National Healthcare Information System Integration: A service oriented approach

USHA BATRA[1], SAURABH MUKHARJEE[2]
Department of CSE
ITM University[1], Banasthali Vidyapeeth University[2]
Gurgaon[1], Banasthali[2]
INDIA
batrausha@rediffmail.com[1], mukherjee.saurabh@rediffmail.com[2]

Abstract- Healthcare in our home country, India is a cause of concern even after 63 years of Independence. There is a need to create world-class medical infrastructure in India and to make it more accessible and affordable to a large cross section of our people. Introduction of information technology in healthcare system may eventually enhance the overall quality of national standards. The success in current healthcare system requires reengineering of healthcare infrastructure for India. For this, there is a high requirement in India to invest in IT infrastructure to provide interoperability in healthcare information system. Also, integration of IT with healthcare system may lead to open connectivity at all levels (i.e. InPatient and OutPatient care), ensuring that patient information is available anytime and right at the point of care, eliminating unnecessary delay in treatment, avoiding replication of test reports, improving more informed decisions and hence leading to improved quality of care.With this intent, this paper attempts to present software design patterns for Service Oriented Architecture (SOA) and its related technologies for integrating both intra and inter enterprise stovepipe applications in healthcare enterprise to avoid replication of business processes and data repositories. We aim to develop a common virtual environment for intra and inter enterprise wide applications in National Healthcare Information System (NHIS). The ultimate goal is to present a systematic requirement driven approach for building an Enterprise Application Integration (EAI) solution using the Service Oriented Architecture and Message Oriented Middleware (MOM) principles. We aim to discuss the design concept of Enterprise Application Integration for integration of a healthcare organization and its business partners to communicate with each other in a heterogeneous network in a seamless way.

Keywords— Enterprise Application Integration (EAI), heterogeneous, Message Oriented Middleware (MOM), Service Oriented Architecture (SOA), stovepipe applications, National Healthcare Information System(NHIS)

I. INTRODUCTION

Recently there is a growing interest in propagating the deployment of successful service oriented architecture for integration of enterprise wide applications in Indian healthcare systems. A number of approaches to the design of enterprise application integration have been proposed, e.g. [1, 11, 13, and 14]. An information system is mainly built on formula of point to point communication between applications. This results in large number of connections and hence leads to muddle integration inside and across enterprises. Message Oriented Middleware is a middleware technology and supports a set of Service Oriented architecture infrastructure that allows applications and data sources to share information and hence integrate stovepipe applications in heterogeneous distributed environment. Enterprise Application Integration is a support to minimize downtime and increase value by reducing the number of connections to a large extent. It is very common to talk either of IT or business applications but very rarely we talk about integration of IT with business applications. Today's successful companies must manage to increase reliance by having the right system in place and providing its customers a common virtual environment. According to research firm Gartner, application integration accounts for approximately 35 percent of the total cost of application design, development and maintenance in an enterprise [14]. Enterprise Application Integration is a term attributed to the integration of variety of applications including stovepipe applications using a common middleware [2, 12]. Healthcare integration is a kind of enterprise integration that requires special attention. According to Dr. Kevin Fickenscher, executive Vice President , International Healthcare "A more reasoned approach will help make India an International leader in promoting best health practices, creating appropriate societal incentives, and deploying systems that actually provide the foundation for enhanced care, better access and lower cost"[15]. However, there are many hospitals that confirm the interoperability in terms of integration of applications and process within its network of hospitals. For example, Apollo, MAX, Fortis, Paras etc. Although all the hospitals of Apollo chain are interconnected but the Apollo group of hospitals are not connected to Apollo clinics, Apollo pharmacies etc. Also, Apollo hospitals are not connected to any other hospitals like MAX or Fortis or Paras and so is the case with other group of hospitals. So, in case patients goes to Apollo hospital and gets all the treatments at one time and suppose next time he goes to some other hospital of say Max group then he/she should undergo all the diagnosis and test reports again. But this duplication of test reports can totally be eliminated by enabling the healthcare infrastructure with IT (total integration of healthcare system leading to national healthcare information system) and thereby allowing the availability of patient information at the point of care. This paper aims to discuss the various design requirements for improving the healthcare system.

The aim is to integrate healthcare organization with its business partners in a manner that they all interoperate with each other in a heterogeneous environment over network in a seamless way. Message Oriented Middleware is a key enabling technology for helping meet interoperability requirements and avoid stovepipe systems [2, 3, 12]. The design concept of Service Oriented Architecture for successful enterprise application integration supports interoperability and allow 'on –the- fly' information exchange among different systems in a loosely coupled environment that follow a standard protocol.

The paper is organized as follows: in section 2 we introduce the middleware architecture; in section 3 we describe design patterns for Service Oriented Architecture; in section 4 we present Message Exchange Patterns for Service Oriented Architecture [4]. In section 5, we present the Application Level Design Patterns to show how interaction across different Hospitals is achieved with Simple Object Access Protocol message exchanges, the final section contains our conclusion and future directions.

II. IMPORTANT ASPECTS OF NHIS

We intend to design a specification for exchange of data between the various stake holders of the health care sector. To make the next giant leap, we need seamless interchange of data amongst the various healthcare enterprises at different levels. Sadly the heterogeneous nature and its mammoth size make such data interchange facility a huge challenge. We intend to design a national healthcare information system which allows various healthcare enterprises to exchange information in a seamless way in heterogeneous environment. The nation demands for an intelligent healthcare system that should major cover the following aspects:

- High Availability: Availability of patient care at the point of care
- Quality Healthcare: Quality of care vs. cost of care
- Affordable: to large cross section of people
- Reliability
- Payment System
- Medical Insurance a mandate
- Growth of Medical Tourism
- Growth of telemedicine

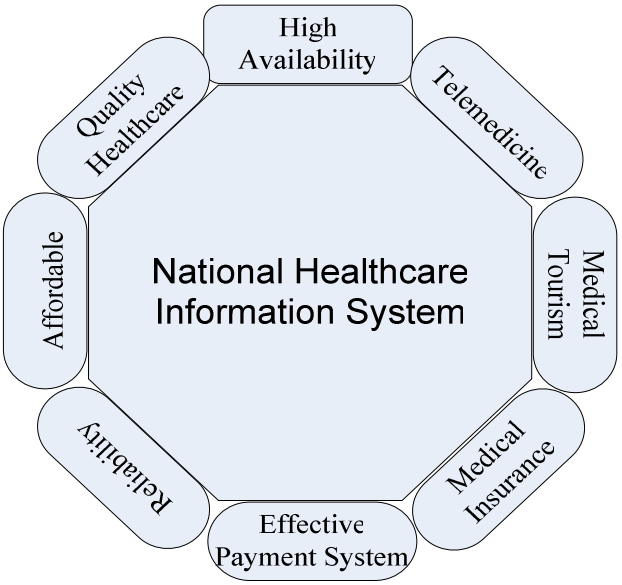

Figure 1: Important aspects of National Healthcare Information System

A. *The Middleware*

In Patterns of Enterprise Application Architecture [5], Martin Fowler warns that loosely coupled, distributed system architectures that look great on whiteboard can easily become "An architect's dream

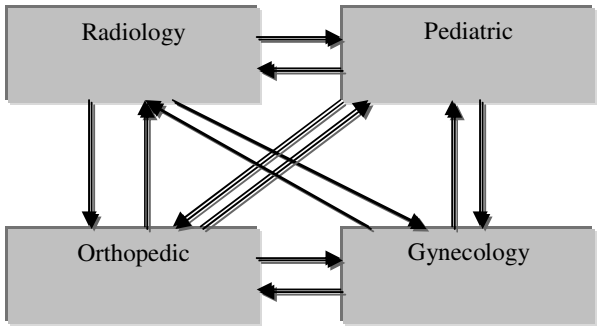

Figure 2 (a): point to point communication

and a developer's nightmare". Middleware is enabling technology for interoperability by adhering to services in distributed environment that have standard protocols. The term middleware attributes to software technology that solves heterogeneity and distribution problem and coins distributed services that have standard programming interface and protocols. ObjectWeb defines middleware as: "The software layer that lies between the operating system and applications on each side of a distributed computing system in a network" [6].

Not only this, middleware [7] is especially integral to modern information technology based on XML, SOAP, Web Services and Service Oriented Architecture. Liu Jing

Yong [8] categorized middleware into four categories. They are procedural middleware, message oriented middleware, transactional middleware, and object middleware. The message oriented middleware allows a variety of applications to exchange data and/or communicate by sending and receiving messages. The middleware lie in the middle layer, making this (integration) possible and does not require recreation of homogeneous components every time when the different applications want to interact in heterogeneous environment on different platforms.

The communication among varied applications in a distributed environment results in a mesh network i.e. every system is potentially connected to every other system in the network. A major problem with such mesh network is the number of possible connections, because if there are n systems in a network then there would be possibly N* (N-1)/2 connections if each system is connected with all other systems in the enterprise. A message oriented middleware solves this problem by introducing

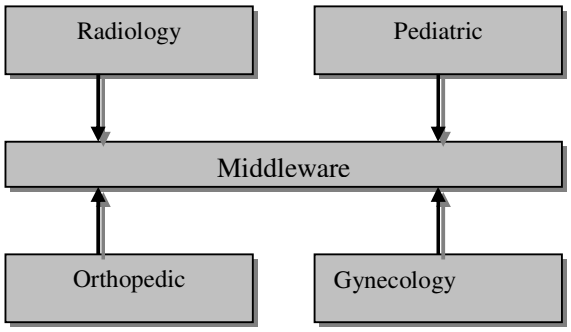

Fig. 2(b) Introducing middleware n/w in healthcare system

a message bus that receives messages from all the systems and routes them to appropriate receiver. We consider the example of four departments communicating in a healthcare system (see fig. 1 (a)), this creates a mesh. The number of connections creates even more complexity and results in unmanageable state when the number of departments increases. Moreover, it leads to tightly coupled point to point integration of different applications. The overall cost and complexity in integrating applications can be reduced by introducing a middleware which significantly reduces the number of connections (see fig. 1(b)). Now, each department has to pass its messages to middleware which routes them to appropriate department. The middleware layer allowed application decoupling; enabling N*(N-1)/2 connections in point-to-point technique to be reduced significantly to N number of connections. For example, as is shown in fig. 1 (a), only 4 applications are communicating in a point-to-point fashion which requires total of 6 connections, if we add more number of applications i.e. for 5 applications, the number of possible connections would be 10. Likewise, for 20 applications to fully integrated, we need 190 connections and for 200 applications 19,900 number of connections. This results in a mesh that is difficult to handle and it adds the overall cost of integration solution. In healthcare system each hospital comprises many applications and there is a requirement of integrating these applications, but point-to-point technique for integration is not suitable.

The commonly agreed communication protocol for communication among N number of applications can be achieved once all the stovepipe applications commonly agree on the middleware solution. Examples of commercial middleware solutions are IBM WebSphere Message Broker, Microsoft BizTalk server etc.

III. Service Oriented Architecture Design Patterns for NHIS

In 1996 Yeffim V. Natiz from Gartner defined SOA as " a style for multi-tier computing that help organizations share logic and data among multiple applications and usage modes." According to Grady Booch " Service Oriented Architecture Design Patterns is an important contribution to the literature and practice of building and delivering quality software intensive systems." The most important aspect of Service Oriented Architecture is that it separates the service's implementation from its interface. Service consumers view a service simply as a communication endpoint supporting a particular request format or contract [9]. How service executes service requested by consumers is irrelevant; the only mandatory requirement is that the service sends the response back to the consumer in the agreed format, specified in contract. In order to design large scale Service Oriented Architecture, we need to follow the following implementation standards:

Service Oriented Architecture can support variety of design patterns including: Asynchronous Messaging Patterns, Conversation Patterns, Orchestration Patterns, Process/Workflow Patterns, Endpoint Patterns, and Security Patterns etc.

Messaging is the backbone of Service Oriented Architecture. According to Robert Shimp, vice president of Technology Marketing at Oracle: "EAI requires specific knowledge of what each application provided ahead of time. Service Oriented Architecture views each application as a service provider and enables dynamic introspection of services via a common service directory, Universal Description Discovery and Integration of Web services (UDDI)."

A. Message Exchange Patterns for Service Oriented Architecture

The messaging design patterns [2, 11] allow the interchange of information by exchanging messages between components and applications. The Message exchange design pattern can be applied to varied

problems in a distributed scenario. Here, we propose a simplified interaction model of different hospitals, clinics, and other business partners in a healthcare system. The various components communicate through asynchronous message exchange. According to Gregor Hohpe[9] "Asynchronous messaging is fundamentally a pragmatic reaction to the problems of distributed systems."
Sending a message does not require both systems to be up and ready at the same time. Furthermore, thinking about the communication in an asynchronous manner forces developers to recognize that working with a remote application is slower, which encourages design of components with high cohesion (lots of work locally) and low adhesion (selective work remotely)." A healthcare system to be completely automated needs to provide a transparent access to remote components despite of protocol or the communication means used.
Service Oriented Architecture [4, 10, 21] and its supporting concepts SOAP (XML) is required to achieve a well organized EAI.

B. The proposed framework

Figure 3 depicts the theoretical connection of various hospital, clinics and pharmacies of healthcare system. These organizations are built on various platforms, data base management system, operating systems but call for communication and exchange of data among each other. The main objective of the proposed framework is to provide an effective solution of integration of existing applications to increase the revenue and overall productivity of healthcare system.

IV. UML Design Patterns

The Unified Modeling Language (UML) [17, 18] is an object-oriented design notation that provides basic building blocks to model software intensive systems, such as abstractions that represent structure and behavior of a system, relationships that state how the abstractions relate to each other, and diagrams that show interesting excerpts of a set of abstractions and relationships.
Using design patterns help produce good design, which helps produce good software. The ability to work with design patterns in conjunction with Unified Modelling Language (UML) is a major benefit. We need to determine who will be the end users of the universal healthcare system. For example, the various stakeholders for the NHIS are: Patient, Hospital, Private practicing doctor, Small clinics, Radiologist, Pathology, Pharmacies, Health Ministry, Food and Aid programs, Medical Research, Medical Insurance, etc. The use case scenario for maintaining patient information is shown in figure 4. the use case scenario for handling medical stock inventory is shown in figure 5.

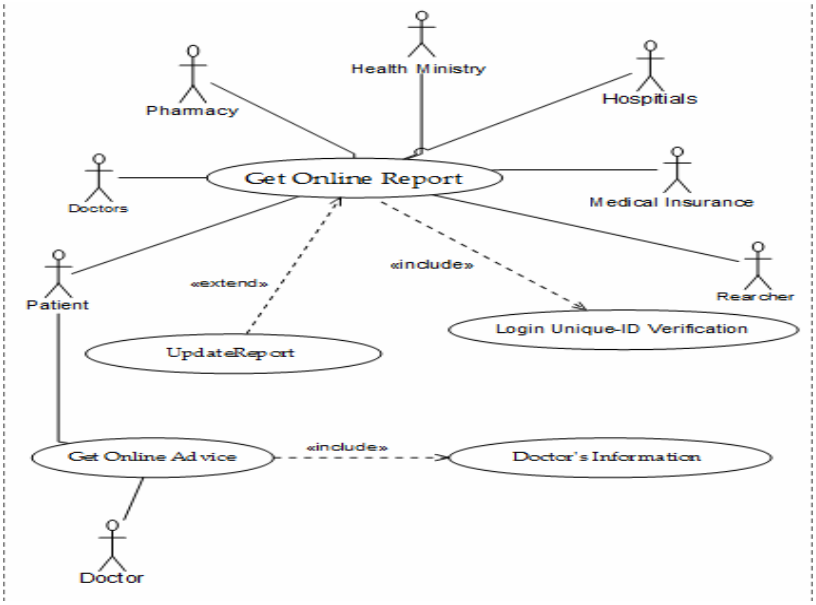

Figure 4 : Use Case scenario for accessing patient report information

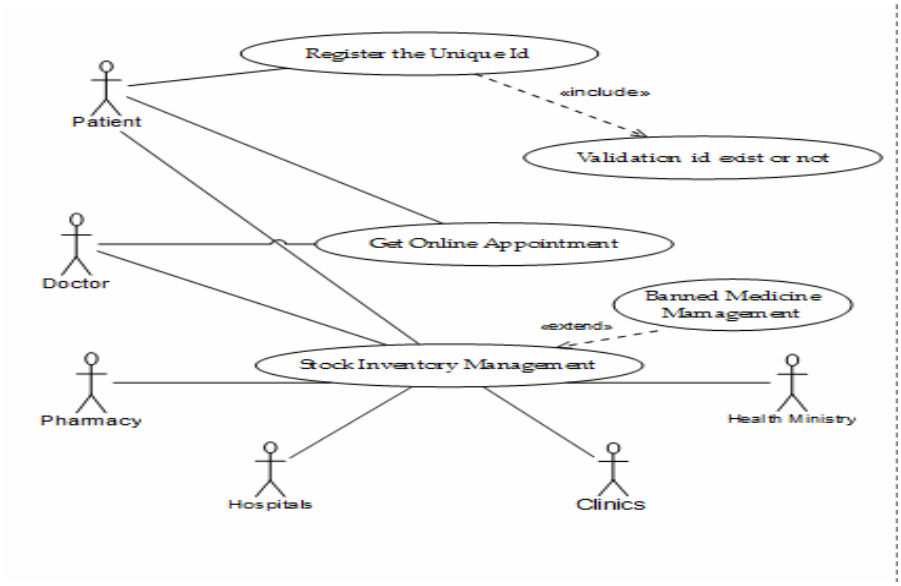

Figure 5: Use Case scenario for Stock Inventory(medicines) Management

V. Application Design Level Patterns: Web Services Design Approach

Service-oriented architectures (SOAs) and web service technologies have been proposed to respond to some central interoperability challenges of heterogeneous health information systems (HIS) [16].

A. Web Services

The main characteristic of a web service is that service provider publishes its service description and places it on service directory. [22] Service consumer queries the service directory to find out the kind of services available in the directory and to locate a service to communicate with the provider.

B. Web Services Definition Language (WSDL)

This is an industry accepted language in which service provider can write the description of its services before finally writing it to the directory.

C. Simple Object Access Protocol (SOAP)

SOAP is an industry accepted protocol to communicate with service directory. Service provider and service consumer talk to the service directory using SOAP protocol. SOAP uses Hypertext Transfer Protocol (HTTP) and its Extensible Markup Language (XML) as the mechanisms for information exchanges.

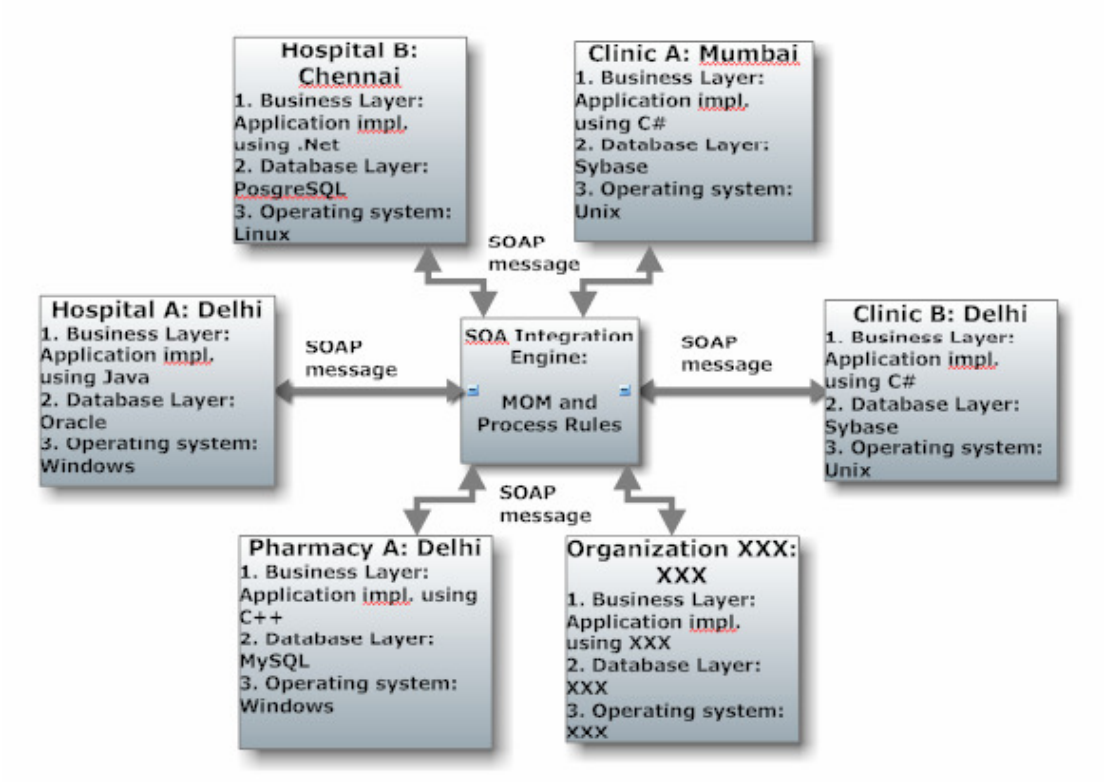

Figure 3: Integrated National Healthcare System

D. SOAP Request Message

As a first step Hospital A (Delhi) made a request to get details of patient in Hospital B (Chennai). The structure of SOAP request message asking for patient details is as shown in figure 6 (a).

E. SOAP Response Message

As a next step Hospital B (Chennai) sends a response with transaction id. This transaction id can be used for further communication. See Figure 6(b).

VI. CONCLUSIONS

Enterprise Application integration among systems available in heterogeneous distributed environment poses a great challenge. Middleware provides critical link between diverse resources and applications that follow standard protocols. Message driven Service Oriented Architecture brings together the dominant concepts of message-driven integration with its attributes and hence a pragmatic approach to solving healthcare systems integration.

The design patterns for Service Oriented Architecture based integration of healthcare enterprise defines the SOAP request-response messages for exchanging patient details etc. across different hospitals to provide the best of InPatient and OutPatient service integration within and across healthcare system.

VII. FUTURE WORKS

In future we wish to implement the message driven integration logic of healthcare enterprise system using Spring AOP. The future work will be manifolds. As a first step, we have to continue designing and analyzing how the spring framework (Spring AOP) encapsulates infrastructural services and applies them to a base application, highlighting the strengths and the problems of server-side component model. Applying the observations obtained in step 1 above to other services. A comparison of the two approaches centralized vs. distributed and experimental findings on how Spring AOP can be a promising approach to eliminate important shortcomings of the centralized healthcare information system and finally applying best practices to build the gap between patient and

doctor and to provide best medical infrastructure to each citizen of the country utilizing the concept of National Healthcare Information System.

```
POST /Patient Details HTTP/1.1
Host: www.IntegratedHealthServices.com
Content-Type: text/xml
Content-Length: 780
SOAPAction:
"http://www.IHS.org/query#GetPatientDetai
ls"
<SOAP-ENV:Envelope
xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/enve
lope/"
SOAP-
ENV:encodingStyle="http://schemas.xmlsoa
p.org/soap/encoding/">
<soap:Header>
<m:Transanction xmlns:m="http://
IntegratedHealthServices.com"
soap:mustUnderstand="1"
xmlns:xsi="http://www.IHSw3.org/XMLSch
ema-instance"
xsi:schemaLocation="http://www.IHS.org/tr
ansaction/">
PatientDetails
</m:Transanction>
</soap:Header>
< SOAP-ENV:Body>
<m:GetPatientDetailsRequest
xmlns:m="http://www.IHS.org/query">
< firstname>Anisha< /firstname>
< lastname>Chaudhary< /lastname>
< /m:GetPatientDetailsRequest>
< /SOAP-ENV:Body>
< /SOAP-ENV:Envelope>
```

Figure 6(a): Structure of SOAP Request Message

```
HTTP/1.1 200 OK
Content-Type: text/xml
Content-Length: 725

< SOAP-ENV:Envelope
xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelo
pe/"
SOAP-
ENV:encodingStyle="http://schemas.xmlsoap.o
rg/soap/encoding/">
<soap:Header>
 </soap:Header>
< SOAP-ENV:Body>
<sm:GetPatientDetailsResponse
xmlns:m="http://www.IHS.org/query">
<m:responseAccepted>true</m:responseAccept
ed>
    <!--transaction id-->
    <m:requestId>1277</m:requestId>
< address>mumbai< /address>
<phoneno>+91-0144-678900</phoneno>
<email> anisha@rediffmail.com</email>
<medicareno>IHS0239</medicareno>
<healthcarecardno>IHS0127</healthcarecardn
o>
<sickness>   cancer </cancer>
< /m:GetPatientDetailsResponse>
< /SOAP-ENV:Body> <
 /SOAP-ENV:Envelope>
```

Figure 6(b): Structure of SOAP Response Message

REFERENCES

[1] JOURNAL ARTICLE: Zachman. J. , A framework for information systems architecture. IBM Systems Journal, 1987:p276-292
[2] W. Gropp, Tutorial on MPI: The Message-Passing Interface, http://www.new-npac.org/projects/cdroms/cewes-1998-05/reports/gropp-mpi-tutorial.pdf, 2009.
[3] David E. Bakken, Richard E. Scantz, Richard D.Tucker. Smart Grid Communications: Qos Stovepipes or Qos Interoperability? http://www.gridwiseac.org/pdfs/forum_papers09/bakken.pdf
[4] SOA in Practice: The Art of Distributed System Design, Nicolai M. Josuttis. O'Reilly , ISBN-10: 0-596-52955-4
[5]Patterns of Enterprise Application Architecture, Martin Fowler, 2002, Addison-Wesley
[6] Krakowiak, Sacha. "What's middleware? ." ObjectWeb.org. http://middleware.objectweb.org/. [7] http://en.wikipedia.org/wiki/Middleware
[8] . Liu Jing Yong, Zhang LiChen, Zhong Yong and Chen Yong. Middleware-based Distributed System Software Process. International Journal of Advanced Science and Technology, Volume 13, 2009.
[9] Enterprise Integration Patterns. Gregor Hohpe and Booby Woolf . ISBN 0321200683 650 pages Addison-Wesley.
[10] Dominique Guinard, Vlad Trifa, Stamatis Karnouskos, Patrik Spiess, Domnic Savio, "Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and
On On-Demand Provisioning of Web Services,"
IEEE Transactions on Services Computing, vol. 3, no. 3, pp. 223-235, July-Sept. 2010, doi:10.1109/TSC.2010.3
[11] M. Baker, B. Carpenter, and A. Shafi, "An Approach to Buffer Management in Java HPC Messaging," Lecture Notes in Computer Science, vol. 3992/2006, pp. 953-960, Springer, May 2006.
[12] usha batra, saurabh mukharjee. "Enterprise Application Integration (Middleware): Integrating stovepipe applications of varied enterprises in distributed middleware with Service Oriented Architecture." .Proceedings of IEEE ICNCS'2010.
of ICNCS'2010.
[13] Lu Liu, Deyu Kong, Yi Li and Zhe Liu. "An Approach to Enterprise Application Integration Based on Ontology Semantic Description" IFIP International Federation for Information Processing, 2008, Volume 255/2008, 977-982, DOI: 10.1007/978-0-387-76312-5_21 http://www.springerlink.com/content/r4822g00427h2381/
[14] http://www.fiorano.com/products/products_application_integration_mission.php
[15] Zhang, C., Gao, D. Jacobsen, H. Generic Middleware substrate through Modelware, Middleware 2005:314-333
[16] Juha Mykkänen, Annamari Riekkinena, Marko Sormunena, Harri Karhunenb and Pertti Laitinenc.
"Designing web services in health information systems: From process to application level"
International Journal of Medical Informatics Volume 76, Issues 2-3, February-March 2007, Pages 89-95
journal homepage: www.intl.elsevierhealth.com/journals/ijmi
[17] Object Management Group (OMG). Unified Modeling Language Specification. Version 2.0, Mar. 2003.
[18] Rambaugh, Jacobson Booch, UML Reference Manual. Addison-Wesley, 1998.
[19] Aimrudee Jongtaveesataporn, Shingo Takada. "Enhancing Enterprise Service Bus Capability for Load Balancing". WSEAS TRANSACTIONS on COMPUTERS, Volume 9, 2010 ISSN: 1109-2750.
[20] FLORIJE ISMAILI, BOGDAN SISEDIEV. "Web Services Research Challenges, Limitations and Opportunities." WSEAS TRANSACTIONS on INFORMATION SCIENCE & APPLICATIONS http://www.wseas.us/e-library/transactions/information/2008/28-359.pdf
[21] ZAIGHAM MAHMOOD. "Service Oriented Architecture: A New Paradigm for Enterprise Application Integration"
Proceedings of the 11th WSEAS International Conference on COMPUTERS, Agios Nikolaos, Crete Island, Greece, July 26-28, 2007
[22] XIN JIN. "Research on the Model of Enterprise Application Integration with Web Services" Proceedings of the 3rd WSEAS International Conference on COMPUTER ENGINEERING and APPLICATIONS (CEA'09)