DA SILVEIRA, Marcos; GUELFI, Nicolas (2009). « The concept of a platform for remote monitoring patient's health state ». Published in Med-e-Tel 2009, The International Educational and Networking Forum for eHealth, Telemedicine and Health ICT, 1-3 April.

THE CONCEPT OF A PLATFORM FOR REMOTE MONITORING PATIENT'S HEALTH STATE

M. Da Silveira[1], N. Guelfi[2]
[1]Public Research Center Henri Tudor, SANTEC, marcos.dasilveira@tudor.lu
2A, rue Kalchesbrück, L-1852, Luxembourg, LU
[2]University of Luxembourg, LASSY, Luxembourg, nicolas.guelfi@uni.lu
6, rue Richard Coudenhove-Kalergi L-1359 Luxembourg

Abstract: **This paper presents a concept of platform for developing medical web-services. The goal is to improve the interoperability of eHealth systems in Luxemburg. The idea is to define different levels of designing and propose interfaces to connect. The architecture of the platform is based on SOA (Service Oriented Architecture) concepts and adapted to e-Health needs.**

Introduction

In a digital society, the concept of services has changing and words like e-service and web-service enforce their place in the market. In this new context, RESIST project contributes to transform the current medical system into a more adapted *e-service based system* supported by technologies from the ICT domain. The main goals of this project are:

(a) The identification of needs from medical sector in order to implement home monitoring services;

(b) The selection of a set of technologies that can be used to implement these services;

(c) The definition of an architecture that promote interoperability between medical applications;

(d) The implementation of a case study that validate the proposed concepts.

This paper presents the last three goals, the first goal was presented in [1]. The definition of the platform had to take into account the existing infrastructure of Luxembourg. HealthNet is an example of this infrastructure. It is a secure telematic network for health care professionals in the Grand-Duchy of Luxembourg. All hospitals and the majority of Laboratories are connected via HealthNet and they can safely exchange medical data. It is also opened to healthcare professional and the number of adherents is increasing.

This scenario has driven RESIST team to propose different solutions according to the category of potential users of the platform. For remote monitoring purposes, the users were classified into: Patients, Healthcare Providers and System Managers. Each category of user will have an adapted system to connect to (and exchange data within) HealthNet:

(1) Patients will use the *home monitoring system* (MSY). It is composed of a set of sensors and a central unit (e.g., PC, PDA, etc.) that collect patients vital signs, store them (locally or not) and send them to EHR (Electronic Health Record) servers. We adopt the hypothesis that EHR servers are managed by hospitals and distributed over the country;

(2) System Managers will use the *monitoring center system* (HMC), normally placed in hospitals. It has the role of mediate communications between the other users of the platform. It can also manage the access writes of users, store and exchange medical data, publish web-services (e.g., UDDI registry), configure and make the maintenance of equipments, manage urgent alarms, etc;

(3) Healthcare providers will use the *medical terminals.* They are access points for medical data and can have specialized applications that support their activities. For example, it can have macros in Excel® that read EHRs from the HMC database and present the patient's health state or it can be part of a more complex application (e.g., GECAMED® if the user is in a private cabinet or a HIS if he is using the hospital application) to access and manage the patient's EHR.

Combining these three sub-systems to compose a Remote Monitoring System is a hard challenge. It requires an interdisciplinary knowledge as well as, Interoperability, Security, Legislation, Web Standards, Medical Standards, etc. Without an appropriated coordination, the relations between users, services providers, public administrations (e.g., Social Security, certification authority, etc.) can quickly become very complex and drive the system to chaos.

RESIST contributes to reduce this complexity acting over two weak points: the coupling of functionalities of the platform and the designing process.

In the first case, the functionalities of the platform will be performed by small pieces of softwares with relative autonomy that cooperate to perform more complex activities. In other words, the platform is base on SOA (Service-Oriented Architecture) concepts and each sub-system is composed of a set of web-services. The communication between web-services is done by messages exchange and web-service technology is well known by their loose coupling properties and the facility of maintenance.

In the second case, RESIST proposes to split up the designing process into 5 different levels where each level requires a sub-set of knowledge. The levels were grouped into 2 layers: one for business designing and the other for services designing.

Separating both layers gives more flexibility and dynamism to the system. In the upper layer, business process experts can easier specify rules and constraints or define new business combining services throughout a standardized language, like BPEL. In the lower layer, experts on specific domains can improve the quality of the system by updating, deleting or added new services. This lower layer is used, for example, by computer scientists to design (and implement) web-services. Each new service is published in an internal registry and used by the upper layer. A BPEL engine is used to connect the two layers. It can dynamically choose the web services available in the internal registry and use them to execute a set of specific tasks requested by the business. As shown in Fig. 1.

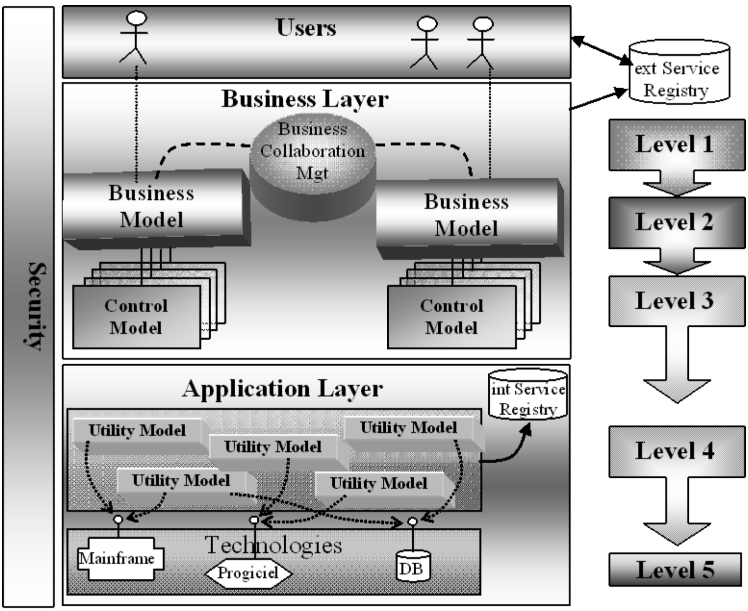

Fig. 1: RESIST conception levels for services designing

A practical example of this strategy can be seen when a healthcare professional decides to define a clinical treatment. If all pieces of the treatment exist already (as web-services) then the healthcare professional only needs to select them and define the rules to perform it (association, sequence, timing, etc.). However, if new functionalities are necessary, then he/she can request it by defining the properties of the functionalities and send this information to the lower level. In a second phase, computer scientists use this information to design and implement the web-service.

The implemented case study is composed of 2 sub-systems and 9 web-services (see Fig. 2). The patient has at home the MSY sub-system and can measure the vital signs throughout the patient application. This application has an adapted interface (according to the disease or the age of the patient) and executes a BPEL code. Each measurement is done by the Local Measurement Service and stored by the Storage Service. The synchronization of local and remote data follows the set up of the system (continuously, periodically, event-driven). The remote database keeps the history of the patient while the local database keeps only recent information. All messages are encrypted and the identity of the users is checked every time that the connection is requested (a watchdog limit the connection time), for security, a session key is generated after the connection. Thus, a national certificate authority is mandatory (represented here by the Identification Service). In the hospital, the HMC sub-system will receive the patient request (Measurement service or Configuration service), check the identity and, if successful, perform the request. 3 types of requests were implemented: Configuration update, Data Storage and Alarm.

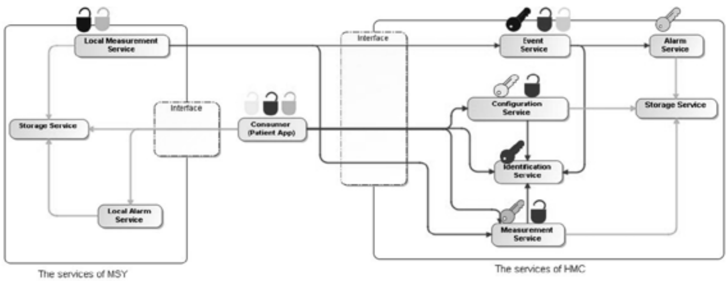

Fig. 2: RESIST case study: Remote Monitoring System

References

[1] M. Da Silveira, and N. Guelfi. “RESIST: A Platform to Support Medical Web-Services: A Case Study”, Global Telemedicine and eHealth Updates: Knowledge Resources, Vol. 1, 2008, pp. 107-111