1st National Symposium on e-Health and Bioengineering - EHB 2007,
16-17th November, 2007, IAŞI, ROMANIA

Service-Oriented Architecture for Health Systems

Lenuta Alboaie, Sabin Buraga
"A. I. Cuza" University of Iaşi, Faculty of Computer Science;
e-mail: {adria, busaco}@infoiasi.ro

Abstract — Many modern health systems are represented by one ore more Web applications which reside on different sites, possibly using certain incompatible authentication procedures and access policies. Patients and medical staff must perform multiple logins and additional tasks, and they must spend time to search (vital) information located in many places. In this paper, we shall discuss diverse solutions provided by actual paradigms and standards like SOA and Web services to develop a secure and efficient health platform.

Keywords: *e-health, design, platform, SOA, Web services*

1. INTRODUCTION

In this moment, many modern health system architectures are hard to be used in urgency situations by all kind of users: patients, medical staff, and auxiliary personnel. These users want to use provided services, some of them very important – for example, a medical record of a patient in the case of a major accident. For this, the users must search vital and, simultaneously, private information, so they must authenticate on multiple servers to find what they are looking for.

In this paper, we discuss diverse solutions provided by actual paradigms and standards like SOA and Web services [1] to develop a secure and efficient health platform. Also, we provide details regarding certain technologies that make possible to interconnect information from disparate systems, in order to provide secure and efficient access to health information.

When we mention "access", we consider the following points of view:

- Patient's point of view: his/her request for medical appointments (for example, the system can offer the possibility to a complete registration prior an appointment, to save time), pay bills, view and edit account information (maintain personal information);
- Medical staff's point of view: a quick access to similar case treatment(s) and possible recommendation(s), information regarding patients (disease history, regular prescriptions, etc.), information about the relations between patients – e.g., very useful in the case of infectious afflictions.

To develop a complex and useful system that offers services which we previous described, we need to employ paradigms and/or standards such as SOA (Service Oriented Architecture) and Web services.

In the next section, we will describe essential aspects regarding these paradigms and standards.

The paper continues with the general architecture of our proposed platform called Telemon. We will give certain information about the provided services, the user-interaction aspects and the modularity of the system. The paper will conclude with some examples of related health systems and several directions of research.

2. WEB STANDARDS OVERVIEW

The term *Service Oriented Architecture* [1, 10] refers to the design of a distributed system. SOA is not a new technology. It is a novel design methodology and architecture aimed at maximizing the reuse of multiple services (possibly implemented on different platforms and using multiple programming languages).

Reusability is one of the characteristics for a service.

The general architecture for a service is depicted in the Figure 1.

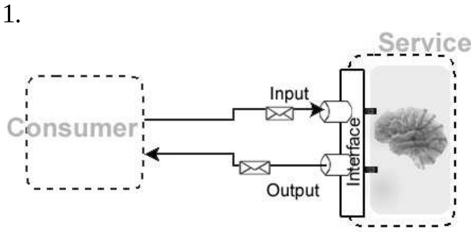

Figure 1. Service architecture

A service is a software component that can be access to provide functionality to the consumer.

The service implementation is hidden behind its interface and is not relevant for the consumer (for example, a component that needs that functionality provided by the service).

These services, their descriptions and their relationships, along with the principles governing its design and evolution form the SOA architecture.

In a SOA platform, the services generally have some important characteristics [10]:

- services are individually useful – they are *autonomous*;
- services must be *loosely coupled*. This term implies that services discover the needed information at the time they need it. The benefits offered by this characteristic are: flexibility, scalability, ability to be easily replaced, and fault tolerance;

- services can be composed to provide other services – *composability*. This promotes the *reuse* of existing functionality;
- services can participate in a workflow. An operation performed by a service will depend on the messages that are sent or received – service *choreography*;
- services can be easily discovered (*discoverability*), eventually in an automatic manner. For that, services must expose details (and additional meta-data) such as their capabilities, interfaces, policies, supported protocols. Other details such as the programming language or the information about the platform are not useful for consumers and – usually – are not revealed.

Figure 2 illustrates the general SOA architecture.

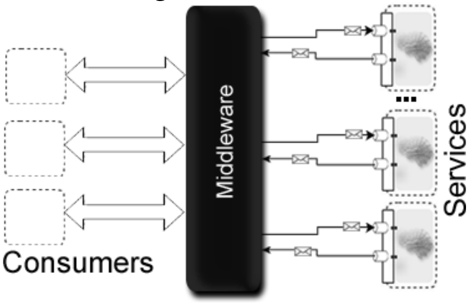

Figure 2. General SOA architecture

Middleware is a key layer in a SOA system, and at this level we can find:

- Interface services;
- Messaging services;
- Management services;
- Mediation services;
- Security services.

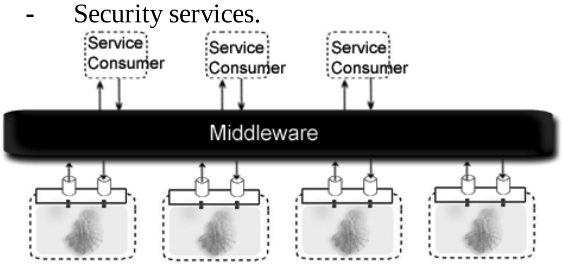

Figure 3. Middleware level

We can remark from Figure 3. how the consumers of the given services and the service providers are "tied" at the middleware level. When some component requests a service, this request is routed via the middleware. The middleware (using the provided services) determines where the service is located and redirects the message to its target.
This is the simple scenario, because the middleware does more than just the basic routing messaging.
Based on the messaging service, the middleware is able to manage different message types. The management service can also monitor performance. The interface service offers support for standards, but it can offer some "adapters" for accessing the service via a non-standard interface.

We can also mention that every SOA platform can implement in its own way the middleware layer, but it is important to respect the principal characteristics of a SOA architecture.
Since now, we described SOA without mentioning the Web services. SOA is an architectural style, while Web services can be viewed as an implementation technology.
The key specifications used by the Web services are:

- XML (eXtensible Markup Language) [4-6] – a meta-language that became an ubiquitous paradigm for (semi)structured information interchange;
- SOAP (originally known as Simple Object Access Protocol) [9, 12] – an XML-based protocol which allows the exchange of structural information (the request and the response of a service) in a distributed and decentralized environment;
- WSDL (Web Services Description Language) [2, 12] – an XML-based language used to describe a Web service (service attributes, service interface and other properties).

The two concepts – SOA and Web services – will be used together in our proposal, but they are not mutually dependent. See [10] for more details.

3. TELEMON HEALTH PLATFORM: GENERAL ARCHITECTURE

In this section, our aim is to demonstrate that SOA and Web services are the right choice for our health platform architecture.

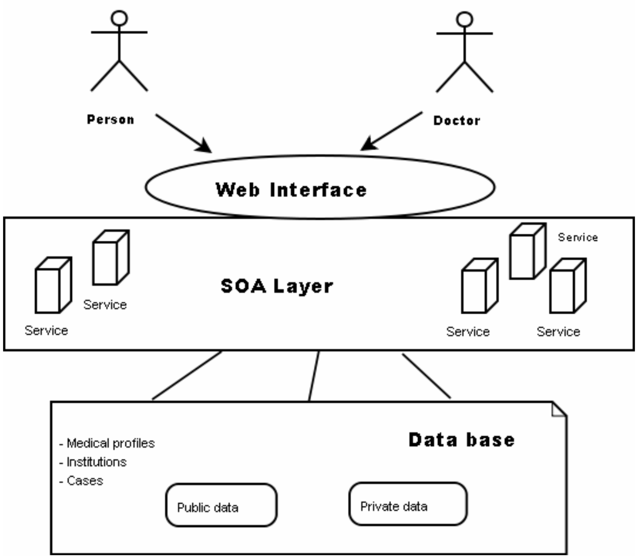

Figure 4. *Telemon* architecture

3.1 Vision

We will start to outline the general structure of a health system – called *Telemon* – that conforms to SOA principles, and for that we will break out the business logic (the processing part) in different pieces – services or components.
The general architecture of our proposed platform is depicted in Figure 4.
The first step is to identify for our platform all units of business logic and functionalities (tasks) to be performed.

Mainly, the system will offer to patients the possibility to access urgent services and to contact directly the medical staff (e.g., family doctor or nurse). Also, from hospitals, policlinics, ambulances medical staff (doctors, nurses) can permanently supervise the patients' status (especially those with special health problems such as chronic disease – e.g., hypertension).

The patient record file will be dynamically supplied via certain proper devices (in some cases, they will be wireless such as mobile phone, PDA etc.). All these actions are possible by using the *Telemon* platform which will use the social Web technologies.

3.2 Provided services

In essence, the system will offer several important Web services for the patients and the medical staff. We will enumerate below all these services and we will identify clearly in the future work the role of each of them:

- creating the users profiles;
- managing the users profiles;
- accessing information about patients (for example, disease history);
- accessing information about similar cases;
- processing (bio)signals;
- producing alarms regarding patients' physical status;
- searching the nearest medical unit in emergency case.

Information related to users (e.g., patients, doctors, medical personal, system administrators) will be stored in a distributed manner by using certain database services.

3.3 Modularization

For our architecture we can imagine that our middleware has tree main modules:

- *service module*: its function is to matching up the requester to provider;
- *workflow module*: its function is to direct the choreography. In other words this module does the coordination task;
- *registry module*: its function is to help finding easily a service when multiple functions exist and in this condition we must have a strict evidence of all available services. Therefore, the registry module maintains information like:
    - Interface descriptions (e.g., WSDL);
    - Meta-data which can represent relations between services, service level agreements.

The aimed architectural solution will be a multi-platform one, loosely coupled, facilitating the integration of applications, services and systems at the Web level.

The system performance will be assured due to the scalability of the architecture based on actual Web standards (XML, SOAP, REST, WS-* etc.) [4, 12].

3.4 User-interaction aspects

From the point of view of the Web interaction with the users, the developed system intends to offer a proper user experience. The performed activities must be effective, efficient and secure, both at the conventional Web browser level and at the level of navigation applications available on the wireless devices.

Also, the provided user-interface will have support for the persons having various disabilities, according to WAI (*Web Accessibility Initiative*) [12].

Another important characteristic will be the support for social networking capabilities, such as managing – for each user – the lists of related contacts (relatives, friends) with similar afflictions, their locations in the proximity of known hospitals, and other addition aspects. Also, the system must be able to inform – in a proper manner – the users about the known pharmacies, clinics, medical offices, etc.

5. CONCLUSIONS

The paper gave an overview regarding the use of the SOA paradigm in the context of collaborative health Web-based system, proposed to be implemented via Web services and related XML technologies.

After, we presented the most important aspects of SOA and correlated standards, we described the *Telemon* platform. Section 3 illustrated the general architecture, essential modules and the provided services of this system. Also, we discussed certain aspects regarding the user-interaction.

Other proposals of complex real-time health systems are RODS [8] and AEGIS [7]. Unfortunately, these systems does not use SOA paradigm. Another interesting SOA-based initiative is *Healthnex* [11].

Further directions of research will include the use of semantic Web technologies [4, 5] to properly model the health-related information (patient profile, medical services etc.). Another aspect will concern the automatic orchestration of the implied *Telemon* services and the management of interoperability – see [10] and [12].

REFERENCES

[1] L. Alboaie, S. Buraga, *Web Services* (in Romanian), Polirom, 2006

[2] D. Booth, C. Liu (eds.), *Web Services Description Language (WSDL) Version 2.0 Part 0: Primer,* W3C Recommendation, 2006: http://www.w3.org/TR/wsdl20-primer

[3] T. Bray et al. (eds.), *Extensible Markup Language (XML) 1.0 (Fourth Edition)*, W3C Recommendation, 2006: http://www.w3.org/TR/XML/

[4] S. Buraga, XML Technologies (in Romanian), Polirom, 2006

[5] M. Daconta, L. Obrst, K. Smith, *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management*, John Wiley & Sons, 2003

[6] B. Daum, U. Merten, *System Architecture with XML*, Elsevier Science, 2003.

[7] B. Reis et al., *AEGIS: A Robust and Scalable Real-time Public Health Surveillance System*, Journal of the American Medical Informatics Association, 14(5), September 2007

[8] F. Tsui et al., *Technical Description of RODS: A Real-time Public Health Surveillance System*, Journal of the American Medical Informatics Association, June 2003: http://www.j-amia.org/cgi/content/abstract/10/5/399

[9] N. Mirtra, Y. Lafon (eds.), *SOAP Version 1.2 Part 0: Primer (Second Edition)*, W3C Recommendation, 2007: http://www.w3.org/TR/soap12-part0/
[10] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall PTR, 2005
[11] * * *, *Healthnex*: http://healthnex.typepad.com/
[12] * * *, *World Wide Consortium*: http://www.w3.org/