web Services Business Process Execution Language Version 2.0”, OASIS Standard, 2007.

[17] Oracle SOA Suite, http://www.oracle.com/technology/products/soa/soasuite/index.html.

[18] National Public Health Partnership Legislation Reform Working Group, “Notifiable Diseases & Notification Mechanisms”, Endorsed by AHMAC November 2000.