33rd Annual International Conference of the IEEE EMBS
Boston, Massachusetts USA, August 30 - September 3, 2011

SERVANDO: An extensible platform for home-care services providing

T. Teijeiro, P. Félix, J. Presedo and A. Gándara

***Abstract*— This paper presents an extensible distributed platform that aims to speed up the development of personalized telemedicine systems, dealing with a series of recurrent problems in this kind of system, particularly: (1) functionality encapsulation and reuse in a set of services; (2) communications between the patient's home and the hospital, through a flexible scheme for bidirectional message exchange; and (3) the interaction between patients and the system. Home supervision is carried out through last generation smartphones. To date, the platform has been used for the follow-up of patients with COPD and cardiovascular diseases.**

I. Introduction

The progressive incorporation of ICT in the home determines the appearance of new opportunities that cannot be ignored by the public health systems, considering the serious problems they have to deal with. These can be summarized as: i) the progressive aging of the population, with the consequent increase in the prevalence of chronic diseases; ii) the organization of its activity based on the specific and episodic treatment of diseases in acute phase; or iii) the difficulties for an adequate incorporation of new research-derived knowledge into routine care.

Telemedicine brings together a set of strategies, architectures, techniques and devices that deal with some of the aforementioned problems. Most of the solutions we find in the literature have as objective the management of a concrete disease, developing in most cases adhoc solutions for the control of the different processes involved in patient supervision: physiological signal acquisition, communications, processing, interaction with the patient, etc. Some proposals attempt to generalize the solutions to the problems of signal acquisition and the transmission, compression and cyphering thereof [1], [2], but we are still far from multipurpose solutions which take advantage of code reusing.

In our approach, we have designed and built a new extensible architecture with the basic objective of enabling the adaptation to the supervision of different diseases. This architecture will allow the dynamic integration of different medical services, responsible for providing the execution logic of the medical actions which will configure the task flow to be carried out during the supervision.

T. Teijeiro, P. Félix, J. Presedo and A. Gándara are with Centro de Investigación en Tecnoloxías da Información (CITIUS), 15782, Universidade de Santiago de Compostela, Spain. (email: tomas.teijeiro@usc.es)

This paper is organized as follows: In section II we describe the system architecture, explaining how the different components work and interact with each others. Section III focuses on the technologies we have used in the implementation. In section IV, we present some practical results regarding the deployment of our solution in a real telemedicine scenario. Conclusions are exposed in section V.

II. System architecture

A. Basic concepts

The architecture of SERVANDO is aimed at supporting the execution of medical actions in the patient's home, implemented by an extensible set of services. We define more precisely these two main concepts:

1) Medical action: Activity that is part of the home management of a disease, whose execution is made possible by the platform. Medical actions permit the characterization of the physiological state and evolution of the patient, the effective management of the therapeutic administration, or a better understanding of the disease. Examples of medical actions are spirometry tests, ECG monitoring, or blood pressure measurements.

2) Medical service: Distributed functional abstraction responsible for the execution and control of the different instances of a set of medical actions. An example is a monitoring service, responsible for obtaining and processing the patient's physiological signals.

B. Hardware arqchitecture

As shown in figure 1, at hardware level SERVANDO is distributed into two kinds of systems: (1) a *central information system*, consisting of a server, located at the medical centre, which coordinates the home supervision of the patients, manages communications, stores the information relative to each patient, and permits the visualization of the generated alarms during the supervision and the corresponding physiological parameters; and (2) a *mobile device* used by the patient at home, responsible for the acquisition of physiological data and the general interaction with the patient. In the following sections we shall describe the architecture components that make possible the supervision of the patient.

C. Medical services

Medical services constitute the extensibility model of the platform. These entities can be dynamically added

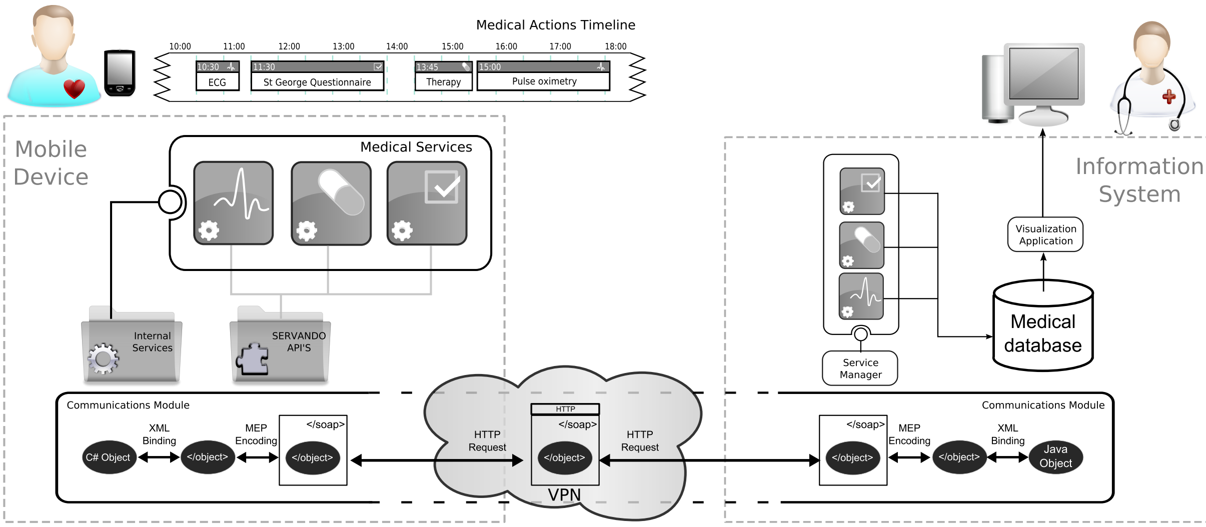

Fig. 1. Platform Architecture

or removed, which enables SERVANDO to be adapted to different supervision scenarios.

The architecture proposes an inversion of control model [3], in which the platform manages the main execution flow (including starting and stopping the platform, network and system state control, etc), while it delegates all tasks relative to the execution of medical actions to medical services, which will have access to both the native operating system APIs such as those offered by SERVANDO. This makes necessary to ensure that the installed services are trusted.

To date, the following medical services have been implemented:

1) Monitoring service: Its function is to manage the dialogue between the platform and a set of sensors responsible for the acquisition of biosignals, which will be processed to find relevant events in the course of the disease of the patient. The implementation is based on a previously designed multipurpose system for intelligent ubiquitous monitoring [4], which has been modified to adapt its design to the inversion of control model of SERVANDO . The service provides two kinds of actions:

1) Punctual measurement of physiological parameters: Type of medical action that results in a data vector bound to a concrete instant of time. Examples of actions of this type are blood pressure measurement, spirometry, temperature measurement, etc. Values are accumulated in form of a pseudoperiodical temporal series that can be systematically sent to the medical centre or processed to find interesting episodes (fever, for example).
2) Physiological signal monitoring: Type of medical action that results in a periodical temporal series automatically acquired. Examples of actions of this type are electrocardiography or pulse oximetry. Physiological signals are acquired by sensors with Bluetooth connectivity and processed to find relevant events, such as ischemia episodes, arrhythmias, or apnea episodes, among others [5]. In this manner, we aim to provide valuable information with respect to the state and evolution of the patient beyond simple signal acquisition and transmission, which is clearly ineffective in terms of health resource management. Moreover, this service attends to requests for signal fragments (e.g. electrocardiogram), which will be transmitted to the server on demand.

2) Therapy service: Service responsible for the management of the therapy prescribed to the patient, through simple notifications of drugs and doses, with visual information to make easier the proper identification of the concrete drugs and other helpful information presented in a suitable form to patients with visual difficulties. To perform this type of action, the patient will simply confirm its completion.

3) Questionnaire service: This service has been designed for the presentation and processing of questionnaires that will be filled out as forms via the mobile device screen. Some interesting general purpose questionnaires are quality of life tests, such as the SF 36 Health Survey test or, in the context of COPD patients supervision, the St George Respiratory Questionnaire. This service also allows the patient to record symptoms.

The distributed nature of the medical services lies in the management of the communications. As is explained in detail in the following section, the communications API permits the bidirectional transmission of messages between the two endpoints of a service (one of them situated in the mobile devices and the other in the central server). Therefore, services will define their own communications model in a flexible manner, in terms of both data types to transmit and the exchange scheme.

D. Communications module

This component has been designed to support any communication scenario that may be necessary in the context of remote supervision, which includes continuous transmission of physiological signals, punctual communication of symptoms, periodical transmissions or on-demand communications. This makes it necessary a high abstraction level and flexible communications system.

In our approach, we have opted to offer a mechanism of message exchange in which each service will be able to send or receive any kind of message that adheres to an object-oriented design. Transmissions can be performed according to a set of provided message exchange patterns (MEP) that we describe below:

1) Synchronous send: This pattern corresponds to a simple remittance of a message in a synchronous way. The invoker attaches the object to be transmitted and waits until the transmission has ended, receiving a delivery report as a result.

This pattern has been designed for non-frequent discrete communications, or for those necessarily synchronous. For example, the communication of relevant events, such a cardiac arrhythmia, fits this pattern.

2) Asynchronous send: It corresponds to a simple send of a message in an asynchronous way. The service invokes the method attaching the object to transmit, and it is immediately able to continue its job, while the transmission is performed in an independent thread. The invoker will not know the state of the transmission, and its delivery is not guaranteed.

This is a low-cost fast pattern. For this reason, its main application is for small, non-critical, frequent communications, such as, for example, the periodical transmission of messages for the clock synchronization between the mobile devices and the server.

3) Asynchronous send with delivery confirmation: This pattern is equivalent to the previous one, except that it allows the state of the communication to be checked. As result of the invocation, the service will receive a *ticket* that will make it possible to check if the delivery ended properly, if it is still in process, or if it ended with errors.

This communication model is designed for periodical transmission with variable criticalness. For example, the continuous sending of ECG signal can be adapted to this pattern, since if a determined fragment of signal cannot be properly sent, the service can decide to forward it or not, based on its relevance.

4) Reliable asynchronous send: This pattern ensures as far as possible the correct message delivery, even if errors occur during transmission. To achieve this, the transmission is retried during a period selected by the invoker. If after that period the transmission cannot terminate correctly, the service will receive a report.

This model is designed for the periodical communication of relevant events. For example, the periodical transmission of trend diagrams by the monitoring service uses this pattern, allowing it to ignore the transmission process completely.

5) Send/receive: This corresponds to a conventional request/response transmission. The service will invoke this method attaching the object to transmit, and as a response it will receive another object from the other end-point. Objects can be of any type, and the responsibility of adding the semantics needed to determine the type of the received object lies with the invocator service. This communication pattern is synchronous.

The use of this pattern is basically oriented to the achievement of remote requests, such as, for example, the on-demand transmission of ECG fragments.

Over these five basic patterns the services can build new complex patterns, such as subscription, periodical send, etc. This system allows medical services to choose both the data model and the communication model to be used, and facilitates their definition. In the case of message types, the service should simply define the class model that it will work with, and the instances of that classes will be transmitted as messages. As regards message exchange patterns, any complex pattern can easily be implemented by encapsulating some of the basic patterns provided.

E. Other APIs and internal services

Apart from the service extensibility model and the communications module, a set of programming APIs are provided to simplify the implementation of new medical actions. Moreover, a set of internal helper services have been created for managing specific aspects while the platform is running. A detailed description of these components is provided below:

1) SERVANDO APIs: A set of graphical APIs is provided to simplify the development of new medical services. These allow the creation of usable, simple, and visually attractive user interfaces, as well as providing internationalization and notifications support. Moreover, a log API is provided for debugging purposes.

2) Service manager: This component is responsible for obtaining the set of medical services that should be loaded during the starting of SERVANDO from the general settings of the platform, and then starting them, registering them as medical action providers, and giving them the appropriate access to the SERVANDO APIs.

3) Persistence engine: Responsible for ensuring the persistence of all data needed for the platform to work properly (general settings, execution logs, etc), as well as for recovering that information from local storage and making it available to those components that require it.

4) System Monitor: This module is the responsible for controlling those conditions needed for the platform to operate correctly, such as the battery charge state.

5) File transfer service: This service allows files to be transmitted between the central system and the mobile devices, with the objective of gather execution logs, apply remote updates, etc.

6) Clock synchronization service: Its objective is to keep the clock time on the mobile devices synchronized with the central system, which is a basic task for the correct synchronization of the monitoring events.

III. Technologies and implementation

A. Development platforms

For the implementation of our solution we have chosen Java SE for the central server side, and .NET Compact Framework for the development on the mobile devices. The main reasons for this choice were the good performance and compatibility of both platforms, and the previous experience we had on these platforms, which has encouraged faster development. Still, the choice of open technologies for communications and data representation has permitted us to port the mobile devices side to the Android system, which will hold the main line of development in the future.

B. Message interchange technology

To implement the designed message interchange mechanism, we have opted for the adoption of Web services technologies, including XML, SOAP, and HTTP. As shown in figure 1, at application level, messages are represented as objects of the programming language of each one of the communication endpoints. When a transmission is made, objects are converted to a compatible XML representation (using the XMLSerializer API in .NET and JAXB in JAVA), and are transmitted through SOAP/HTTP following one of the implemented message exchange patterns. Due to the need for mobile devices to work as servers (see section II-D), it has also been necessary to implement an own Web server in this side, as well as a SOAP gateway to dispatch incoming messages.

C. Communications security

To guarantee that no data can be intercepted by third parts during the transmission between mobile devices and the information system, we have opted for the integration of virtual private network technology (OpenVPN software) in our communications scheme. Besides strong encryption and authentication on all communications, this choice has many advantages, such as transparent security for the application, generic data compression or identification by IP address.

IV. Results

At the time of writing, SERVANDO is being used in an experimental study in collaboration with the *University Hospital of Santiago de Compostela* for the follow-up of patients with COPD and cardiovascular diseases. In this prototype, we have used a Toshiba TG01 mobile phone, which interacts with Bluetooth sensors for the acquisition of ECG signal, blood oxygen saturation and spirometry measurements. Since no medical results are available yet, we can conclude that the proposed scheme is technically feasible with current devices. Respect to signal processing algorithms, as shown in [4], the CPU usage is about 2%. Regarding communications, the transmission rates are highly dependent on the patient's Internet connection, but with a basic DSL line with an upload rate of 528 bps we achieve a transmission rate of 47 KB/s, which is enough even for real-time signal transmission.

V. Conclusions

In this article we present the design and implementation of a distributed platform for providing home-care services, establishing a framework for the development of applications for ubiquitous medical services, with implementations for mobile devices. In short, the main contributions of our work are:

- The system is extensible, offering an interface for the building of services which implement new functionalities, both for patient supervision and for the proper working of the platform.
- We provide a series of solutions and services that solve common problems in patient follow-up and that can be used regardless of the medical application to which the system is oriented, including communications APIs, execution logs, user interaction, transmission security, etc.

With this proposal, we aim to promote standardization in the development of telemedicine systems by reusing components, which will facilitate the construction of complex and multipurpose home-care systems.

As a future work, we have planned the development of a new module for managing a supervision agenda that will control the temporal organization of the executions of medical actions, allowing the protocolization of the follow-up process.

VI. Acknowledgement

This project was supported by the Spanish Ministry of Science and Innovation (MICINN) under grant TIN2009-14372-C03-03, and by the Xunta de Galicia under grant INCITE08SIN002206PR.

References

[1] F. Sufi, Q. Fang, S. Mahmoud, and I. Cosic, "A mobile phone based intelligent telemonitoring platform," in *3rd IEEE/EMBS International Summer School on Medical Devices and Biosensors.*, 2006, pp. 101–104.

[2] E. Zeeb, G. Moritz, W. Thronicke, M. Lipprandt, A. Hein, F. Muandller, J. Kruandger, O. Dohndorf, A. Litvina, C. Fiehe, I. Luandck, F. Golatowski, and D. Timmermann, "Generic platform for advanced e-health applications," in *12th IEEE International Conference on e-Health Networking Applications and Services (Healthcom)*, 2010, pp. 201–208.

[3] M. Fowler, "Inversion of Control Containers and the Dependency Injection Pattern," *martinfowler.com*, 2004. [Online]. Available: http://martinfowler.com/articles/injection.html

[4] J. Presedo, M. Tarascó, T. Teijeiro, and P. Félix, "A PDA-based modular and multipurpose system for intelligent ubiquitous monitoring," *WISP 09*, 2009.

[5] J. Presedo, J. Vila, S. Barro, F. Palacios, R. Ruiz, A. Taddei, and M. Emdin, "Fuzzy modelling of the expert's knowledge in ECG-based ischaemia detection," *Fuzzy Sets and Systems*, vol. 77, no. 1, pp. 63–75, 1996.