Proceedings of ICCCC 2008, Băile Felix Spa-Oradea, Romania
15-17 May 2008

Semantic Integrity Control in the Database Layer of an e-Health System

Lenuta Alboaie Diana Gorea Victor Felea

Abstract

The paper presents a secure and efficient e-Health platform using the actual paradigms and standards like SOA and Web services. We focus on the semantic integrity aspect in the database layer of the proposed system. We also provided an overview of the situations that can affect consistency, as well as the the way we approach each of these.

Keywords: semantic, integrity control, e-Health, design, platform, SOA, Web services, OpenID, decision support

1 Introduction

At this moment, many of the existing e-health system are difficult to use in emergency situations by its users: patients, medical staff, and auxiliary personnel. The main reason is that the architecture of those systems consists of several Web applications residing on different sites, often using certain incompatible authentication procedures and access policies. The patients and medical staff must perform multiple logins and additional tasks, and they must spend time to search (vital) information located in many places. Furthermore there are only a few existing systems that meet the scalability requirements.

In order to develop a complex and useful e-health system we need to employ paradigms and/or standards such as SOA (Service Oriented Architecture) and Web services.

This paper describes focuses on the database layer of our e-Health system, called Telemon [16] , that conforms to the above mentioned architecture. It continues previous work described in [2], in which the general architecture of Telemon is presented. The current work focuses on the semantic integrity control [1] issue in the database layer of the system. This aspect is treated by exploiting the services that Telemon's SOA architecture offers.

2 Functional and architectural perspective of Telemon e-health system

In this section we describe the functionality and the general architecture of Telemon e-Health system.

2.1 Telemon - expected outcome

Telemon's goal is to offer services to both the patients and the doctors. The system will offer to patients the possibility to access emergency services and to contact medical staff directly (e.g., family doctor or nurse). Also, from hospitals, policlinics and ambulances, the medical staff (doctors, nurses) can permanently supervise the patients' status (especially those with special health problems such as chronic diseases - e.g. hypertension).

The patient's record file will be supplied dynamically via certain proper devices (in some cases, they will be wireless such as mobile phones, PDAs etc.).

The system provides a set of Web services for the patients and the medical staff. We list below some of these services: creating users profiles, managing users profiles, accessing information about patients (for example, disease history), accessing information about similar cases; processing (bio)signals, producing alarms regarding the patient's physical status, searching the nearest medical unit in emergency situations, information related to users (e.g., patients, doctors, medical personal, system administrators). All these data is stored in a distributed manner by using database services.

2.2 Telemon Architecture

2.2.1 Overview

Telemon is intended to allow real time patient monitoring, transferring results to local sub-systems, which at their turn will update the central system. This update will be done depending on some factors that we will discuss in next section.

The general architecture of our health system platform is depicted in Figure 1.

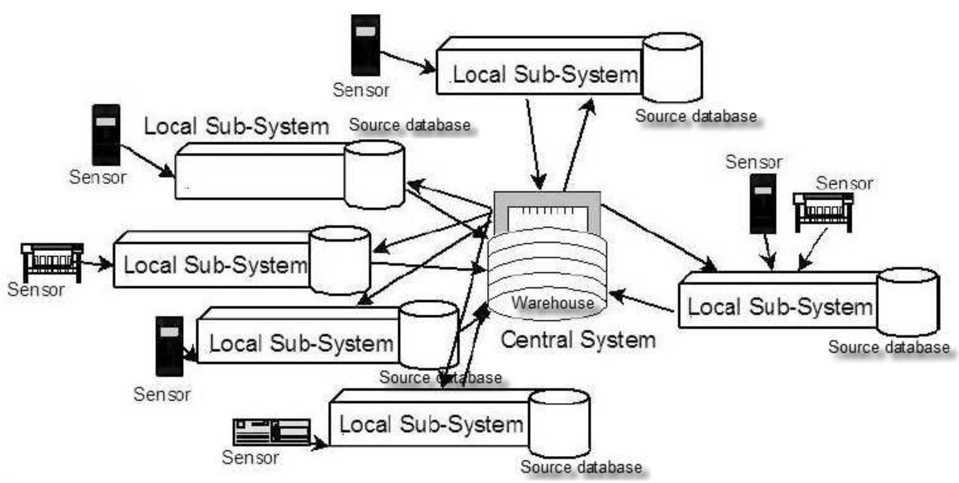

Figure 1: Telemon architecture

The sensors are devices that acquire information (e.g. SpO2 - oxygen saturation, EKG) from patients. They have the ability to transmit information to the proximal local sub-systems. In the same time, the sensors information can be accessed from various mobile devices such as: ambulance terminals, PDAs, laptops etc.

Medical staff can access information from the local sub-system based on an authentication service, as well as complete information in the central system.

In Telemon the information can be easily accessed using an automated authentication and authorization system that allows a user that is authenticated and authorized on a local sub-system, to be automatically authenticated and authorized on the central system. The procedure works the other way around as well, that is, the user that is authenticated in the central system will be automatically authenticated in all the local sub-systems. The authorization to access information is based on user types and user groups (family doctors, specialist doctors, statisticians, researchers, patients). Furthermore, logging out from a local sub-system involves logging out from the central system.

To achieve this we use a single sign-on (SSO) mechanism - which according to [3] is a system whereby a single action of user authentication and authorization can permit a user access to all computers and systems where he has access permission to, without the need to enter multiple passwords. Single sign-on reduces human error, a major component of systems failure and is therefore highly desirable for this kind of e-Health systems. In the future work we shall propose an OpenID module (which is a decentralized single sign-on system) [4] that will ensure the whole authentication process used in Telemon.

Ensuring a right authentication and authorization module contributes significantly to the maintenace of the database integrity.

2.2.2 Architectural view on the components

We outline the general structure of the components - local subsystems and the central system, both of which conforming to SOA principles - and for each of them we decompose the architecture in several levels.

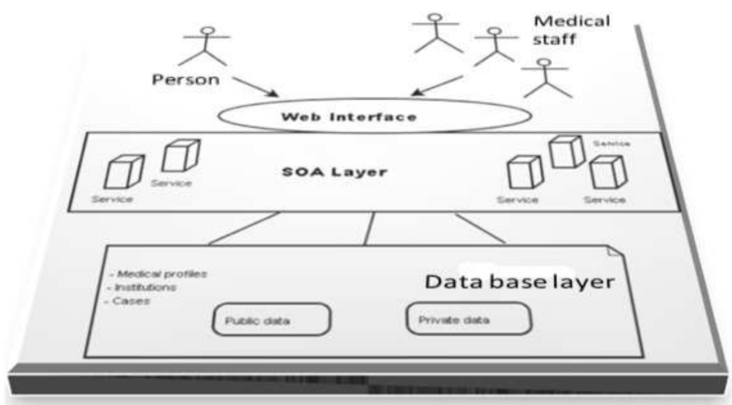

Figure 2: A sliced overview of a component

Each of the components consists of the following layers:

User-interaction Layer - from a user interaction perspective, our system is enriched with accessibility capabilities. The provided user-interface supports users having various disabilities, according to WAI (Web Accessibility Initiative) [15].

The performed activities must be effective, efficient and secure, both at the conventional Web browser level and at the mobile application level.

An important feature will be the support offered by the GIS [5] web services, which will offer to authenticated patients information about the known pharmacies, clinics, medical offices in their proximity.

Telemon core level - from the technical point of view the system conforms to the SOA paradigm. The term Service Oriented Architecture [6], [7] 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).

Telemon core level consists of three main modules:

- Service module: its function is matching up the requester to the provider (e.g. a doctor monitors the patient condition and wants to access his medical record. For that, he accesses a service called GetPatientRecord. The request is delivered to the Middleware, where the service module matches it to the corresponding service provider) ;

- Workflow module: its function is to do the choreography. In other words, this module does the coordination task;

- Registry module: its role is to easily identify the most appropriate service among all the existing services that provide the same function. Therefore we must have a strict evidence of the available services. The registry module can maintain information like:

  Interface descriptions (e.g., WSDL) [13], meta-data that can represent relations between services, service level agreements.

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

The Database level. Telemon comprises two types of databases. The one located at the sub-system level is an operational data base that records data sent by the sensors (they are named *source databases*). The other is a *warehouse* and is located at the central system level.

Data warehousing refers to a set of technologies for improving the decision making processes. In this respect, Telemon includes a Decision Support Module (DSM), which will use a system named DeVisa, described in [10]. The system stores and processes prediction models using XML technologies [9] and is intended to provide feasible solutions for dynamic knowledge and decision support integration into applications. A use case would be when a doctor needs assistance in taking a decision about a treatment for an emergency situation. DeVisa provides prediction services that will be integrated as a module in Telemon, due to the chosen SOA architecture.

The core functionality of DeVisa is available via web services. The prediction models stored in DeVisa are built on top of the central warehouse, and uploaded in DeVisa via web services.

3 Database integrity

In this section we describe the semantic integrity control in Telemon. Semantic integrity [1] control(SIC) is a technique meant to ensure database consistency. Other techniques that assures database consistency are: concurrency control, reliability and protection.

We can maintain the database consistency by enforcing a set of constraints. There are two kinds of constraints:

- structural constraints (e.g., unique key constraint in the relational model)
- behavioral constraints

The structural constraints can be expressed using compiled individual assertions of relational calculus, as depicted in [1]. The behavioral constraints typically involve data from multiple sites (source and/or central databases) and are expressed using predicates as described in the following sections.

There are two main cases to consider when tackling the consistency issues in Telemon: the warehouse consistency and the source database consistency. In general the consistency in databases may be affected after one ore more updates take place. In section 3.1 we describe the possible types of updates that may happen in Telemon and their consequences. In section 3.2 and 3.3 we will focus on a set of assertions representing mainly behavioral constraints that contribute to enforcing Telemon's database layer integrity.

3.1 Update scenarios

This section presents the context and the scenarios that can affect the consistency of the database. As we stated in 2.2.2 the database level of Telemon is formed by two types of databases: source databases and central system warehouse.

The updates for the source databases originate from two sources:

- continuous data sent by the sensors
- the data that are operated into the system by the medical staff

The source databases feed the warehouse with data. Our system, partially decentralized, ensures the data transfer from the local sub-systems to the central system.

To optimize the data traffic we designed two different types of warehouse updates: instant updates and periodical updates. Periodical updates are done on a regular basis and they apply to all the data. The instant updates depend on an emergency threshold, which refers to the severity of the patient's condition. In the cases where the severity is above the emergency threshold, an instant update is applied in the warehouse. This is crucial when an emergency case occurs and the doctor can consult the specialists that are logged in the central system at that time. The specialists logged into the central system are automatically logged on to the local sub-system (using the openID module). In this way they can interfere in the patient treatment directly in the local sub-system in whose proximity the patient is situated. The information concerning the treatment syncs back with the central system via the periodical or instant updates. Furthermore, when there is a serious case of a similar type in another sub-system, the doctor who is logged on there consults the warehouse to get the previous treatment in order to take a decision.

3.2 Source database consistency

This section discusses the situation in which the consistecy of the source database can be affected and we establish the assertions to enforce integrity.

In Telemon the access to a patient record can be granted to more than one doctor. The interesting situation is when two or more doctors write observations or treatments in the patient record in the same time. In this case we resolve the concurrency control through the use of a priority mechanism.

In this sense, we established a priority access (for the write action) to the patient records. The doctor that is conducting the consult with a specific patient has the highest priority, so he writes directly in the patient's record, after the record is locked. Should another doctor need to write something in the patient's record in the same time, he actually writes in a temporary buffer. After he writes down all the observations and the patient's record is not locked anymore, the information is dynamically added. In addition to the priority mechanism described above, we consider that updates to the patient record are done conforming to a time-stamp, which represents the moment in time when each of the doctors accessed the patient record.

In the above scenario we stated that the mechanism that formalizes the access priority to the patient's record is based on behavioral constraints. In addition there are structural constraints associated with the relational database model used by the local sub-systems.

3.3 Warehouse consistency

The source database semantic integrity is solved by means of concurrency control mechanisms. In the case of the warehouse there are a few possible scenarios that can affect the consistency and for each of them we will identify the assertions that form the behavioral constraints.

In the regular case of periodical updates it is not necessary to specify Telemon behavioral constraints, because the updates take place according to the general integrity control principles in a database.

Therefore we focus only on the case in which the warehouse is updated from two or more source databases. One possible use case is when a patient migrates in another area than his/her usual sub-system area. Let's assume that patient A is registered in the proximity of the sub-system S1 and the patient travels to the proximity of another sub-system: S2. Thus the S2 sensor will send the corresponding data to the S2 sub-system from the moment it has detected the patient. Furthermore, S2 will trigger a workflow that comprises: a dynamic service from Telemon core level (see [2]) which has the ability to find the base subsystem of the patient - which is S1 in this case and notifies S1 of the fact that the patient is in S2's proximity.

In this circumstance we distinguish two cases:

- During his/her stay in the proximity of S2 the patient does not need emergency medical assistance. Nevertheless Telemon is in continuous surveillance of the patient. After storing the information received from the sensor S2 invokes an update service that has as input the reference to S1 (obtained by the previous service) and transfers the patient's data to S1. Hereby we identify a behavioral constraint that states that while A is in S2's area, S2 stores his records and afterwards it enforces the update between the S2 and S1. This constraint avoids unnecessary updates of the central system, because the periodical updates (see 3.1) are done by S1 after it receives the data.

- During his/her stay in the proximity of S2 the patient needs medical assistance. When S2 perceives emergency information from the sensor it triggers an emergency update procedure that updates the patient data in his/her base sub-system (S1). The doctor that assists the patient logs in S2 and he is automatically logged in S1 (OpenID) and can interact with the A's personal doctor from S1 in order to prescribe the most appropriate treatment given the circumstances. In this case a set of assertions that state that if S2 detects an emergency data it triggers an emergency update to S1.

The assertions form the base of the database consistency control and in conjunction with the authentication and authorization module we achieve the Telemon semantic integrity control.

4 Conclusions

This paper presented the semantic integrity aspect in an e-Health system that conforms to Service Oriented Architecture. The system contains component such as: sensors, local sub-systems and central system. The sensors acquire information from patients and send it to the proximal local sub-systems, which update the central system. The central system stores the data in a warehouse and provides services such as: statistics, decision support, search etc. The security aspects are solved by using a OpenID mechanism, which will be detailed in our future work. We provided an overview of the situations that can affect Telemon's database consistency and the way we approached each of the cases.

References

[1] M. Tamer, P. Valduriez, *Distributed Database Systems*, Prentice Hall, 1999.

[2] L. Alboaie, S. Buraga, "Service-oriented architecture for health systems," *The Proceedings of the national symposium with international participation e-Health and bioengineering- EHB2007,* Nr.2, 2007

[3] *Single Sign-On* , http://www.opengroup.org/security/sso/

[4] *OpenID*, http://openid.net/foundation/

[5] *Guide to Geographic Information Systems* , http://www.gis.com/index.html

[6] L. Alboaie, S. Buraga, *Web Services*, Polirom, 2006

[7] T. Erl, *Service-Oriented Architecture: Concepts, Technology, and Design,* Prentice Hall, 2005

[8] S. Buraga, *XML Technologies*, Polirom, 2006

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

[10] Gorea, D. (2007), "Towards storing and interchanging data mining models," *Proceedings of the 3rd Balkan Conference in Informatics,* volume 2, pages 229-236.

[11] *ARTEMIS,* http://www.srdc.metu.edu.tr/webpage/projects/artemis/publications.html

[12] *semanticHealth,* http://www.semantichealth.org/

[13] *WSDL - Web Services Description Language*, http://www.w3.org/TR/wsdl

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

[15] * * *, *World Wide Consortium*, http://www.w3.org/

[16] * * *, *Telemon Project*, http://thor.info.uaic.ro/ ˜telemonfcs/

Lenuta Alboaie
University Al. I. Cuza Iasi
Faculty of Computer Science
str. G-ral Berthelot nr 16
E-mail: adria@infoiasi.ro
Diana Gorea
University Al. I. Cuza Iasi
Faculty of Computer Science
str. G-ral Berthelot nr 16
E-mail: dgorea@infoiasi.ro
Victor Felea
University Al. I. Cuza Iasi
Faculty of Computer Science
str. G-ral Berthelot nr 16
E-mail: felea@infoiasi.ro
Received: December 31, 2007