

# Communication Architecture for Tracking and Interoperable Services at Hospitals: A Real Deployment Experience

Augusto Morales, Tomás Robles, Ramón Alcarria, and David Alonso

Technical University of Madrid, Avenida Complutense. 30,  
28040 Madrid, Spain

{amorales,trobles,rалcarria}@dit.upm.es  
david.alonso.abarca@alumnos.upm.es

**Abstract.** Any new hospital communication architecture has to support existing services, but at the same time new added features should not affect normal tasks. This article deals with issues regarding old and new systems' interoperability, as well as the effect the human factor has in a deployed architecture. It also presents valuable information, which is a product of a real scenario. Tracking services are also tested in order to monitor and administer several medical resources.

**Keywords:** RFID, OSGi, SIP, Tracking, real-deployment.

## 1 Introduction

Medical environments are dynamic and this affects the way standard processes are accomplished. In recent years, new ideas related to drug tracking, smart medical services, and patient monitoring have been transformed into successful systems. In addition, there are technologies such as: Radio Frequency Identification (RFID)[1], Open Service Gateway Initiative (OSGi)[2], Session Initiation Protocol (SIP)[3] that have certainly demonstrated [4][5][6] through different mechanisms that they can enrich medical environments by extending the capabilities to many devices. As a result, there are new trends for applying alternative technologies in order to address hospital communication issues to support future medical services.

Medical environments have different requirements that restrict the way new technology, architecture, protocol or system can be integrated into an existing communication model. Therefore, there are some problems regarding interaction between the medical staff and the technology itself. This interdisciplinary fact affects the way AAL is provided to patients or handicapped people whom need support in their own house or in hospitals.

This paper is based on a previously proposed architecture [7], where we considered some problems regarding sensors and medical services as well as their integration with legacy hospital networks and processes. This architecture has an open-service model based on OSGi, so new medical services can be deployed without modifying other running applications. In this way, the architecture could be seen as the core of future characteristics provided by an AAL platform.

This paper is focused on the deployment issues the CARDINEA project [8] consortium had to overcome in order to integrate existing medical systems with new features, sensors and services. It also proposes alternatives to address hospital requirements such as: drugs, equipment and medical staff tracking. We describe our fully-operational prototype which carried out real medical data and processes. On the other hand, there are several alternatives for analyzing valid system integrations; though we limit our scope to the deployment itself and the challenges we have to deal to, such as: physical location, resource availability and staff cooperation.

## 1.1 Challenges

Nowadays medical environments have challenges in several areas. One of them is tracking. If we analyze specific scenarios, for example one floor specialized in intern medicine; we conclude that tracking can be applied to many things such as: expensive drugs, medical equipment, nursing and so on.

Actually, some hospitals keep manual tracking of the drugs, since they are received until they are administered to patients. This manual process takes time and is susceptible to human error [10]. Thus, drug tracking has a direct impact on the overall efficiency of a hospital's pharmacy, as well as it also prevents to lose costly or limited drugs.

**Tracking of medical devices** involves an important point of the whole hospital process. In addition, it inherits the same deficiencies, if manual processing is in operation. Expensive devices can be lost, stolen or be inadequately used. Nevertheless, automatic tracking overcomes them, as well as the hospital can optimize its resources when unforeseen events occur.

**Staff tracking** is particularly challenging because it involves humans. It is necessary in critical cases when emergencies occur or when patients need continuous monitoring. It has been proven that medical errors can be minimized [11] by using communication systems, so medical staff monitoring can help to administer drugs in the correct way, amount and time. On the other hand, medical staff tracking is difficult because it involves other human aspects such as: privacy and trust.

Another challenge is concerning patients and their **vital constant monitoring**. Hospitals have systems in order to monitor and send alarms to medical staff when special situations occur. Otherwise, some of those systems are proprietary, so the integration among Hospital Information Systems (HIS) may be complicated. Adding the new features cannot modify or affect existing medical services, process, systems or data.

It is not easy to integrate a single hospital scenario with current and new systems. Integration of technologies is indeed challenging. It includes networks, bio-sensors, medical devices, and specialized database such as HIS. However, another challenge concerns **humans and their willingness** to use new processes, or systems. Thus, integration with health personnel and technical supporting people requires training, as well as direct feedback from them, since there are the people that will adopt the technology. Furthermore, on-site testing is crucial for any successful system and the perception medical staff will experience.

Since our architecture is based on **integration between different medical systems**, and involves human interaction, we have implemented the prototype in the

Hospital de Terrassa, Catalunya, Spain. This functional prototype has been integrated with existing hospital systems, so real medical staff has interacted with it.

In the next section we briefly explain the top level architecture for the prototype. Following that we describe the deployment process, requirements and other considerations for the testing scenario. The next section describes results and problems addressed, and finally we finish with a discussion and lessons learned.

## 2 Architecture

The architecture proposed in the CARDINEA project integrates different modules and technologies. The architecture is based on hierarchical levels and supports different logical or physical areas.

Figure number 1 shows the modules developed and how all of them are integrated in the same architecture with OSGi as the main platform. The modules involved [7] are Open Context Platform (OCP) [9], Enterprise Service Beans (ESB), SIP, Mirth and RFID. The OCP module stores information regarding sensors, patients, staff, drugs, as well as the location data of them. Next, it organizes these data into a defined ontology. The ESB module registers the information received from the Hospital Information System (HIS) into the OCP context. This information is used to create the different entities needed in the ontology for storing all the biomedical data for further analyzing. The SIP module negotiates the communication channels taking into consideration the biomedical information and the current network conditions. The RFID module controls the antennas and RFID readers intended for the detection of the location. Finally, the Mirth module receives parameters from the SIP session and sends/receives the Health Level 7 (HL7) [12] compliant data by using the proper quality of service values.

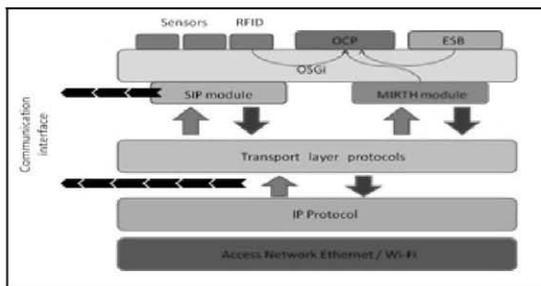


Fig. 1. Central Server Topology Layers

## 3 From Architecture to Implementation

In any real deployment, the architecture may be limited by the existing physical and technological resources of the scenario. On the other hand, the architecture has to be adaptable without modifying its high level functions and purpose. As we mentioned before, our prototype was test-driven in a real hospital scenario which had many constrains.

The physical element affects many technological factors such as: the antenna power for reading RFID tags, the available Bluetooth range and Wi-Fi availability. In this case, a wall's thickness and its material can affect the overall system performance. Alternatively, wired connectivity can be used to ensure good quality network communication, but it can be very invasive for testing a temporary system. Furthermore, a successful architecture deployment needs a balance between functionality, and non-invasive techniques. This last characteristic is undeniably important for medical and other AAL environments, which cannot be frequently modified.

In our deployment there were several wireless devices such as: Health Station, RFID readers and sensors that made use of wireless connectivity. It is clear that these devices are designed to avoid interferences and comply with many safety regulations. Despite this, we had to deal with some packet lost, Bluetooth interference, and in some cases, medical staff complaints. Thus, using wireless devices in medical environments have clear advantages and disadvantages.

The architecture defined several hierarchical levels with the aim of maintaining the separation between different hospital floors. This design allows continuing the communication system running even if one of the network devices fails. The physical and technological resources available in the Hospital did not permit the deployment of the complete architecture based on different roles and servers. As a solution, we unified the Intermediate Server with the Central Server into one single component. This change was suitable because the logical modules are OSGi bundles and can be seamlessly executed in the same server. In this case, the architecture's design based on the OSGi's flexibility allowed us an adaptable and fast deployment.

The prototype was deployed by implementing different software elements of the architecture. Each software element, and its functionality, was executed by an OSGi bundle. As we previously mentioned, the OSGi platform allows a quick integration. Even though, in a real deployment which involves several developers and software components, a successful system requires time indeed. Thus, even when OSGi has methods for publishing bundles and their functionalities, via standardized interfaces, data format and data exchange are still issues in real scenarios.

### **3.1 Deployed Architecture and Workflow**

Finally the architecture has suffered some changes due to the fact that it had to be implemented in a real test scenario. The scenario featured some installation restrictions that were motivated by the placement of equipment and the possibility of disturbing the patients and the health personnel.

On the other hand, we had to make topology changes because of the available hospital resources. Figure 2 shows the topology of the test scenario. In the first step, the health stations collect (1) all the medical information from the biomedical sensors. Next, they send (2) the medical data to the HIS, which is a legacy communication system used by the hospital and stores all the patient records. This HIS also executes data coding in HL7 format, establishes the correct data session and send (3) HL7 data to the Central server. Meanwhile, the RFID sub-system, which is totally independent of other modules, detects the RFID tags from nurses, drugs, patients and sensors, so it can send (4) these events to the Central Server.

The test scenario achieves external communication with the HIS using a 3G router. By doing this, it did not interfere with the other systems installed and complied with hospital's regulations. All the information collected in the Central Server is processed and organized in a previously defined ontology by the OCP module, which also generates a context model. Therefore, any other module in the OSGi platform could later consume this information.

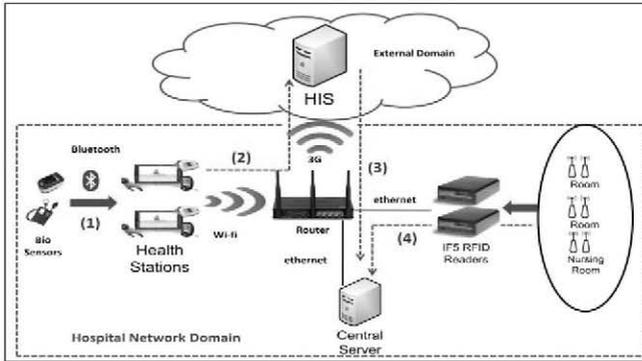


Fig. 2. Deployed system and workflow

### 3.2 Hospital Deployment

The prototype has been test-driven using the Open Source FUSE ESB [13] as the OSGi Framework for the Central server. It also ran Ubuntu 10.04 with Java SDK 1.6 and 1GB RAM. The Health Stations ran Windows XP Embedded version with 1GB RAM. The RFID readers were the Intermec IF5 [14] model, running an embedded version of Linux OS and JADE [15] 3.7.0 version as the OSGi framework.



Fig. 3. Patient Room



Fig. 4. Nurse station



Fig. 5. RFID Tags

As we previously stated, the whole prototype was located in the Terrassa Hospital's fifth floor. This floor specializes in intern medicine. There were three places with RFID passage arches: two patient rooms and one nursing room. Each arch had two antennas to guarantee the reading of RFID tags. The RFID readers, which processed tags and managed antenna reception power, were located on the top of the ceiling. There was

Ethernet communication from the readers to the Central server, as well as Wi-Fi link as a backup up channel. The Central server was placed in the nurse station.

The Health Station, depicted in figure 6, was attached to the nursing trolley, as well as the glucometer, pulsimeter and tensiometer sensors. These devices sent their monitored vital constants to the Health Station, via Bluetooth connection. Then, the Health Station sent these data to the HIS using its Wi-Fi connection.



**Fig. 6.** Health Station and nursing trolley

The nursing trolley had a RFID tag which allowed the system to detect the time when vital constants were registered for data processing. The nurses also carried the same sort of RFID tag in their pockets for a similar purpose. Location was also recorded by monitoring patients and nurses RFID tags. At the beginning of the testing-time, the system noticed a low rate of RFID tag events because of the hospital's physical characteristics, including the metallic nursing trolley employed. As a solution the RFID working frequency was adjusted between 865 and 868 MHz and the reader output power was also set to the maximum available rate (30.0 dBm). These changes significantly improved the system performance. With the proper tuning, the tracking system preliminary results showed that during nine testing days the activity average rate for nurses and trolleys was 12.66 RFID location events per day. The nurse station had 8.77 events and the rooms 3.89. These first results matched with the periodic nurses rounds in the hospital. Gathered data analyzing time has not finished yet. However, at this time, we did extract useful information from health personnel surveys. The system did not affect regular medical task, so medical staff could perfectly continue working with patients which was especially important for the prototype success.

## 4 Discussion

Any evolution from current to future stages in any hospital systems is difficult and risky, especially when it involves a drastic change. Taking this into consideration, we have realized that experimenting with real systems, process and humans is necessary for successful hospital architectures and general AAL environments.

There are many issues regarding deploying new medical systems into real scenarios. One of them is *interoperability of systems*. Our OSGi based architecture was mainly focused on this fact. Nevertheless, for real working scenarios, it is especially difficult to troubleshoot problems when different software components have to interoperate. In addition, this issue can be even worse when remote access support is not enough, because on-site testing can be difficult and sometimes frustrating for

patients, medical staff and so on. Therefore, previous and extensive system testing is a mandatory requirement for any medical deployment,

*Cooperation and feedback* with medical staff and patients are indispensable. Our architecture involved a human component by suggesting to medical staff to always carry a RFID tag in their pockets. In this way, this new sort of request has to be adapted to their comfort and cannot affect their daily tasks. Thus, adding touch screen technologies seems to be less invasive, improves the system usability, and the willingness to continue testing new systems.

Since a prototype cannot affect a Hospital, its normal processes and communication systems, *efficient physical deploying* is a requirement. Our prototype involved bothersome infrastructural changes for the RFID equipment installation. However, once the antennas were in their position, our architecture allowed us to change settings such as: antenna power, antennas covering and so on, in order to adapt the prototype. As a result, *software flexibility* allowed us to avoid physical alterations. Evidently *Privacy* was also taken into consideration. Our experiences in this development, via surveys, suggest to us that tracking, which involves privacy aspects, is not totally-accepted by health personnel. As a result, it was necessary to correctly explain and define the purpose of the collected data. Thus, in our testing system the location information was only employed to obtain statistical data.

#### **4.1 Lessons Learned**

As we expected, the testing scenario had undeniably provided us valuable information. Our data analysis, which will provide us with statistical results, has not been finalized yet, but we have learned lessons regarding this architecture and its capabilities. Taking all the deployment challenges into consideration, we first requested, to the Hospital's administration, two sorts of phases for our testing scenario. The first phase was focused on system tuning, and the second phase for data collecting. However, our short period of two week was not enough because the testing phase was also limited by human factors such as medical staff and patients' availability. Therefore, these types of random factors have to be taken in account when testing time is restricted to the internal hospital policy.

The original architecture envisaged service creation from specialized developers. On the other hand, the medical staff's surveys demonstrated that creating basic and simple services will be one of the best advantages for future hospitals and AAL environments. Thus, personal devices such as: mobile phones or PDA could have mechanisms for creating medical services that interact with HIS or similar hospital systems. In addition, staff location and alarms could be integrated as key service features. Since medical staff's tracking is still an uncompleted feature in our system, a future architecture will have to include location policy modules.

Our prototype made use of several wireless technologies for communicating devices. However, for future deployments it would be positive to support other technologies such as 6LowPAN or Zigbee. This attribute, in conjunction with Wi-Fi, will extend the monitoring capabilities and ensure the control of data flow in case of failure or unforeseen situations.

Alternatively, the participation and interaction of the medical staff was also difficult, for two main reasons. At the beginning, the prototype setup time seemed to be frustrating for them, but it was almost an unavoidable fact. Additionally, the natural human mistrust of new technology affected the expected first feedbacks. Nevertheless, after the training days, medical staff could adopt the new procedures,

and finally noticed system features including the round time optimization achieved by using wireless communication. This fact was directly reflected in having more availability to take care of patients.

To summarize, our deployment experience provides practical feedback in order to adapt and improve future communication architectures. Since testing is one of the methods to accomplish upcoming integration of health services, this paper has identified challenges in current scenarios, as well as analyzing and proposing suitable solutions.

## Acknowledgement

This work is supported by the Ministry of Industry, Tourism and Commerce of Spain, through “Avanza I+D Subprogram” and Project CARDINEA number TSI-020302-2009-43, and TIN T2C2 (TIN2008-06739-C04-03).

## References

- [1] Association of Automatic identification and Mobility, <http://www.rfid.org> (accessed on January 2011)
- [2] OSGi Alliance Website, <http://www.osgi.org> (accessed on January 2011)
- [3] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., Schooler, E.: SIP: Session Initiation Protocol. Internet Engineering Task Force, RFC 3261 (June 2002)
- [4] Lopez, P.P., Orfila, A., Mitrokotsa, A., van der Lubbe, J.C.A.: A comprehensive RFID solution to enhance inpatient medication safety. *International Journal of Medical Informatics* (2010)
- [5] Lin, W.-W., Sheng, Y.H.: Using OSGi UPnP and Zigbee to provide a wireless ubiquitous home healthcare environment. In: *The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies* (2008)
- [6] Sadat, A., Sorwar, G., Chowdhury, M.U.: Session Initiation Protocol (SIP) based Event Notification System Architecture for Telemedicine Applications. In: *Proceedings of the 5th IEEE/ACIS International Conference on Computer and Information Science* (2006)
- [7] Morales, A., Valladares, T.R., Alcarria, R., Alonso, D., Platas, S.: Communication Architecture for Smart Service in Hospital Environments. In: *Proceedings of Workshop of Ambient Assisted Living, IWAAL 2010, Valencia, Spain*, pp. 61–68 (2010)
- [8] Cardinea Project Homepage, <http://cardinea.grupogesfor.com> (accessed on March 2011)
- [9] Open Context Platform OCP2, <http://ants-webs.inf.um.es/ocp2> (accessed on March 2011)
- [10] Hughes, R.G., Ortiz, E.: Medication errors: why they happen, and how they can be prevented. *American Journal of Nursing* 105, 14–24 (2005)
- [11] Benjamin, D.M.: Reducing medication errors and increasing patient safety: case studies in clinical pharmacology. *Journal of Clinical Pharmacology* 43, 768–783 (2003)
- [12] HL7 Health Level Seven International, <http://www.hl7.org> (accessed on January 2011)
- [13] Fuse Source Open source SOA, <http://www.fusesource.com> (accessed on January 2011)
- [14] Intermec Website, <http://www.intermec.com> (accessed on January 2011)
- [15] Java Agent Development Framework, <http://jade.tilab.com> (accessed on January 2011)