A Service Modeling Approach to Service Requirements in SOA and Cloud Computing – Using *A u-Healthcare System Case*

Weider D. Yu Binal Joshi Priyanka Chandola

Computer Engineering Department,
San Jose State University,
San Jose (Silicon Valley), CA 95192-0180
EMAIL: Weider.Yu@sjsu.edu

***Abstract*-Service Oriented Architecture (SOA) and Cloud Computing are becoming ubiquitous nowadays. However, applications deployed in the environment are required to meet many time critical as well as functional requirements. In this paper, we aim to provide a service-modeling approach to model the requirement and design of different SOA based services by using Service Oriented Modeling and Architecture (SOMA) and employing Service Oriented Modeling Framework (SOMF) modeling styles and assets [6]. The paper work also includes the development of a u-Healthcare web based software system, which uses the service model as a foundation to ensure secure provisioning of health services to its prospective clients.**

Keywords-Service Oriented Network, Service Oriented Modeling Framework, Service Oriented Modeling Architecture, Cloud Computing

I. Introduction

Cloud Computing has become a common platform for service providers who can offer different kinds of services and, the clients who expect to find most of the services they need in a single place. Though it is very convenient for the clients to use these services, some of the security concerns are to be addressed, such as the integrity of the data that is being transferred, the authenticity, and the confidentiality of the data [7].

The concept of SOA encourages developers to follow design principles which allow applications as domain independent. The security provisioning approach taken (e.g. by IBM) combines Information Technology with business. This is an important concept and one of the ways in which services can be modeled from both perspectives [8].

This modeling paradigm suggests that unlike object oriented modeling in which all the modeling effort goes primarily into the design or architectural artifacts, the focus should be concentrated upon simulating the real world.

II. Background and Definitions

A. Service Oriented Architecture(SOA)

Service Oriented Architecture (SOA) is a business-centric IT architectural approach that supports business integration. A deployed SOA-based architecture provides a loosely-integrated suite of services that are reusable, platform independent and can communicate with each other.

B. Service Oriented Modeling and Architecture

Service Oriented Modeling and Architecture (SOMA) is a lifecycle methodology which serves a service-modeling platform. It consists of the three phases namely, Identification, Specification and Realization

C. Service Oriented Modeling Framework

Service Oriented Modeling Framework (SOMF) is a service oriented modeling language which serves as a tool for designing different service oriented models. Services can be divided into various types like atomic and composite services. A variety of operations like aggregation, decomposition, transformation etc can also be performed on the services to alter their behavior as per different scenarios in order to model optimized and efficient systems.

The increased number of services provided through Cloud has led to lots of competition among the vendors, which in turn impacted the security and privacy of the data handled by the services. Although there are lots of successful security mechanisms out today in the Cloud, the level of security cannot be rated as high [3].

III. System Experiments

This modeling approach emphasizes on putting efforts in the analysis, design, and planning stages of application. Our project provides a platform

978-1-61284-696-5/11/$26.00 ©2011 IEEE

independent model which extends the concepts of secure applications by proposing a service modeling approach for creation of services which aim at providing a secure environment for any application in the Cloud. We have approached the modeling using SOMA lifecycle along with SOMF design notations rather than UML (Unified Modeling Language) as SOMF is a discrete service oriented framework [2].

SOMF design diagrams can be drawn using SOMF Notations and Tool set (shown in Figure 1).

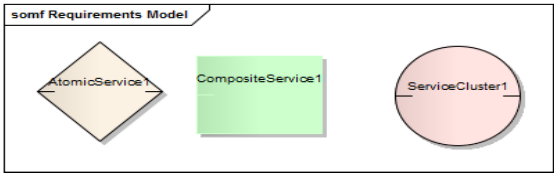

Figure 1. SOMF Notations

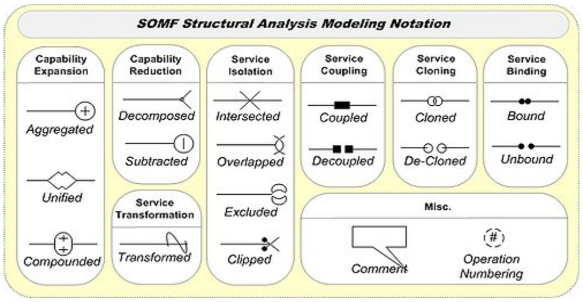

Figure 2. SOMF Tool Set

The Figure 3 demonstrates the user validation performed in u-Healthcare requirements modeling for authentication and authorization [9].

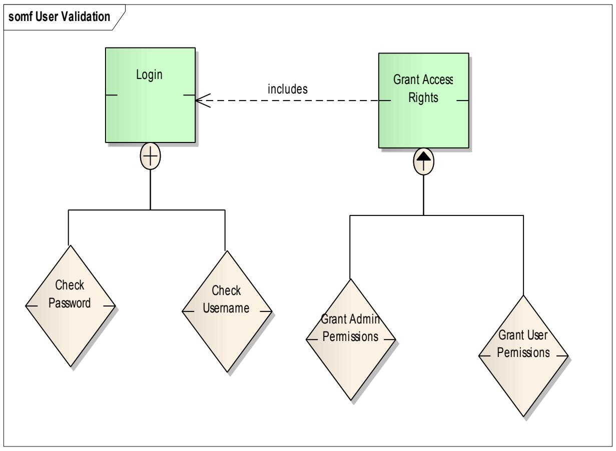

Figure 3. User Validation in u-Health Case

IV. System Design

A. Modeling Architecture

This system architecture is based on SOMA lifecycle model. The main phases in SOMA include Business Modeling and Transformation, Solution Management, Identification, Specification, Realization, Implementation, Deployment, Monitoring and Management. We have mapped the Identification, Specification, Realization phases of this model to the design while defining the architecture of the system. The identification Phase in SOMA, talks about using Goal Service Modeling (GSM) approach for identifying the key goal of the system [1].

The key goal is then decomposed into sub-goals to elaborate on the requirements. We have identified 'security' as the main goal of the system, as security is a key challenge for applications deployed in a SOA and Cloud Computing environment [5].

The services are at the atomic level and serve one key purpose. The Cloud enables the provisioning of applications through compositions of such atomic services. The Specification phase emphasizes on aggregation of Service Components (at high level design).

The Realization Phase of SOMA lifecycle focuses on the technical feasibility of service design and on the selection of appropriate tools and technology [4].

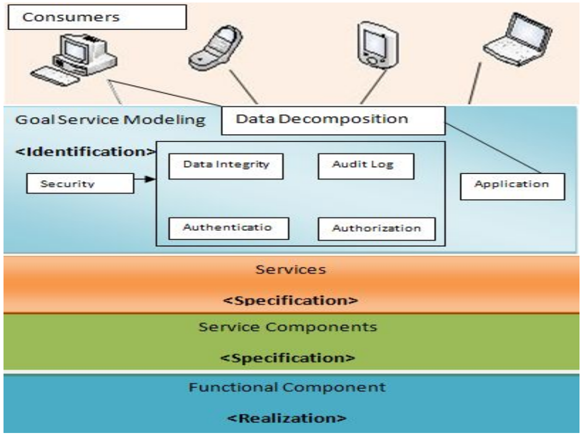

Figure 4.System Architecture Diagram

B. Architecture Subsystem

- *Data Integrity*

Data integrity has to be consistent for all the transactions that are performed between the client and the server. In our approach, the data integrity subsystem comprises of services like encryption, decryption and message authentication code for maintaining data integrity. Depending on the level of security required, there may be additional mechanism to ensure the integrity of data. To enable secure deletion of data, data shredding mechanism can be employed. The number of passes to overwrite the data can be chosen based on the requirement.

- *Audit Log*

Audit log subsystem is used for recording or auditing the activities that are performed on the system. The audit log system compromises of two main services which are auditing the activities of a system and maintaining the audit logs that are generated for future reference. Auditing is performed at all stages of the system.

- *Authentication*

The purpose of this subsystem is to check the identity of the user in order to protect the system from being accessed by unauthorized users. In our system we have used 'check password' and 'check username', which verify the password and the username respectively of the user trying to access the system.

- *Authorization*

This subsystem ensures access control over data by providing proper and controlled access to the user. In our subsystem we have enabled authorization based on the type of user. These user types are part of implementation of this subsystem. Authentication is required before granting authorization to the user.

C. Project Implementation

Our service model is based on the goal of providing secure services on Cloud. We have used Microsoft Windows Azure Platform for building and deploying the application and SQL Azure for database. The entire application has been developed and tested on local development environment before deploying it on Cloud [6]. The database has been built and stored in SQL Azure Cloud. For simulating the exact environment Microsoft Visual Web 2010 express edition and Microsoft SQL server 2008 have been used [1].

The middle tier implementation has been implemented using Visual Web Developer Express 2010. The middle tier includes modules such as home page, account creationg page, login page, physician page, patient page, and payment page. Each of the modules has its own features implemented. For example, the Physician page allows a physician to access features like 'chat with a patient', 'create patient report', 'view patient report', 'modify patient report' and 'diagnose the patient remotely'.

V. Performance and Benchmarks

Service based architecture faces some major inevitable challenges to provide adequate QoS aspects for Web services[5]. One of the key aspects of the modeling process is the identification of key performance indicators (KPIs) to provide an objective and quantitative basis for evaluating the degree to which the goal has been fulfilled. KPIs and metrics identified during this process are used to measure, monitor, and quantify the success of the SOA solution in fulfilling business needs subsequent to the building and deployment of services.

The objective of this model is to provide a framework for building secure services using the SOMA lifecycle. Our methodology includes platform-independent modeling of services. To access the degree of completeness and correctness of this model, we have identified a "**Goal vs Requirement Efficiency**" measure. It captures the comparison between the identified Goal for service creation and the subsequent requirements generated. The key concept behind this measure is to calculate the degree to which the identified security Goals (such as. Authentication, Authorization, Data Integrity and Audit Log) are generic and robust enough to be used in any domain specific application. Based on the percentage of utilization, each requirement set has been graded as 'Poor', 'Satisfactory' and 'Good' with respect to the security goals. The same technique can be used for assessment of other phases of SOMA. We have used the following criteria for defining the threshold of each grade. This is the benchmark for our model:

Goal vs Requirement Efficiency (%)

= (Number of Requirements Fulfilled by Goal) / (Number of Requirements Based on Goal) * 100

Table 1: The Benchmark of the Model

Range	Grade
<50%	Poor
50% to 65%	Satisfactory
>65%	Good

Based on the above criteria and the calculated measures, we applied the benchmark shown in Table 1 to evaluate the u-Healthcare system, and observed that none of the Goal vs Requirement Efficiency measure was below 50%. All of the grades were evaluated as either 'Satisfactory' or 'Good'.

VI. Summary

The main idea behind developing and modeling this project is to introduce more reusability by developing various services that can be platform independent and shared among different parties. The services can be created using any language hosted on a platform.

We are able to interact with various types of services which comprised of web services and local services. Some services are created by external parties while others are self-developed. Since services can be designed in many ways, we have created a set of services that fetch data from the database and some services that perform business logics. There are also components at the presentation layer that utilize the third party services like calendar and map.

VII. Conclusions

SOA offers significant benefits by providing loose coupling, platform neutrality, standards based implementation, and rigid contracts for version independence. We aimed to provide a service-modeling approach to model the requirements and design of different SOA based services by using SOMA and employing SOMF modeling styles and assets.

Since Security and Privacy are no more just limited to IT issues and are rather a business challenge, in order to have more secured and reliable services, security policies must be factored in as a part of the application. Keeping this in mind, our service model aims at reducing the Security and Privacy issues which pose as a serious concern for most users who use Cloud Computing applications or work in a SOA environment [2].

We have demonstrated the way of building a SOA based healthcare application using SOMA architecture and SOMF service modeling language. SOMF can be used for modeling each phase of SOMA.

In SOMA method, we identify the main activities in a broader scope during the initial stage, and further elaborate and refine the scopes in the later stage using SOMF. Thus it fits well with the actual software development environment where requirements change incrementally throughout the development cycle.

References

[1] Buecker, A., Ashley, P., Borrett, M., Lu, M., Muppidi, S., and Readshaw, N. "Understanding SOA Security, Design and Implementation". (2007, November) Retrieved March 24th, 2010, from http://www.redbooks.ibm.com/redbooks/ pdfs/sg247310.pdf

[2] Andrei, T. "Cloud Computing Challenges and Related Security Issues". Retrieved March 2010 from http://www.cs.wustl.edu/~jain/cse57109/ftp/Cloud/ index.html.

[3] Marcia, S. "Security Challenges With Cloud Computing Services". Retrieved March 2010, from http://searchsecurity.techtarget.com/news/ article/0,289142,sid14_gci1368905,00.html.

[4] Arsanjani, A., Ghosh. S., Allam. A., Abdollah. T., &Ganapathy. S., & Holley. K. SOMA: "A method for developing service-oriented solutions". (2008, August 6). Retrieved March 24th, 2010, from http://www.research.ibm.com/journal/sj/473/arsanjani.html.

[5] Axel, B., Paul, A., Martin, B., Ming, L., Sridhar, P. and Neil, R..“Understanding SOA Security Design and Implementation- Design and Implementation” http://www.redbooks.ibm.com/redbooks/pdfs/sg247310.pdf

[6] Azure - Part 1 - Introduction http://geekswithblogs.net/shaunxu/archive/2010/02/24/azure---part-1---introduction.aspx

[7]Yu, W. D.,Jothiram, V. "Security in Wireless Mobile Technology for Healthcare Systems", in Proc. The 9th IEEE International Conference on e-Health Networking, Applications and Services, June 19-22, 2007, Taipei, Taiwan, pp. 308-311.

[8] Brodkins, J.,"Gartner: Seven Cloud-computing Security Risks", www.informworld.com, July 2, 2009.

[9]Yu, W. D.,Ramani, A. "Design and Implementation of A Personal Mobile Medical Assistant", The Journal on Information Technology in Healthcare, Volume 4, Number 2, April 1 2006, pp. 92-102.