Ann. Telecommun. (2010) 65:545–556
DOI 10.1007/s12243-010-0164-4

User-driven context aware creation and execution of home care applications

Carlos Baladrón · Javier M. Aguiar · Javier Gobernado · Belén Carro · Antonio Sánchez

Received: 30 September 2009 / Accepted: 9 March 2010 / Published online: 19 May 2010

Abstract In the Internet world, the tags Web 2.0 and user centricity have been used in recent times to reflect the democratization of the Web where end users are no longer passive entities and have assumed a key role in which they create content themselves. The evolution of this paradigm in the Internet has even given birth to tools like Microsoft Popfly or Yahoo Pipes, thanks to which end users with no specific computer skills are able to create their own highly personalized services. In an environment like home care, where each user has very specific individualized needs, the application of this new paradigm seems very promising, and to promote that philosophy porting, this paper presents a system to allow each individual and his/her carers to build, execute, and manage the specific personalized monitoring, comfort, automation, and safety applications required. Additionally, thanks to built-in, context awareness capabilities, the applications created are automatically adapted to the context of the users inside the proposed system, eliminating the need for frequent human–machine interaction which is of the uttermost importance for impaired users.

C. Baladrón (✉) · J. M. Aguiar · J. Gobernado · B. Carro
Communications and Signal Theory and Telematics Engineering Department, University of Valladolid,
Campus Miguel Delibes 47011 Valladolid, Spain
e-mail: cbalzor@ribera.tel.uva.es

J. M. Aguiar
e-mail: javagu@tel.uva.es

J. Gobernado
e-mail: jgobher@ribera.tel.uva.es

B. Carro
e-mail: belcar@tel.uva.es

A. Sánchez
Telefónica I+D,
Parque Tecnológico de Boecillo 47151 Valladolid, Spain
e-mail: ajse@tid.es

Keywords Home care · User centricity · SOA · Web services · Home assistance · Rehabilitation

Abbreviations

IT	Information technology
OPUCE	Open Platform for User-centric Service Creation and Execution
SPICE	Service Platform for Innovative Communication Environment
SMS	Simple Mobile Services
GUI	Graphical user interface
AP	Access point
ebXML	Electronic Business eXtensible Markup Language
XCAP	XML Configuration Access Protocol

1 Introduction

The home care environment is wide enough to host all kinds of imaginable patients, including impaired, blind or deaf people, persons with high risk of heart attack, old people with reduced mobility, Alzheimer's patients, etc. It is obvious that the requirements for home care exhibited by each of them vary greatly; from simple glucose and diet monitoring for diabetics to a complete physical care for quadriplegics.

Human carers, including both professionals and family, are normally responsible for the home care tasks. This means that these carers are usually in charge of repetitive procedures taking place in the physical proximity of the patient during long periods of time. This results in high workloads for health professionals which have to dedicate a lot of resources to find, train, and maintain experts carers in each specific area, but also for the family which sometimes have to set aside their personal lives and jobs in order to give proper care to the patient, ending usually in heavy physical and psychological stress for everyone.

Recent e-health trends try to offer solutions for this situation in the form of information technology (IT)-based sensors and devices that can be operated remotely or automatically, relieving the carer from the need to stay physically near the patient or even from performing certain actions. Proposed solutions range from special purpose sensor systems for specific pathologies [1, 2] or intelligent robotic devices [3] to generic programmable frameworks for telecare [4–6] and Web-based solutions [7]. On the other hand, domotics are also experiencing great advances, and control systems like [8] and [9] may also represent an aid for people with mobility problems to program for instance how to automatically turn on and off room lighting.

However, all these solutions still present two key problems: First, most of them are aimed at very specific pathologies and scenarios or are programmed with very specific functions; second, they are definitely not user friendly and lack usable interfaces, so IT experts are required for fine tuning the system to each patient in a process that may take days or even weeks of work. Specifically, [10] for instance presents a good example of how a pervasive context awareness system can help greatly improve the quality of life of a patient and his/her carers, but as it requires IT experts to implement each service, it will take an enormous amount of resources to implement highly personalized applications.

It is therefore very difficult to adapt the operation of these systems to the specific requirements of each user, making them unsuitable as integral home care solutions. They are very useful at controlling a specific facet of a pathology, but it is impossible (or extremely difficult in the best case, requiring a dedicated IT professional) to adapt them to help in everyday's life in order to perform tasks like autoconfiguring a health device or sending warning messages depending on the patient geolocation, activity, and/or preferences.

This paper presents a system for the user-driven integral home care of a wide array of patients, from impaired users who want to automate the functioning of a smart home to patients which require constant monitoring from an expert system. The idea behind this system spawns from the user-centricity paradigm arising lately in the Web 2.0 environment which promotes the creation of high-level abstractions and control interfaces to allow end users to manage several aspects of their IT experience.

This paper is organized as follows. Section 2 introduces the user-centricity paradigm and its current state of the art. Section 3 deals with how this paradigm can be applied to the home care scenario and presents the proposed system. The architecture of the home care system is explained in Section 4 and its implementation and application to test scenarios exposed in Section 5. Finally, a set of conclusions is extracted in Section 6.

2 Trends in user centricity

Web 2.0 [11] is the name that some people give to the new landscape of the World Wide Web where social networking and interactive symmetric communication has substituted the old unidirectional paradigm where developers built sites with information and people consumed it. In Web 2.0, every user is a provider and a consumer at the same time.

The key feature of the Web 2.0 is user centricity, a philosophy where the users themselves create the contents of the Web. Sites like Youtube.com, Wikipedia.com, or Sourceforge.com allow the individuals to upload their own multimedia files, knowledge, or code to be consumed by other users. The latest trend in user centricity is the user-centric service creation, thanks to which end users with no specific knowledge are able to create their own applications using high-level abstraction and composition interfaces. Examples of this are the online mashup editors from Yahoo and Microsoft, Yahoo Pipes and Microsoft Popfly, or the cluster of user-centric service creation projects from the European Commission Seventh Framework Programme, including Open Platform for User-centric service Creation and Execution (OPUCE) or Service Platform for Innovative Communication Environment (SPICE).

These tools provide interfaces for the end user to create simple applications following the "jigsaw model": A palette of atomic functionalities (building blocks) is offered and the user drags and drops some of them, linking their inputs and outputs defining a workflow. In order to create a service to forward messages from an email account to a mobile phone using short message service (SMS), the user would select the Receive_Email and the SMS building blocks and would link the output of the first one to the input of the second.

Specifically, along this paper, the OPUCE service model [12] will be used. This model is event-driven and sees the building blocks as entities which expose certain actions, fire certain events, and exhibit certain properties, which may change during the execution. Using this model, the email forwarding service using SMS previously mentioned would link the when_email_received event of the Receive_Email building block with the Send_SMS action of the SMS building block, creating a workflow that would almost be readable in natural language: "when an email is received, send an SMS". The property body of the Receive_Email block would be linked to the property body of the

Send_SMS block so as the content of the SMS sent is the same as the content of the email received.

3 User centricity applied to home care

3.1 User-centric home care services

The main idea behind this work is that the user-centricity paradigm presented in Section 2 can be applied to the home care domain to obtain very useful results. Currently, if a person taking care of an Alzheimer patient wants a service which automatically sends an alert SMS to his/her mobile phone when the patient leaves home, an expert is required to design and implement an ad hoc system.

However, in a user-centric system, that person will only need to build the required service. In this case, users will open the service editor, select from the palette the basic building blocks GPS Location, Proximity, and SMS, and create a workflow linking GPS Location (configured with the ID of the tracking device of the patient) to the Proximity block (configured with the coordinates of patient's home) and the event Outside_Location of the Proximity block to the action Send_SMS of the SMS block.

Other patients and carers with very different requirements will be able to use the same system for completely different home care applications just by changing the basic building blocks used, which may include domotic controls, biometric sensors, direct multimedia contact with medical experts, etc.

Along this work, a multipurpose home care system of the kind required for this user-centric operation will be described.

3.2 Rationale for applying user centricity to home care

As shown in Section 2, the user-centricity paradigm has already permeated the Internet and the Web, and it is slowly entering other domains. There are two key factors to decide if a given domain could benefit from the user-centricity paradigm:

- The need that users of the system have for specific personalized applications
- The granularity of the personalization required

In the domain of office applications, every user has the same needs. They want to be able to write and format text, perform account operations, etc. Therefore, it is very unlikely that a lot of users would like to design and implement its own office application. Something similar happens for instance in the domain of operating systems. The need that most users of operating systems have is simply to effectively control the device's hardware. That does not mean that some very specific individuals have very specific needs for these domains. They can always hire a team of developers and design their own specialized office applications and operating systems at a low level. The cost to implement a user-centric set of tools to build personalized office applications or operating systems is potentially very high, and given the slow amount of users that would be interested in them, the market prize for the user centric tools had to be so high that it would be probably cheaper to simply hire a team of developers.

Additionally, the granularity of the personalization required is also an issue in these domains. Even in the case that a user-centric tool for the creation of personalized operating systems could be built, the granularity of the control obtained would be too gross for most of the personalization required. The user-centric paradigm is based on the "jigsaw" model where a new complex application is created putting together atomic building blocks. Therefore, some parameters are completely out of reach for the user acting as the creator: those used to hardcode the building blocks on its own. When a new operating system has to be built due to specific needs, those needs are at a very low level (performance issues, new hardware, etc.), and therefore, it is impossible to abstract anything by hiding parameters inside the building blocks.

The case of home care applications is completely different though. First, users of home care applications have very specific personalization requirements due to the high numbers of variables shaping each user's situation: the specific pathology, physiology, biometric signals, lifestyle, home and workplace layouts, customs, etc. An application built to warn an aged individual if patient has not taken his/her pills before going to sleep is completely useless for an impaired youngster who wants to automate the operation of the lights of his/her house based on his/her indoor location.

Second, the level of granularity for the applications is adequate to allow a proper abstraction level because home care applications could be defined using the specific language of the domain. Low-level concepts (performance issues, transmission protocols, memory allocation, etc.) are normally not involved in the definition of these applications and therefore could be hidden inside the building blocks.

Because of these two reasons, the home care domain is a perfect candidate for the application of the user-centricity paradigm.

3.3 User-centric system for home care

The aim of the user-centric system for home care proposed is to act as a coordination center for all the electronic devices surrounding the life of an individual who requires home cares and allow an integrated orchestration of all of them to accomplish complex tasks that they could never address isolated. To control this coordination layer, a set of user tools are built on top to allow the easy implementation of coordination applications using graphical interfaces and a high abstraction level designed for users with no specific programming or computer related skills. Additionally, a context management subsystem retrieves, processes, and serves context data in order to allow the implementation of automatic context-based adaptation. This structure is reflected in Fig. 1.

The system can be implemented in two different configurations:

Collective configuration The system is owned by a home care organization (e.g., a hospital, an association of impaired people, etc.) and lies inside a server in that organization's premises. The communication with the end user devices is performed through the Internet and the system is prepared to handle several users at a time. This configuration is suitable for groups of people with the same basic requirements or people unable to use the system themselves that need expert remote monitoring. For instance, in the case of people suffering from Alzheimer or high attack risk, the respective medical units in a hospital may install the system in their offices and register their patients, then implement common home care applications for each group. That could imply monitoring applications to track their location when they leave home unattended or sending alerts to the emergency services and the patient himself when the cardiac rhythm shows dangerous parameters.

Individual configuration The system is owned by the patient or his/her carers and lies inside a PC located in the patient's home. Communication with end user devices can still be carried over the Internet, but local methods such as wired and wireless local area networks can also be employed. This configuration is suitable for individuals who can use the system on their own to implement applications and have very specific, personal needs, for instance an impaired person who wants the system to automatically turn on and off the lights depending on the indoor location.

In both cases, three kinds of actors have been defined for the operation of the system:

- The patients are the persons (aged, impaired, ill, etc.) who benefit from the home care system, the subjects of the home care applications. They could use the user tools to implement their own applications.

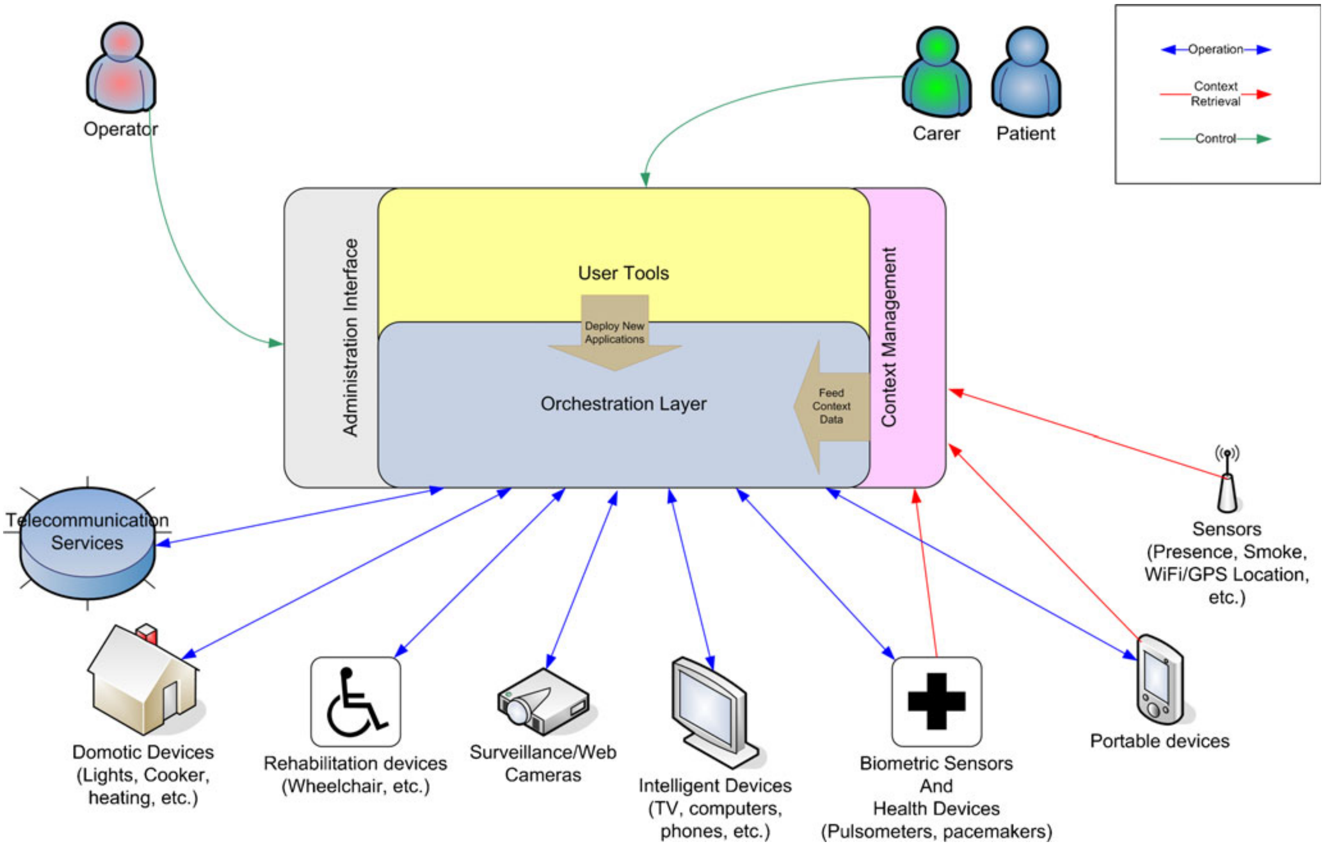
Operator
Carer
Patient
Operation
Context Retrieval
Control
User Tools
Deploy New Applications
Orchestration Layer
Administration Interface
Context Management
Feed Context Data
Telecommunication Services
Sensors (Presence, Smoke, WiFi/GPS Location, etc.)
Domotic Devices (Lights, Cooker, heating, etc.)
Rehabilitation devices (Wheelchair, etc.)
Surveillance/Web Cameras
Intelligent Devices (TV, computers, phones, etc.)
Biometric Sensors And Health Devices (Pulsometers, pacemakers)
Portable devices

Fig. 1 User-centric home care system

- The carers are the people directly taking care of the patient. They could be the family or professionals dedicated to care activities. The activities they perform in taking care of the patient could be partly automated, assisted, or helped by the home care applications of the system. They can also implement new applications through the user tools.
- The operator is the owner and administrator of the system. It could be a hospital, a medical foundation, or even the patient or one of the carers, depending on the configuration (collective or individual) chosen.

4 System architecture

User-centric service creation has been exhaustively promoted in the Web 2.0 and Telco 2.0 environments lately, as it has been exposed in Section 2. The OPUCE project [12, 13] designed a platform for the creation and execution of converged services, merging the infrastructure of telco operators with Internet Web services and including a set of tools oriented to non-expert end users. In the user-driven home care system proposed, several modifications have been performed over the OPUCE prototype in order to adapt it for the home care environment, as detailed along this section.

The global architecture of the home care system proposed is depicted in Fig. 2. Three main blocks are identified. The Management Portal (based on the OPUCE Web Portal, includes the Service Editor and the Building Block Manager, evolved versions of the Advanced Editor and Base Service Manager), the Service Execution Environment (comprising the Building Block and Composed Service Repositories—known in OPUCE terminology as Base Service and Service Repositories, respectively—Building Block Container and Orchestrator), and the User Information Manager.

4.1 Management portal

The Management Portal is the graphical user interface (GUI), a Web portal embedding a set of tools designed to control all the functions of the home care system. Specifically, it is employed to:

- Configure the system and introduce user data;
- Manage execution of services;
- Compose new services;
- Install new service building blocks.

It is built as a collection of portlets for each feature and runs on top of JBoss Application Server. Being a Web tool means that it could be used remotely for cases where the system needs to be controlled on the move or outside the application environment, especially when the patient is not able to control the system on his/her own and a remote carer has to do it. However, it can also be operated locally so it is not necessary to consume an Internet connection.

The OPUCE Web Portal was originally designed to support services which mainly were started, executed once, and killed, and therefore the only service management action available was "start." The user pressed the run button, the service was executed, and when the workflow was finished, it was purged from the execution environment. The OPUCE Web Portal

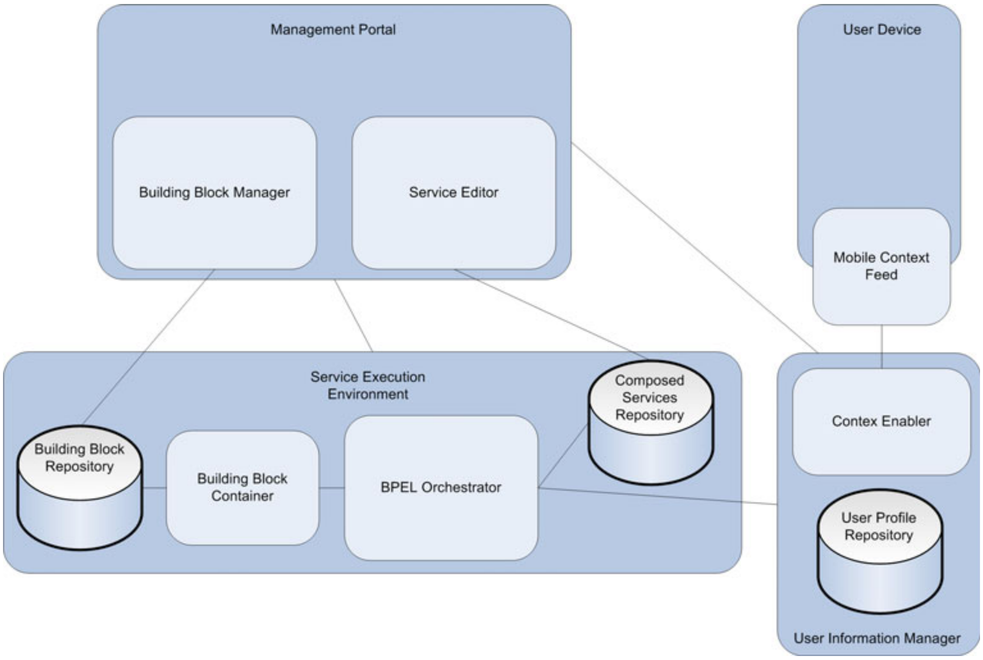

Fig. 2 User-centric home care system architecture

lacked a functionality to allow monitoring and stopping of active services.

More advanced services, however, may be active for a period of time, monitoring certain parameters or waiting for certain events and reacting accordingly. A clear example is a service to automatically turn on and off the lights when the patient enters or exits a room.

Therefore, an interface to see exactly which services are being executed at a given moment and to start and stop them by pressing a button has been added to the home care system Management Portal.

Two additional tools are embedded inside the Management Portal, the Service Editor and the Building Block Manager.

- The Service Editor is a graphical tool designed to allow the easy composition of new services. The user builds a workflow in the canvas area by dragging and dropping building blocks from the palette in the left and linking the inputs and outputs of these blocks specifying the service logic. This logic follows the event–action service model presented in Section 2. The configurable parameters of each building block are accessible through pull-down menus. Management operations like load and save are available, and they are linked to the Composed Services Repository.

Additionally, the Service Editor offers two support functionalities for the service composition: a service checker and a service simulator. The service checker is a tool that scans a service description applying a set of constraints and determines if the service is free from formal errors. When a user finishes composing a service, and before saving it, the service checker is run to identify common errata, like a building block not linked to anything else, an empty parameter, etc. The service simulator allows the execution of a service over a synthetic environment. The user then can run the service and check if it behaves actually as intended under controlled conditions. For instance, a service including an unbounded loop which sends an SMS may be unintentionally introduced in the composition by the service creator, but it could be formally correct, so the service checker will not show any errors. Once executed, the service will send a myriad of SMSs, costing a lot of money. The user acting as service creator may use the service simulator to identify this kind of composition errors which cannot be automatically detected by the service checker.

- The Building Block Manager is a tool to install new building blocks into the system. It takes the code of a Web Service as an input in order to automatically generate a Building Block description, able to run in the execution environment, and store it into the Building Block Repository. The Building Block Manager is linked to this repository so the system operator can always monitor which Building Blocks are installed and perform management (i.e., updating, cleaning, etc.) operations over them.

Adding new building blocks to the system expands its functionality. For instance, a user may have recently acquired a new domotic device to automatically control room lights through a WiFi interface. That interface could be wrapped as a Web Service and introduced in the Building Block Manager to be installed in the system. Afterwards, that user may introduce light control features into his/her compositions.

4.2 Service execution environment

The Service Execution Environment comprised four different elements:

- A Building Block Repository used to store the description of the building blocks. It exposes a combined Electronic Business eXtensible Markup Language (ebXML)/XML Configuration Access Protocol (XCAP) interface to the rest of the platform elements.
- A Composed Service Repository used to store the description of the composed services. It also exposes an ebXML/XCAP interface.
- A Building Block Container which is an application server where the building blocks of the repository are running as Web Services. It is possible to count with several containers to host Web Services implemented with different technologies (Java, C#, etc.). The system is, however, implemented with the JBoss application server.
- The Orchestrator which is in charge of interpreting the logic of the composed services, invoking the building blocks hosted in the container whenever necessary.

The OPUCE platform also included a module called Service Lifecycle Manager [14] which was in charge of performing complex deployment tasks and resource management. Those operations are necessary for a huge system designed to be used by thousands of people with several machines acting as containers, but in the relatively small environment of home care, they are not required. Therefore, a very simple, scaled-down version of the Service Lifecycle Manager has been put in place for this scenario, with the only task of deploying the building blocks into the container and the service logic into the orchestrator.

However, in the case of a big system providing home care services to a big community of users (a complete hospital or health institution for instance), it is possible to use the advanced Service Lifecycle Manager to perform complicate deployment, provisioning, and resource management actions.

4.3 User information manager

The User Information Manager is a software module designed to retrieve, store, and provide user's context and profile information to the services so they can be adapted to each user automatically. The User Information Manager comprised two databases:

- The User Profile Repository which is accessed by each user of the system to manually store profile data and preferences.
- A Context Enabler which automatically retrieves context data from the user. It is able to aggregate data from different sources, like GSM-based location servers offered by telecom operators, but also from a small application installed in the user's device (PDA, smartphone, etc.) which collects additional data, such as GPS location, applications being used, conversational status, etc.

This personal information is consumed dynamically by the Service Execution Environment to allow automatic personalization of services. For instance, a service may be composed to send an SMS to a relative whenever an old person leaves home. The same service could be shared among all clients of a home caring association because when composing the service, the parameters ID User to locate and target phone could be set to the special values user.ID and user.carer_phone (the list of these special values known as "user sphere" [12] is presented in the Service Editor, so they are very easy to use). Then, at execution time, the Service Execution Environment would retrieve the values ID and carer_phone from the User Information Manager for the specific patient which is consuming the service.

Additionally, context data served by the User Information Manager could be very useful if coupled with advanced reasoning algorithms [15], resulting in machine learning of user's habits and intelligent adaptation performed automatically.

5 Validation and use cases

In order to demonstrate the feasibility of the proposed idea, a prototype implementation of the home care system has been built. A set of service building blocks has also been put in place in order to show the different kinds of features that the system is able to handle and to compose a variety of demonstration services. While the system is completely functional, including the interfaces with the real world, in order for it to be fully operative, it is enough to implement Web Service controllers for the hardware devices and plug them into the building blocks.

5.1 Service building blocks

These test building blocks have been included in the prototype system:

- Wheelchair Monitor building block: This building block exposes several dynamic parameters of a wheelchair, like geolocation, velocity, acceleration, yaw, etc. The chair is equipped with several sensors to retrieve the data and a small computer acting as a Web server through a WiFi/3G interface.
- Holter, Blood Pressure, Pacemaker, and Cardiac Events Monitors building blocks: This collection of building block grants access to the parameters measured by the corresponding devices.
- Implanted Defibrillator Actuator: This block allows the remote operation of an implanted defibrillator.
- GPS Location: This block provides GPS-based location data based on GPS suitable for outdoor environments.
- WiFi Location: This block provides precise location based on WiFi triangulation in adapted multi-access point (AP) indoor environments.
- Domotic Control: This block allows the remote operation of a domotic house, such as turning on and off lights, TV, oven, etc.
- Ambiental Data: This block offers data about temperature and humidity of the user's environment through integrated sensors in a smartphone.
- Proximity: This block calculates the distance from one set of coordinates to another or from one set of coordinates to a fixed point/area.
- If..Then: This block allows the introduction of flow control conditions.
- SMS: This blocks receives and sends SMS messages.
- (Video)Call: This block manages the establishment of multimedia calls among different parties.
- Health Expert Interface: This block represents a GUI running at a hospital or a doctor's mobile device where alerts and medical data are shown and decisions about the workflow of a given service could be taken.

5.2 Validation scenarios

Three validation scenarios have been successfully implemented to illustrate the potential of the system under very different circumstances, patients, requirements, and aims.

5.2.1 Wheelchair control

The first scenario assumes an impaired person which uses a wheelchair. He/she would like to automate the lighting system of his/her home so the lights of a room are turned on when he/she enters and off when he/she leaves. He/she uses a WiFi-enabled smartphone and installs a multi-AP WiFi-based precision location system at home. Additionally, the house is equipped with a domotic control system which allows automatic operation of several parameters like lights and TV. The user also owns a PC where he/she installs the home care system.

He/she uses the creation tools to set up the workflow depicted in Fig. 3. The location is fed into several proximity blocks (one per room in the house) configured with the coordinates of each room. These blocks fire the event is_near when the smartphone enters the corresponding coordinates and is_not_near when the smartphone exits them. These events are fed into the appropriate actions of the Domotic Control block.

Additionally, if the user wants to set up an emergency service to detect if he/she has suffered a fall with the wheelchair, he/she could also use the blocks Wheelchair Monitor, feeding the acceleration and yaw parameters into the If..Then block to identify situations where the chair could have fallen down the stairs or from certain height and in that case fire an event that will send an SMS to a carer.

5.2.2 Heart monitoring

In this use case, it is assumed that the cardiovascular unit of a hospital has installed the home care system in order to implement an automatic risk detection and actuation service on high risk patients. These patients will be equipped with 3G smartphones and specific monitoring and health equipment (Holters, pacemakers, and implanted defibrillators). The doctors at the hospital could use the creation environment to set up the risk tracking service as shown in Fig. 4.

The Holter, Blood Pressure, Pacemaker and Cardiac Events Monitors feed data into two If..Then blocks, together with the Ambient Data. One If..Then block is configured with conditions that identify when the user is doing too much exercise for his/her environmental conditions by examining heart rate, temperature, humidity, etc., and if those conditions are met, an SMS alert is sent to the patient. The other If..Then block monitors signals from the Pacemaker, Implanted Defibrillator, Holter and Cardiac Events Monitors to search for high heart attack risk situations. If such a situation is developing, the Health Expert Interface block receives an event, which causes the interface to pop up in the smartphone of the doctor in charge, showing the cardiovascular data of the patient and allowing a better tracking and crisis management.

5.2.3 Alzheimer patient tracking

In this use case, the carer of a person suffering Alzheimer installs the home care system in his/her PC in order to automate tracking via a smartphone which could be attached

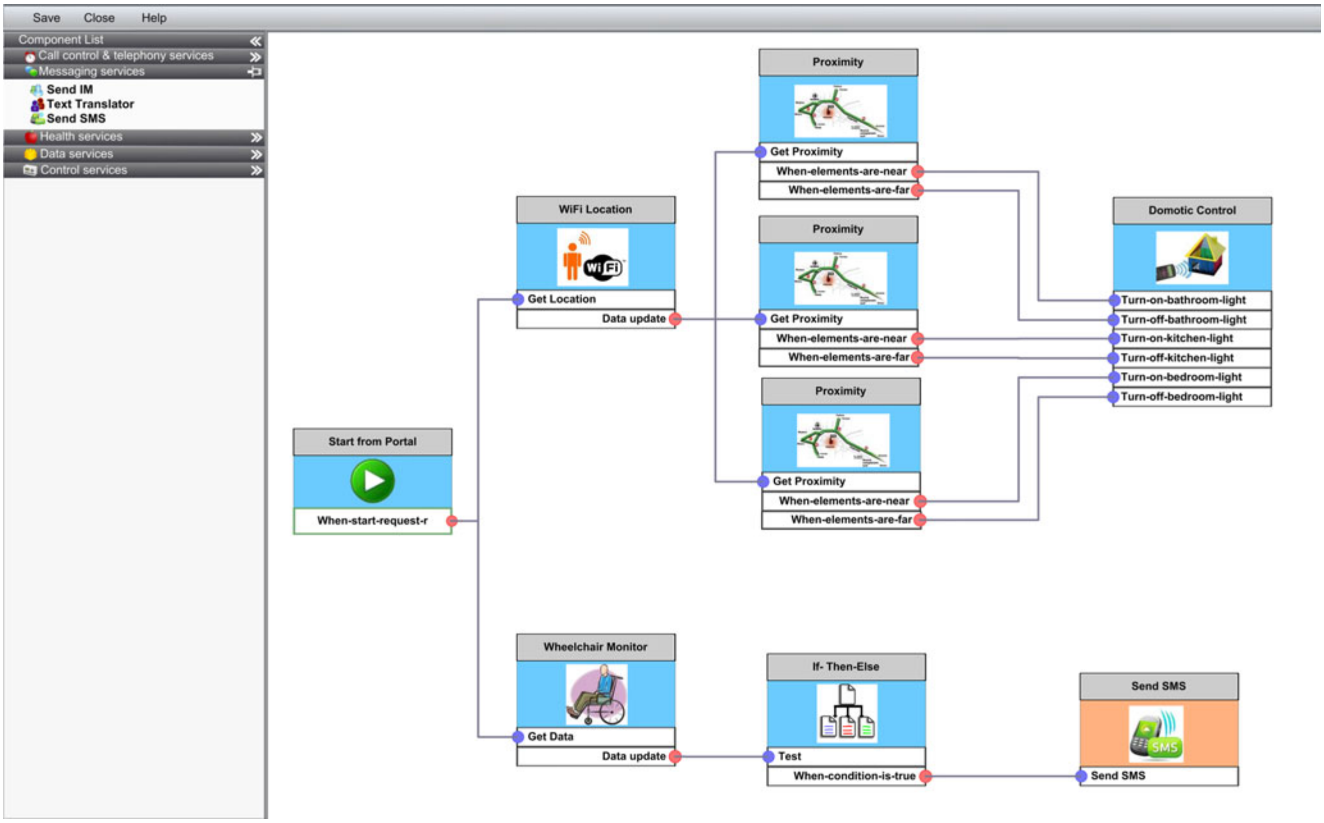

Fig. 3 Wheelchair control scenario workflow in the Service Editor

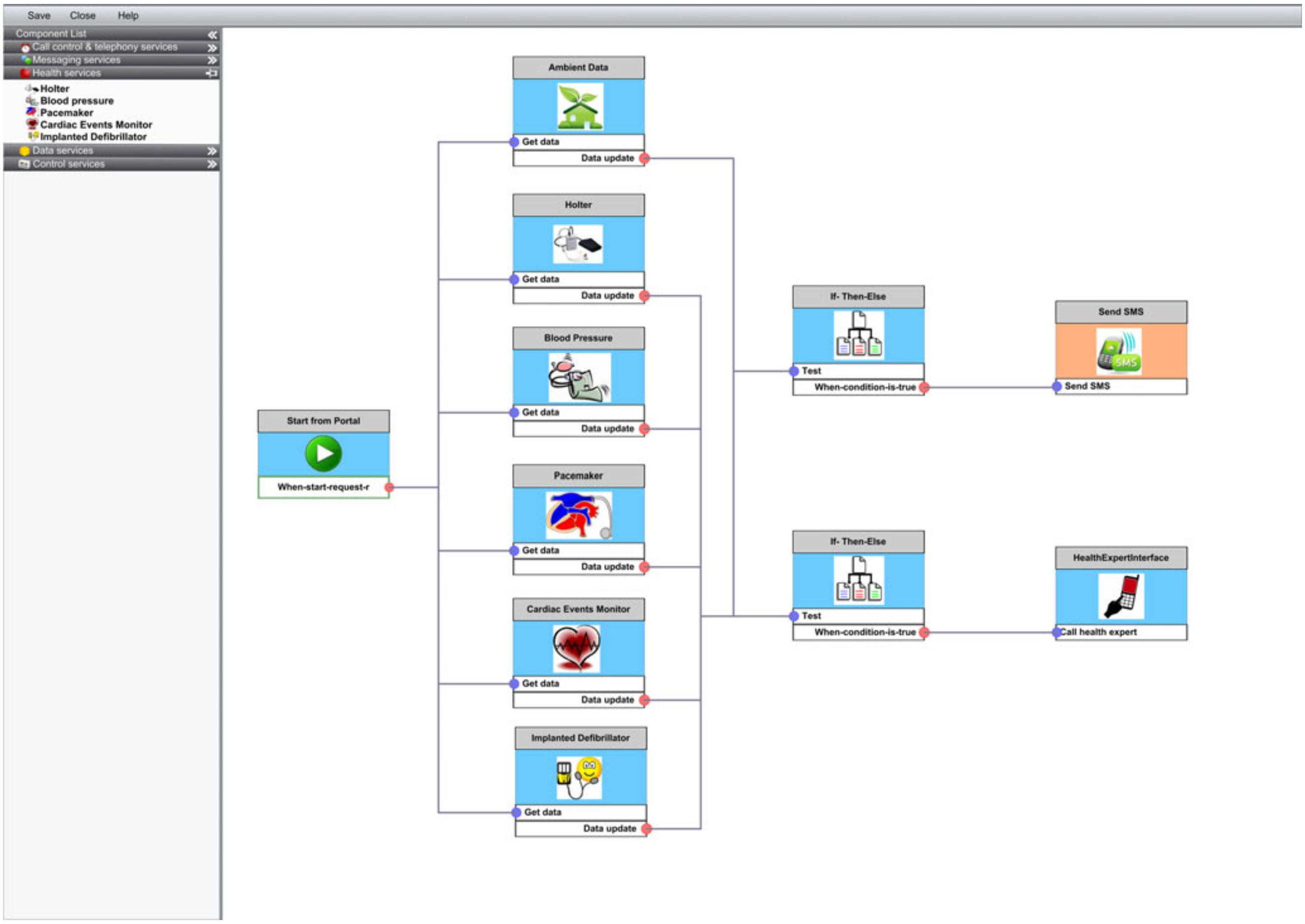

Fig. 4 Heart monitoring scenario workflow in the Service Editor

somehow to the patient so he/she cannot simply forget it at home. The GPS location of the patient and domotic data are fed into two If..Then blocks. One block identifies if the cooker has been on for too much time, identifying a situation where the patient may have forgotten that the cooker was on. The other identifies if the patient has left home and sends an SMS alert, so the carer may monitor the patient's location easily. This service is depicted in Fig. 5.

5.3 User validation

In order to demonstrate the feasibility of the approach suggested, a validation test has been carried out with a group of real end users, with the main aim of studying if the service creation model is usable for non-experts. A total of ten test subjects were employed in this study, including three health professionals and two physically impaired

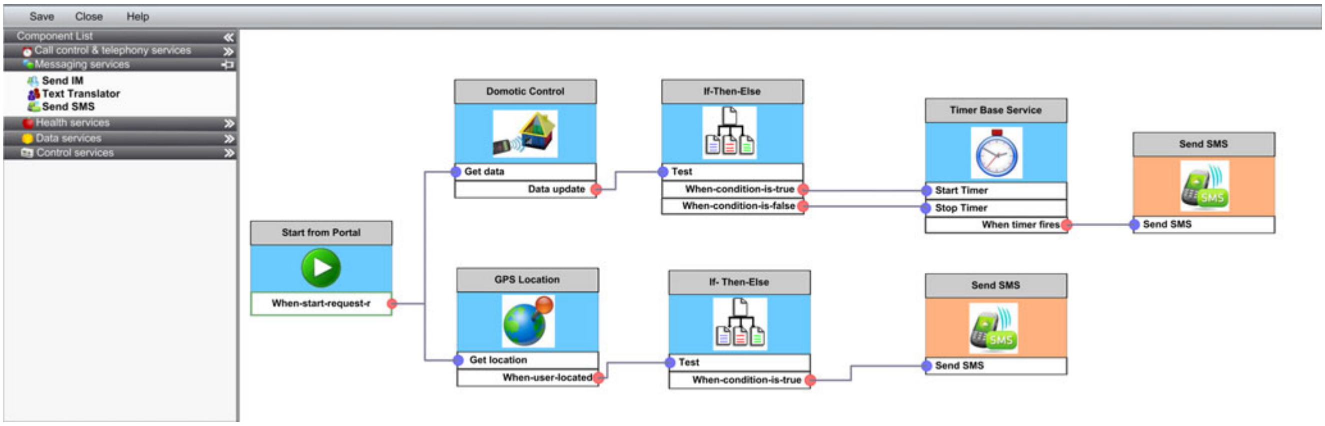

Fig. 5 Alzheimer patient tracking scenario workflow in the Service Editor

individuals. No one had any specific computing skills, besides basic, user/office usage of PCs in some cases.

The test carried out comprised four stages:

- First, the subjects were given a brief generic presentation on the aims of the home care system, not giving any details on how to use it, and then they were left alone to play with it trying to build a very simple service (send a SMS when a timer fires).
- Second, the subjects were given a complete, detailed training session on how to use the system. The composition model was explained in detail, and the tools and editor demoed properly explaining all the abstractions and service creation process step by step.
- Third, the users were left alone again with the system and told to build the use case services detailed in Section 5.2 from a textual description of the scenario.
- Finally, the users were left to play freely with the tools, trying to build services that they would find useful in their real lives.

Feedback was retrieved at the end of each stage by running questionnaires among the test subjects. After analyzing it, the conclusions of this study are as follows:

- The interest attracted is high. After the tests, all subjects stated that the system seems to have a lot of potential and that they would be interested in using it.
- The most recurrent comment about the system is the satisfaction of being completely in control of the service logic and be able to build whatever is required, completely adapted to the specific context of each individual. Health professionals highlighted that doctors and health experts can implement the services directly, avoiding the need of interfacing with a technician.
- Ease of use and user experience at the end of the experiment is regarded as "good", averagely scoring 6.6 out of 10 points. However, while still all test subjects were able to complete the tasks of the first, unassisted stage, the time employed for it was generally long and the frustration scores high. After the half an hour training session of the second stage, operation with the system during the third and fourth stages was much more intuitive, and subjects were satisfied.
- This reveals the need for developing detailed documentation and tutorials, including training videos. Additionally, one recurrent comment among the subjects was the need for a more complete contextual help system.
- Additionally, while the basic service model was regarded as intuitive, the configuration process of some building blocks with several parameters was sometimes frustrating. Feedback shows that building blocks have to be designed carefully, with several users preferring a larger palette with simpler blocks with only one or two actions/events and only one or two parameters. This means for instance having a building block for turning on and off the lights of the bathroom and another one for the kitchen instead of having only one to control all the lights of the house.
- One of the most important concerns expressed by the users in their feedback was the need for a long building block palette. When building personalized services during the fourth stage, they asked for specific blocks that were not present in the system. An important lesson here, which was also a concern before the validation stage, is the need to develop a big building block list and promoting the usage of the building block manager to let third parties introduce new functionalities into the system.
- Health professionals expressed their concerns about robustness and availability, wondering if the system is suitable for life-or-death applications. In its current state, the system is not prepared for those cases, but redundancy and fail-safe features could be further added to it.

6 Discussion

After the validation stage, the most important conclusion that can be extracted is that while it takes some time to learn, the service model proposed is suitable for its usage among non-expert individuals that have at least a basic level of understanding of technology (which are for instance capable of configuring a mobile phone). Still, someone who is not familiar at all with computers and consumer electronics at a basic user level will most probably not be able to use the system.

The literature shows that the Web 2.0 advances are being seized by the home care community to foster new applications, specially taking advantage of the increasing availability of mobile devices, wideband multimedia communications, and easy-to-use Web-based applications [16–18]. However, all this applications are fixed in the sense that they have been programmed ad hoc with a very specific set of functionalities and features in mind. Ku and Huang's [19] for instance is a rather generic framework for home care, but it still does not allow a deep personalization of the functions.

This limitation is being recognized by the community, and as such, there is an emergent trend to promote user-centric development of home care applications. Walderhaug et al. [20] for instance present a unified methodology for rapidly and consistently developing new home care tools embracing the entire community from health professionals to software developers, but it does not eliminate the need for technicians building the applications. The system proposed along this work on the contrary allows health professionals, carers, and patients to take control themselves and design their own services.

The proposed system is therefore especially valuable as an easy-to-use interface to build everyday services. Special purpose tools will still be more robust and trustable than the services developed with the proposed system and therefore more suitable for critical applications. However, it is unlikely that any software brand will release an application for instance to send an alert to the carer whenever an elder patient with dementia opens the oven, and it is in the field of these small, highly personalized (yet still very useful for carers and patients alike) services where the proposed system finds its homage. Therefore, its key feature over other options is its versatility.

7 Conclusions

The home care system presented along this paper demonstrates that the application of user-centricity and Web 2.0 paradigms to the home care environment is not only feasible but also brings a lot of advantages to the field, such as increased flexibility and lower technological and economical entry barriers for potentially interested users.

The system is built over the OPUCE platform [15] which has been modified to fit the home care domain. Besides developing a set of new, home care-specific building blocks, modifications to the original platform include the simplification of the Service Lifecycle Manager module (which was originally developed to deploy and manage software components over complex telco operator infrastructures) and the addition of the service start and stop feature, which was required for a proper operation of the always active, continuously executing monitoring service usually required in home care applications.

With the proposed system, quite a lot of functions are directly available for people searching for simple home care systems, such as automatic SMS-based alerts or location tracking. For this kind of applications, a simple PC and a smartphone are all what is needed to implement the system. While being simple, they are enough to greatly improve the quality of life of old people and other not heavily impaired patients together with their families.

However, the use cases presented demonstrate that the system is also suitable for advanced health applications with just the addition of the appropriate hardware.

In all cases, the system allows people with no specific computer skills, including carers and health professionals, to directly implement the specific services they need, effectively taking away from the chain the requirement for computer experts to implement services.

These kinds of user-centric systems achieve the decoupling of the IT and health planes in the home care environment. The IT professionals just need to provide Web Service interfaces for their hardware and software because home carers and health experts are able to design their highly personalized services on their own.

References

1. Fensli R, Gunnarson E, Gundersen T (2005) A wearable ECG-recording system for continuous arrhythmia monitoring in a wireless tele-home-care situation. Proceedings of the 18th IEEE Symposium on Computer-Based Medical Systems, pp 23–24
2. Paradiso R, Loriga G, Taccini N (2005) A wearable health care system based on knitted integrated sensors. IEEE Trans Inf Technol Biomed 9(3):337–344
3. Hans, M, Graf, G, Schraft, R D (2002) Robotic home assistant Care-O-bot: past–present–future. Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication (ROMAN'02), pp 380–385
4. Williams G, Doughty K, Bradley DA (1998) A systems approach to achieving CarerNet-an Integrated and Intelligent Telecare System. IEEE Trans Inf Technol Biomed 2(1):1–9
5. Bricon-Souf N, Anceaux F, Bennani N, Dufresne E, Watbled L (2005) A distributed coordination platform for homecare: analysis, framework and prototype. Int J Med Info 74(10):809–825
6. Maglaveras N, Koutkias V, Chouvarda I, Goulis DG, Avramides A, Adamidis D, Louridas G, Balas EA (2002) Home care delivery through the mobile telecommunications platform: the Citizen Health System (CHS) perspective. Int J Med Informatics 68 (3):99–111
7. Bellazzi R, Montani S, Riva A, Stefanelli M (2001) Web-based telemedicine systems for home-care: technical issues and experiences. Comput Methods Programs Biomed 64(3):175–187
8. Pellegrino P, Bonino D, Corno F (2006) Domotic house gateway. Proceedings of the 2006 ACM Symposium on Applied Computing, Dijon (France), pp 1915–1920
9. Jimeno R, Salvador Z, Lafuente A, Larrea M Uribarren A (2004) An architecture for the personalized control of domotic resources. Proceedings of the 2nd European Union symposium on Ambient Intelligence, Eindhoven (Netherlands), pp 51–54
10. Stanford V (2002) Using pervasive computing to deliver elder care. IEEE Pervasive Computing 1(1):10–13
11. O'Reilly T (2009) What Is Web 2.0. O'Reilly Network. http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, accessed September 2009
12. Sánchez A, Baladrón C, Aguiar JM, Carro B, Goix LW, Sienel J, Ordás I, Trapero R, Bascuñana A (2009) User-centric service Creation and Execution. At your service: service oriented computing from an EU perspective. MIT Press Series on Information Systems, ISBN 978-0-262-04253-6, June 2009
13. Project OPUCE web site. http://www.opuce.tid.es/, accessed September 2009
14. Cipolla D, Sienel J et al. (2007) Web service based asynchronous service execution environment. Workshop on Telecom Service Oriented Architectures, Springer Lecturer Notes on Computer Science
15. Baladrón C, Aguiar JM, Carro B, Sánchez A (2008) Integrating user-generated content and pervasive communications. IEEE Pervasive Computing 7(4):58–61
16. Botsis T, Demiris G, Pedersen S, Hartvigsen G (2008) Home telecare technologies for the elderly. J Telemed Telecare 14(7):333–337

17. Botsis T, Hartvigsen G (2008) Current status and future perspectives in telecare for elderly people suffering from chronic diseases. J Telemed Telecare 14(4):195–203
18. Smith GE, Lunde AM, Hathaway JC, Vickers KS (2007) Telehealth home monitoring of solitary persons with mild dementia. Am J Alzheimer's Dis Other Dement 22:20–26
19. Ku H, Huang C (2010) Web2OHS: A Web 2.0-based Omni-bearing homecare system. IEEE Trans Inf Technol Biomed 14:(2):224–233
20. Walderhaug S, Stav E, Mikalsen M (2009) Experiences from model-driven development of homecare services: UML profiles and domain models, models in software engineering. Lect Notes Comput Sci 5421:199–212