FACTA UNIVERSITATIS
Series: **Electronics and Energetics** Vol. 28, N^o 3, September 2015, pp. 439 - 456
DOI: 10.2298/FUEE1503439M

AN ARCHITECTURE FOR PERVASIVE HEALTHCARE SYSTEM BASED ON THE IP MULTIMEDIA SUBSYSTEM AND BODY SENSOR NETWORK

Vanja Mišković[1], Djordje Babić[2]

[1]Faculty of information technology, Slobomir P University, Doboj, RS, Bosnia and Herzegovina
[2]School of Computing, Union University Belgrade, Serbia

Abstract. *One of the most promising applications of sensor networks is mobile health monitoring. The key concept of New Generation Networks (NGN) is IP Multimedia Subsystem (IMS). The possibility of using mobile devices as gateways between sensor networks and IMS has led to the development of integrated solutions such as the one proposed in this paper. Event-based SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) architecture is considered as the best solution for IMS based mobile health monitoring. This paper also describes usage of the Session Initiation Protocol (SIP) protocol to communicate with the IMS core, whereas data are transmitted within the body of SIP messages. Thus there is no need for additional transport protocol. Presence Information Data Format (PIDF) is used as data format and data privacy is controlled by XML Configuration Access Protocol (XCAP), which also provides the ability to manage groups of patients.*

Key words: *Body Sensor Networks (BSNs), Context-awareness, IP Multimedia Subsystem (IMS), pervasive healthcare.*

1. INTRODUCTION

The various applications of monitoring patients can be divided among the following categories: prevention, healthcare maintenance and examinations, home care [1]-[4], intelligent hospital [5], [6]; incidence detection, emergency intervention [7]; and pervasive healthcare applications [8]-[13]. Subject of this paper is the pervasive healthcare. It is "healthcare to anyone, anytime, and anywhere by removing locational, time and other restraints while increasing both the coverage and the quality of healthcare" [13].

Nowadays, most of these applications, particularly pervasive ones, use Body Sensor Networks (BSNs). The BSN consists of a set of wearable or implanted sensors, which monitor vital signs or movements of the human body [14]. A modern context-aware applications

Received September 22, 2014; received in revised form December 24, 2014
Corresponding author: Vanja Mišković
Faculty of information technology, Slobomir P University,74000 Doboj, RS, Bosnia and Herzegovina
(e-mail: vanja.elcic@gmail.com)

that enhance user interaction and interpersonal communication [15] use BSNs as sources of raw data. This kind of intelligent applications is essential to monitor human health, where proper assessment of the health condition is extremely important. A certain kind of mobile device gathers this data/polls sensors and it acts as a gateway to the central server or it processes data locally.

There has been significant number of research projects in this area. In [4], a wearable computing platform, called Mithril, with sensors that can continuously monitor the vital signs of users, motoric functions, social interaction, sleep quality, and other health indicators, has been proposed. Mithril is used to study human behavior and to recognize different behaviors for creating context–aware interface with the computer. A project called Ubiquitous Monitoring Environment for Wearable and Implantable Sensors (UbiMon) [8] has for its goal to provide continuous and undisturbed health monitoring system. The system consists of five main components: BSN nodes, local processing unit (LPU), central server (management), patient database and workstation (monitoring). Wireless sensors that can be used here are: ECG sensor, SpO_2 sensor (oximeter), acceleration sensor, etc. In addition, the Compact Flash card was developed for a Personal Digital Assistant (PDA), where all collected sensor data are transmitted through a WiFi/GPRS network for long-term storage and trend analysis.

Further examples of similar projects are MobiHealth [9] and HealthService24 [10] with the difference in the fact that the processing of data is not performed at the LPU, but that the data is forwarded to a remote server where processing is done. BASUMA [11] is also a similar project but it suggests the use of intelligent sensors in the mesh sensor network. In [12], an ad-hoc sensor network for transferring vital signs to the health care providers has been presented as a result of CodeBlue project. There, the use of an adaptive spanning-tree multi-hop routing algorithm has been explored. ActivePal is a commercial example of a system that is used to visualize data from Ambient Assisted Living Services (AALS). It provides a visual representation of data about the activities of the patient during the day. Principles of context awareness are also studied in ActivePal, Tunstall's ADLife [2], and QuietCare [3] system.

The pervasive healthcare system presented here consists of three parts: BSN, data transmission from BSN to observers, and intelligence for context awareness. This contribution presents a novel system architecture for a pervasive healthcare system which is based on IP Multimedia Subsystem (IMS) for data transmission from sensors to the central unit. Thus, we focus here on the data transmission part. The architecture fully corresponds to the existing standard and supported functions of IMS without requiring additional server components. The raw data are collected by using BSN which consists of desired sensors. The collected data is transferred to the central server using the messages of Session Initiation Protocol (SIP). These messages are a basic feature of the SIP extensions, known as SIP Instant Messaging and Presence Leveraging Extensions (SIMPLE). SIMPLE defines the event-based architecture based on SIP PUBLISH/NOTIFY asynchronous messages. It is also possible to protect data privacy using XML Configuration Access Protocol (XCAP). Presence Information Data Format (PIDF) is chosen for data publishing. The presented architecture is effective because it is based on the existing devices and standards which are supported by leading manufacturers

and operators. Furthermore, we have developed an application based on the adopted system architecture. The application has been successfully tested for several use cases.

The outline of this paper is as follows. Section 2 discusses common sensor networks topologies and actual sensor technologies in terms of application for monitoring of patients. Section 3 is devoted to the analysis how the SIMPLE framework as a part of the IP Multimedia Subsystem (IMS) architecture can be used for data transfer in the proposed system. Section 4 explains basics of the context-aware system and describes how to enhance the given solution with characteristics of a context-awareness. In Section 5, we illustrate several use cases of developed application, and we give the content of relevant messages. Finally, Section 6 gives main conclusions.

2. The Analysis of Body Sensor Networks with Respect to Patient Monitoring

The requirements that need to be satisfied by the application significantly affect the topology and sensor network technology. Designers of sensor networks are faced with often conflicting technical challenges so as to meet unique performance characteristics.

The UbiMon project used a star topology [8]. The star topology involves a centralized architecture, where the intelligence of the system is concentrated in the central hub that is superior to the peripheral sensors in terms of resources such as CPU, memory and batteries. The star topology is a common choice because of its simplicity when the scenario does not require direct communication between sensors. On the other hand, Basuma project is an example where the robust mesh topology is used, which relies on a distributed system with peer-to-peer connection, without a central controller [11]. As a result of the mesh topology, if one component breaks down the remaining parts of the system can still work. This approach is desirable when sensors need to communicate with each other and exchange data, perform even less processing and then send the data to a gateway. Of course, this topology is more complex than the star topology and requires smart sensors that consume more energy.

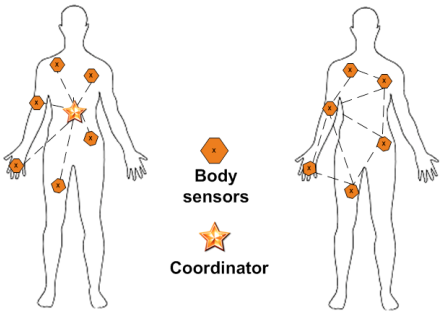

Fig. 1 Illustrations of star (left) and mesh topology (right) for BSN.

In order to facilitate direct communication between sensors and also to reduce the complexity of the mesh network, a compromise solution exists in the form of the cluster-tree topology. If a tree consists of a coordinator node and its associated children nodes

with tree height equal to one, then the cluster-tree topology simplifies to the star topology. There is always a possibility to further spread the tree and to include not only body area sensors but also ambient sensors. Greater complexity is required only by those sensors that provide access to the network. The routing is simple because every node knows its children nodes, and its superior node. The cluster-tree topology always has a bottleneck in the root node. However, if root node is a mobile device which should provide the connection to the IMS and deployment of context-aware applications, then its processing, memory and energy resources are rather sufficient and reliable for transmission of data collected from sensors. Because of its simplicity, low energy consumptions, and simple implementation using mobile devices as coordinator node, we chose cluster-tree topology as a convenient architecture for BSN.

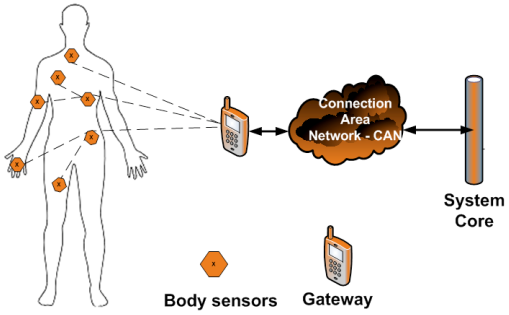

Fig. 2 Illustration of cluster-tree topology.

At the market, there exists number of sensors that can be effectively used for patient monitoring in the proposed system. These sensors use either Bluetooth 4.0 Low Energy standard or ZigBee specification. Bluetooth is a good option to build BSN because of its increasing data rates, low energy and extended battery life. It supports star topology. Devices that have two protocol stacks are called gateways and they collect data from sensors and send them to the storage. For the purposes of medical applications, Health Device Profile (HDP) is developed [16]. Bluetooth SIG determined IEEE 11073-20601 protocol for exchanging data between the HDP layer and the IEEE 11073-104xx Device Specification standard. IEEE 11073-20601 defines a protocol for exchanging data, and IEEE 11073-104xx defines the format of data, including size and encoding. HDP defines two types of devices: sink and source device. The source is a small device that plays the role of a transmitter of medical data. The sink is a functionally rich device that serves as a receiver of medical data, such as smart phone.

ZigBee is another candidate technology for building BSN. ZigBee provides specification for a protocol stack. The first two layers, physical and MAC layers, are taken from the IEEE 802.15.4 standard. In addition, IEEE 802.15.4 creates the foundation for ZigBee upgrading it with the network and application layers to support multi-hop networks. ZigBee multi-hop routing allows wireless environmental sensors scattered all over the house to connect with user's BSN network. ZigBee standard distinguishes three types of devices: ZigBee PAN coordinator, routers and end devices. In this manner, ZigBee standard enables the cluster-

tree topology. ZigBee Health Care (HC) Profile takes most of its attributes of the medical devices from the IEEE 11073 Device Specializations standards family defined for the communication of medical devices [17]. Application scenarios that are supported by the HC profile are: continuous monitoring of the patient, patient's episodic monitoring, alarming scenario, assistance to the elderly and ill, monitoring of sports activities, etc.

The proposed BSN architecture is based on the cluster-tree topology shown in Fig. 2, as already explained above. The basic idea is to relay the system on commercial devices that are already available at the market. As explained above, these devices use either Bluetooth or ZigBee technology. However, at the moment, devices supporting Bluetooth are dominant at the market. The developed application, which is illustrated in Section 5, is focused mainly on the data transmission between a gateway device and registered watchers using IMS. Therefore, the application simulates the BSN and collection of sensor data.

3. Architecture for Data Transmission

As explained above, we decided to use the cluster-tree topology, which consists of one root node which communicates with BSN. The root node can be a typical mobile device (smart phone, tablet) operating under Android, iOS Windows or Symbian OS. In this contribution, we present the system architecture and principles of pervasive health monitoring application which we have developed for Symbian OS using Java ME environment. However, the system architecture and other principles are also applicable to Android, Windows and iOS case as well. The typical mobile device can provide the connection to the IMS and deployment of an application that collects data from sensors according to patient's health profile and send them to the server or watcher. Our intention is to use the existing set of data protocols supported by most of operators. Therefore, IMS represents an architectural framework for providing data transfer from the root node to server and watcher.

As defined in [18], IMS is a global, access-independent and standard-based IP connectivity and service control architecture that enables various types of multimedia services to end-users using common Internet-based protocols. It is based on Session Initiation Protocol (SIP) [19]. IMS primarily deals with issues of heterogeneity of access technologies, addressing schemes, authentication, authorization and accounting (AAA), QoS, security and managing devices mobility. With these characteristics, IMS is a highly scalable framework to carry the information derived from body sensor networks. SIMPLE [20] specification provides a simple and complete event-based architecture without the need for additional application server modules. The basic idea presented here is that sensor data are sent as presence data, and SIMPLE architecture for the transport of presence data uses the body of SIP notify/publish messages.

3.1. Complete architecture of the proposed network

Figure 3 shows the complete architecture of the proposed network for transport of data collected from sensors. The network architecture is very similar to IMS network, as we rely on its services. Here, the role of S-CSCF is to allow the process of authentication, authorization and accounting. It is also responsible to distinguish the initial filter criteria in order to route the

SIP requests to the specific application server, in our case to the Presence Server. Presence Service is responsible for managing information and PUBLISH/SUBSCRIBE/NOTIFY SIP requests. Any information which belongs to the Presence Server is stored on the XCAP server as XML documents. By using XCAP it is possible to change, add, delete elements (nodes) in these XML documents via HTTP methods as explained below. Each type of document has its own folder with XML documents structured in tree form with following leaves:

Presence-rules: They contain the access rights to information which is posted by some presentity; in other words, those are rules which explain who has permission and who does not have permission to access presentity data.

Group List Management: It contains lists of friends who belong to the same group.

Resource List Server: It represents references to the presence data from the same group of friends.

From Fig. 3, it is possible to see that SIP protocol is used for connection between mobile application and P-CSCF. Beside SIP, there is also a direct XCAP connection between mobile application and XCAP server [21]. However, this connection is established if user is successfully registered to the IMS network. Generic Bootstrapping Architecture enables the user to perform authorization process, and to realize a secure connection with any application server in IMS network.

Further in this contribution, we explain in details how IMS entities and services are used for monitoring patients.

3.2. Descriptions of the IMS entities used in the proposed system

A rough outline of the IMS architecture has three main layers [18]. The first layer is the user plane which consists of radio access network, which is also called transport layer. In the middle of the architecture, there is the control layer. Finally, on the top of the architecture is the service layer, as illustrated in Fig. 4. With the service layer separated from the control layer, the service provider may be a third party. Other components depicted in Fig. 4 are described bellow.

Call Session Control Function (CSCF) is a SIP server and the basic entity of the IMS architecture. Most of the SIP signaling is processed through the CSCF. Functionalities that CSCF offers are divided into three clusters: P-CSCF (Proxy-CSCF), I-CSCF (Interrogative-CSCF) and S-CSCF (Serving-CSCF). P-CSCF is the first point of access to the IMS architecture for each user. All SIP signaling traffic to and from the User Equipment (UE) goes through the P-CSCF. I-CSCF is included in the SIP registration process as it joins the corresponding S-CSCF server to the user. S-CSCF is the core of the IMS architecture and provides the logic to invoke and manage the application servers and, if necessary, to deliver the required services. Home Subscriber Server (HSS) is a secure database. It stores subscriber profiles, manages user identities and status (the presence as well as location). S-CSCF, I-CSCF and application servers have the right to access this information. The Application Server is a SIP entity which hosts and implements services. It can have multiple modes, depending on the type of service that is implemented.

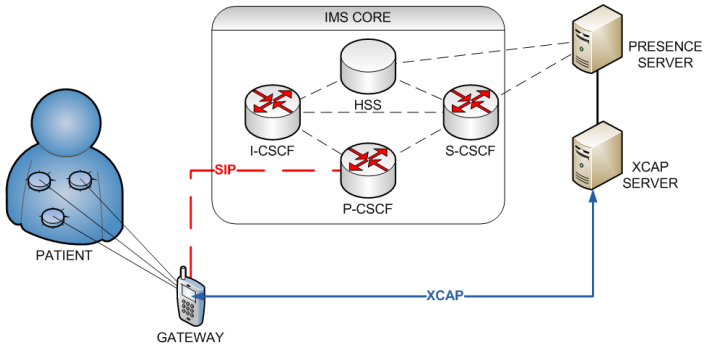

Fig. 3 IMS architecture of the proposed network.

The IMS system users and terminals are uniquely identified. Users have the ability to have multiple profiles to identify services they want to use. These identities can be public or private. IMS uses Authenticating and Key Agreement Protocol (AKA) for authentication, which is based on HTTP Digest access authentication as defined in the IETF [RFC 2617] document. AKA is a challenge-response based mechanism that uses symmetric cryptography, which enables both sides to authenticate and authorize each other.

3.3. Subscribers registration process

IMS registration process begins when the user terminal has to find a P-CSCF address, i. e., the IMS network access point. DHCP/DNS mechanism is commonly used for this purpose. The user sends a SIP REGISTER message to the corresponding P-CSCF. P-CSCF does not know which S-CSCF server is responsible to serve the user. Therefore, P-CSCF asks the I-CSCF for the address of the corresponding S-CSCF. I-CSCF gets this information from a unified database of user profiles HSS by using the Diameter protocol [22], and returns the information to the P-CSCF. After that, P-CSCF forwards user's SIP REGISTER message to the discovered S-CSCF server. S-CSCF takes the user profile from the HSS database using the Diameter protocol. Then S-CSCF answers to the user with the challenge in the Unauthorized SIP message with status code 401. The user authenticates the network and sends a response to the challenge of the S-CSCF server. S-CSCF authenticates users, and retrieves data about the user services from HSS. At the end of this process, S-CSCF sends a confirmation of successful registration to the user in the SIP 200 OK message. P-CSCF and the user terminal can agree about mechanisms of compression, as well as, about using IPSec protocol for greater security of their communication.

3.4. Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE)

SIP's main purpose is to establish, modify and end multimedia sessions [19]. The standard implementation of SIP defines six different methods of SIP requests, which are shown in Table 1. Many of the SIP response codes are inspired by the HTTP protocol. SIP codes are divided into six classes, identified by the first code number as it is shown in Table 2.

Table 1 SIP request methods

Method	Description
INVITE	Session setup
ACK	Acknowledgment of final response
BYE	Session termination
CANCEL	Pending session cancellation
REGISTER	Registration of a user's URI
OPTIONS	Query of options and capabilities

Recently, SIP has been extended to support instant messaging and the presence of IMS service through SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) standard [20]. SIMPLE can easily allow our system expansion and integration with other services and applications aware of the presence (presence-aware) to offer a richer user experience. However, the SIMPLE framework cannot offer audio or video services. These types of services request additional application servers.

Presence Service is built on the top of the event-based SIP architecture. It enables subscribers to know who is available, busy or of no obligation, what the possibilities of their terminals are, and similar status information. Presence Service is one of the standard IMS services.

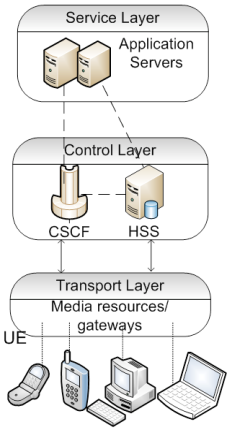

Fig. 4 A simple overview of the IMS architecture

Table 2 Standardized SIP response codes

Class	Description
1xx	Provisional of information
2xx	Success
3xx	Redirection
4xx	Client Error
5xx	Server Error
6xx	Global Failure

Presence Architecture defines several possible roles/entities for each user, as shown in Fig. 5. An entity that provides presence information of subscribers is called presentity, which is short of 'present entity'. The presentity can have several devices connected to it; these devices are identified as Presence User Agents (PUAs). The presentity communicates directly with the Presence Agents (PAs). PA manages the state of various subscribers' events. PA also collects information from PUA through PUBLISH transactions by creating a model of the current state of presentity. It also informs SIP entities called watchers about new publications trough SIP NOTIFY transactions. The observer (also called the watcher) is an entity that requests information about presentity from PA system. This process is done by using SUBSCRIBE requests.

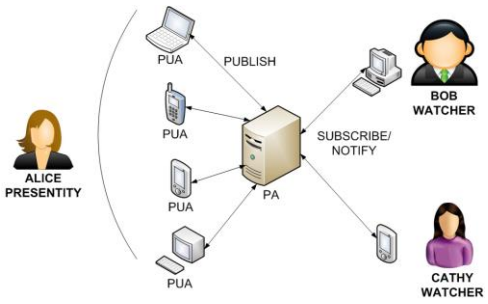

Fig. 5 Entities of presence architecture.

All SUBSCRIBE/NOTIFY transactions, shown in Fig 6, have SIP Event header field that identifies a particular event whether it is a request for information or notification [RFC 3856]. This is the user registration scenario used in our application. It also defines 'presence' event package identified by the value of this header field. SUBSCRIBE may be a short-time operation when it is used to take the current state of presentity information, or it can last for a longer time period to enable asynchronous receiving of notification whenever the published data is changed. Therefore, the watcher must periodically renew SUBSCRIBE before the validity period expires. Current information and duration of SUBSCRIBE request is sent by the PA to the watcher by using the NOTIFY request.

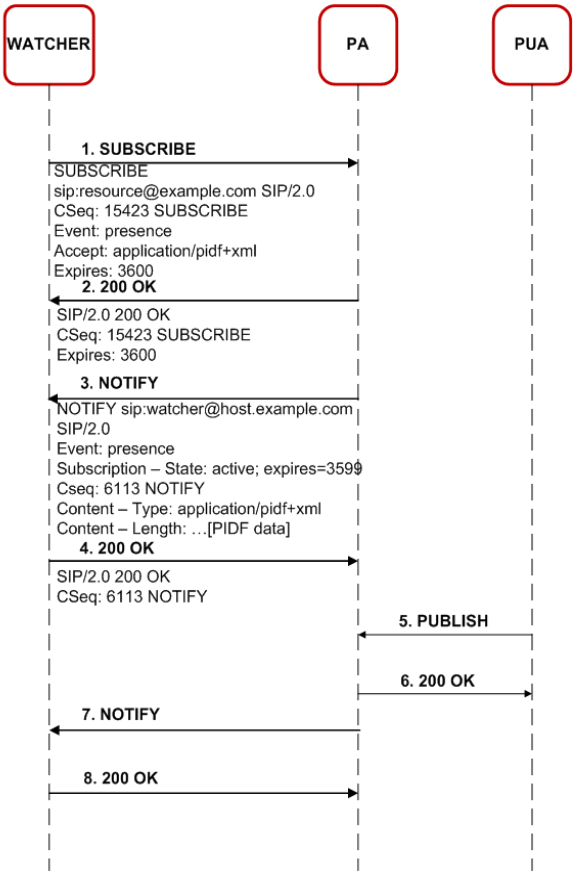

Fig. 6 The use of SUBSCRIBE and NOTIFY SIP messages.

PUA sends PUBLISH request in order to publish new data, as show in Fig 7. The message should include header field identified as Expires that is set to the duration of request. The message also contains new PUA's presence data whose data format is defined in Content-Type header field. Every new successful PUBLISH request obtains a unique identifier from PA, in the form of entity-tag value specified in the SIP-ETag header field. These tags are unique for each user agent. This means that two instances of published presence information can have the same entity tag until they do not belong to the same user agent. If the publication is authorized and successful, the Expires header field of answer will tell about the validity of the publication and also the SIP-ETag will also be specified.

3.5. Presence information data format

The data format used to transfer the presence data is called Presence Information Data Format (PIDF), and its extension is called Rich Presence Extension for PIDF (RPID) [23], [24]. The PIDF defines a basic format for representing presence information from presentity. This format defines a textual note, an indication of availability (open or closed) and a Uniform Resource Identifier (URI) for communication. The RPID includes information about presence information for persons, services, and devices. The examples of data that are supported by the RPID format are: what the person is doing, grouping identifier for a service, when a service or device was last used, type of place a person is in, person's mood, time zone he/she is located in, type of service he/she offers, icon reflecting the presentity's status, etc. Figure 8 illustrates the watcher overview of collected data about patient in our application (note that the sensors are simulated by a simple data generator, the data shown in Fig. 8 can not be interpreted as medical data).

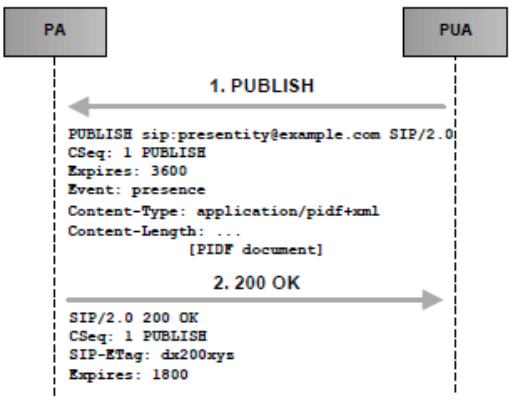

Fig. 7 Publishing data collected from sensors.

Note that a PIDF document and its extension can be used in two different contexts, namely, by the presentity to publish its presence status and by the presence server to notify a certain set of watchers. The presence server may compose, translate, or filter the published presence data, before delivering customized presence information to the watcher. For example, it may perform some of the following operations: merge presence information from multiple presence user

agents; remove whole elements; translate values in elements; or remove information from elements. These mechanisms, that are used to filter calls and other communications to the presentity, can subscribe to the presence information in the same way as a regular watcher. In this way, they can generate automated rules, such as scripts, that govern the actual communications behavior of the presentity. The details are described in the data model document. Since RPID is a PIDF XML document, it also uses the content type application/pidf+xml.

4. CONTEXT-AWARENESS

The sensor data can allow us to create a complete picture of the user's current situation. BSN and implemented special HealthCare Profile gather data which are forwarded to the gateway node. With SIMPLE architecture we get patient's data from gateway into Presence Server. Our architecture resolves all problems of safe access to these data, because it uses secure protocols such as XCAP. XCAP server supports grouping of patients and doctors in corresponding lists. In this way, the health care providers have access to health conditions of all patients in their lists. At the same time, the patients have insight into the list of their care providers with their access rights. Both the patient and the doctor use a mobile application, thus they can be anywhere and anytime, and they can have access to the service.

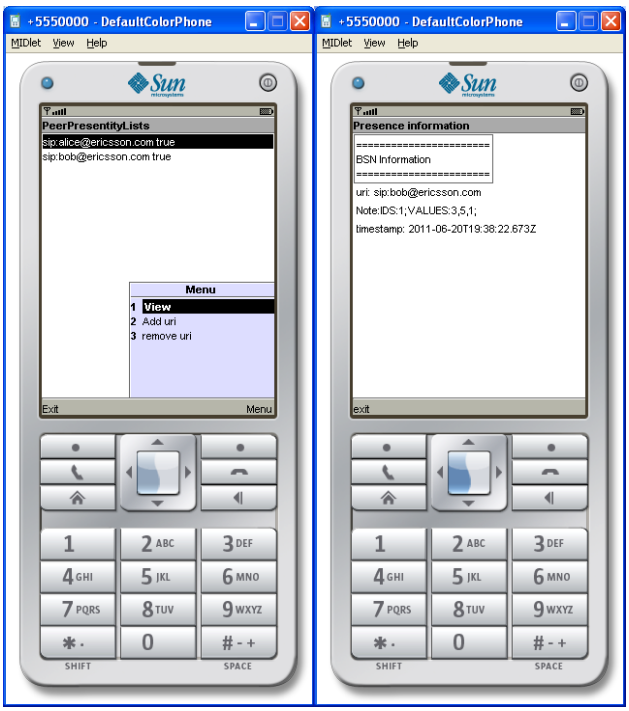

Fig. 8 Presented data collected from sensors at watcher application.

However, the data in Presence Server can give us a true picture of the condition of the patients only with proper reasoning. For example, it is normal that the number of heart beats per minute is about 75. However, when the number of heart beats per minute is greater than 100 then it could be said that this is an abnormal function of the heart, tachycardia. This information is not sufficient to obtain a complete picture and to conclude whether it is an alarming situation or not. It is necessary to have information about the current activities of the person wearing the sensors, because during running and other hard physical activities heart rate per minute is much higher. Accelerometer can give us information about the acceleration of bodies in motion. As we can see, there is a need to have a context-aware subsystem which will analyze collected data and will make decisions.

Common architecture of the context-aware system is shown in Fig. 9 [25]. In general, sensor layer detects and registers new sources of context information, which can result from physical, logical or software sensors. For example, GPS is an example of physical sensor, current browser activities (software) or a combination of sensor information obtained by inference is a logical one. Raw data retrieval layer provides interfaces to return sensor information using abstract functions. Pre-processing/inferencing is referred to quantification and extraction of raw data and handling of conflicting, ambiguous and indefinite information which raises the level of complexity to the next level. This preprocessing/inferencing can be done partly before storing data to the central server or after it. The storage and managing of data are usually separated from application layer. The storage offers a public interface for synchronous or asynchronous access to information.

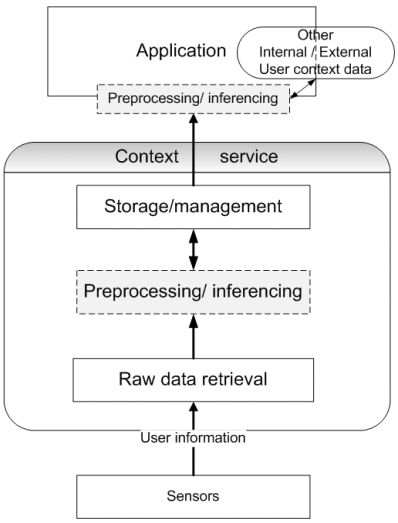

Fig. 9 Layers of common context-aware system.

We do not need to make any changes to the proposed architecture of pervasive healthcare system if pre-processing/inferencing or learning algorithms are in charge of application provider. In service layer application a server would be added, and it would be able to access the presence data. The key point is to store context data for a specific application provider and

application provider determines rules to be respected in the process of reasoning over the set of context data. Also, the mobile phones today have enough processing and memory resources that simple data pre-processing can be carried out on them before they are sent to the system. The implementation of the context awareness in our system is planned in the future.

5 Application Use Cases

As mentioned above, we have developed a mobile application which is based on the proposed system architecture explained above. The system has three main parts: BSN with cluster tree topology, data transmission part based on IMS, and context aware reasoning. The data transmission part is fully implemented in the proposed mobile application. Meanwhile, the BSN is simulated and sensor data is generated. The implementation of context awareness is left for the second stage of our research project.

The mobile application has the following use cases:

1. The registration; Subscription/Unsubscription to the list of the SIP watchers is implicitly added after the registration process, because every patient wants to have access to the list of watchers.
2. The sensors' data are simulated. The presence data are published automatically.
3. Subscription/Unsubscription to the list of the SIP presentities (patients) and revision of the permitted and currently reported data;
4. Determining access rights to the watchers (doctors, friends, family);
5. Review of personal information that is currently published with the ability to manually publish/unpublish them;
6. The deregistration.

Each patient can also be the watcher of his friends/relatives. In the following, the use cases are described verbally with parts of code or through system sequence diagram, and message examples. Section 3.2 explains the IMS entities used in the proposed system: IMS core, Presence server, XCAP server. Any information which belongs to the Presence server is stored on the XCAP server as XML document. Presence server gets data directly from XCAP, and then they are integrated and together called Presence Group Manager (PGM).

Application on a Presence User Agent (PUA) needs to open two ports for two separate connections with system. The first one is for common synchronous SIP communication, such as REGISTER/200 OK, SUBSCRIBE/ 200 OK, PUBLISH/200 OK. The second one is for asynchronous SIP SIMPLE communication based on events. Examples are: Presentity (UN)PUBLISH / Watcher NOTIFY, Watcher (UN)SUBSCRIBE / Presentity NOTIFY. XCAP protocol is based on HTTP protocol, but can also be based on secured HTTPS. This HTTP connection is a direct connection between PUA application and XCAP server and allows manipulation with presence-rules documents.

After successful regular SIP registration, the user is automatically subscribed to the watcher list in order to check the list of assigned watchers and to change their access rights. The process is shown in Fig. 10. If subscription is approved, PGM sends SIP NOTIFY message to asynchronous port with a full watcher list, and SIP 200 OK message to synchronous port as ACK of SIP SUBSCRIBE message. After getting SIP NOTIFY, application sends SIP 200 OK from the port chosen as asynchronous as acknowledgment for using this port for getting information about watchers updates.

In use cases two and five, the same SIP messages are used for sending data, regardless of whether it is automatic or manual transmission. An example of SIP PUBLISH message is given below. This message is provided by UA from SIP port to the IMS core, and further from IMS core to the PGM. In the opposite direction SIP 200 OK message is sent as acknowledgment for the successful data receiving. Data format is application/PDIF + xml, as shown by the Content-Type header field. Within RPDIF are defined other elements about the service, devices or user.

Important header fields of the SIP PUBLISH message are shown in the following example:

```
PUBLISH sip:alice@ericsson.com SIP/2.0
Max-Forwards: 70
Event: presence
CSeq: 312 PUBLISH
Expires: 3600
Content-Length: 645                    ...
Content-Type: application/pidf+xml     ...
```

The following is SIP PUBLISH message with sensor data in the message body:

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns:rpid = "urn:ietf:params:xml:ns:pidf:rpid" ...>
<pdm:person id="p1">
<pdm:note>IDS:1;VALUES:5,1,4;
</pdm:note>
<pdm:timestamp>
2014-06-05T23:02:43.59Z
</pdm:timestamp>
</pdm:person>
</presence>
```

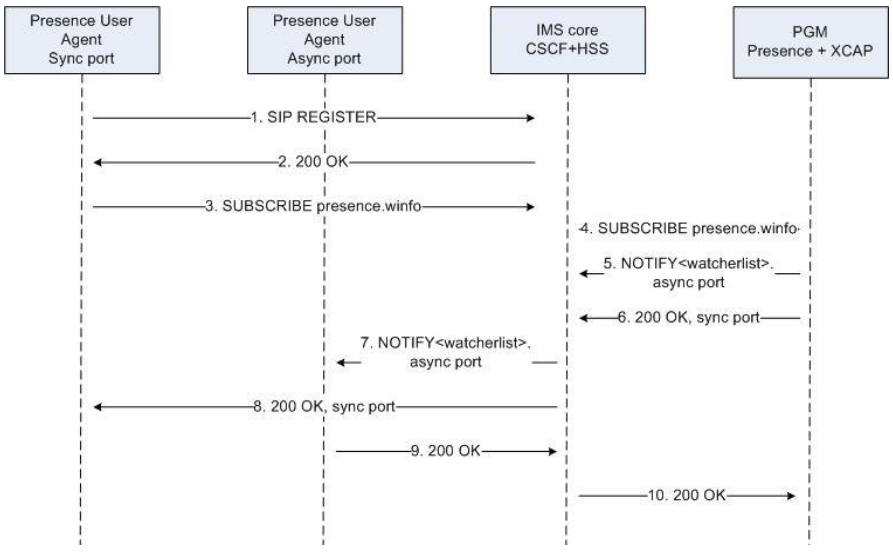

Fig. 10 SIP registration and subscription to the presence watcher list.

SIP UNPUBLISH is a SIP PUBLISH message with the following differences in header fields: Content-Length is 0, Expires is set to 0s and sip-if-match is equal to sip-etag of published data. The following message shows important header fields of the SIP (UN)PUBLISH message:

PUBLISH sip:alice@ericsson.com SIP/2.0
Max-Forwards: 70
Event: presence
CSeq: 317 PUBLISH
Expires: 0
Content-Length: 0
sip-if-match: 0005 ...

In the use case number three, the subscription to the SIP presentity list has same sequence of messages as subscription to the watcher list. But the SIP SUBSCRIBE message has different header fields, such as: Event: presence , Supported: eventlist Accept: multipart/ related,application/ rlmi+xml,application/ cpim-pidf+xml, application/pidf+xml.

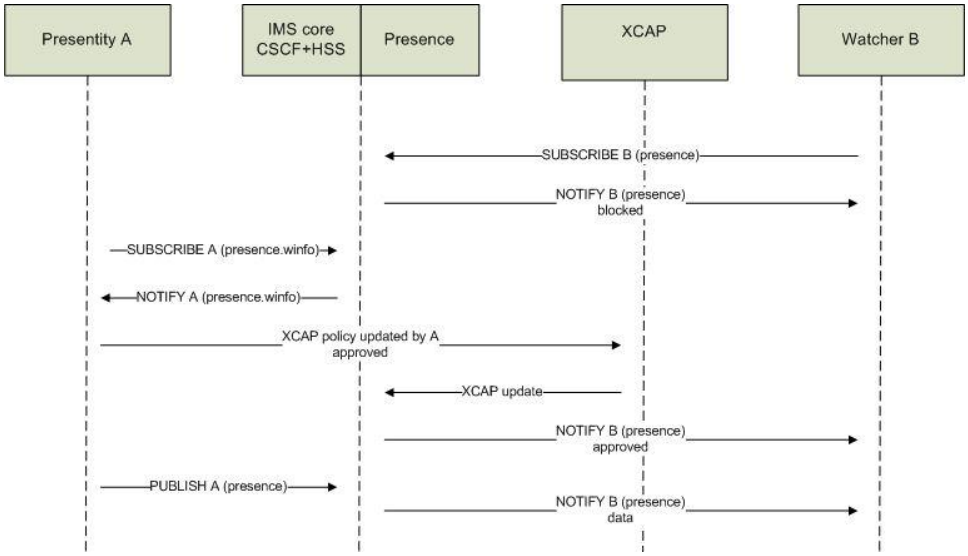

Fig. 11 Events presence.winfo and presence

All changes of lists are done through XCAP protocol, which is based on HTTP protocol and its methods: DELETE, GET, PUT. The HTTP connection is established directly between PUA and XCAP server. The system sequence diagram in Fig.11. illustrates *presence.winfo* and *presence* events. There is also XCAP usage by presentity. It is possible to add (PUT) or remove (DELETE) members of the lists, and presentity can change access rights to the watchers. After subscription watcher waits for confirmation. Presentity can block, polite-block or allow watcher. The action Block tells the server to reject the subscription, placing it in the 'terminated' state. The action Polite-block tells the server to place the subscription into the 'active' state, and to produce a presence document that

indicates that the presentity is unavailable. The action Allow tells the server to place the subscription into the ‘active’ state. Only if watcher is allowed by presentity, the published presence data willtrigger NOTIFY message.

5 CONCLUSIONS

The network architecture based on IMS has been proposed, and we have developed the application for data transmission between patients and watchers. This solution can have many significant purposes and represents a platform for the development of a wide range of applications. We have shown the main elements and use cases of mobile application developed according to the proposed model. The next phase of our research will result in a mobile application for patients’ monitoring for different platforms. We will study applicable context-aware algorithms as well, in order to build a complete system.

Acknowledgement: *This work has been partially supported by the Serbian Ministry of Education and Science under technology development project TR32023 – “Performance Optimization of Energy-efficient Computer and Communications Systems.”*

APPENDIX A

Acronym	Definition
AAA	Authentication Authorization And Accounting
AALS	Ambient Assisted Living Services
ACK	Acknowledgement
AKA	Authenticating and Key Agreement
BASUMA	Body Area System for Ubiquitous Multimedia Applications
BSN	Body Sensor Network
CPU	Central Processing Unit
CSCF	Call Session Control Function
ECG	Electrocardiogram
GPRS	General Packet Radio Service
GPS	Global Positioning System
HC	Health Care
HDP	Health Device Profile
HSS	Home Subscriber Server
HTTP	Hypertext Transfer Protocol
I-CSCF	Interrogative-CSCF
IEEE	Institute Of Electrical And Electronic Engineers
IETF	Internet Engineering Task Force
IMS	IP Multimedia Subsystem
IP	Internet Protocol
IPSec	Internet Protocol Security
LPU	Local Processing Unit
MAC	Media Access Control
ME	Micro Edition (Java)
MIT	Massachusetts Institute Of Technology
NGN	Next Generation Network

Acronym	Definition
OS	Operating System
PA	Presence Agent
PAN	Personal Area Network
P-CSCF	Proxy-CSCF
PDA	Personal Digital Assistant
PIDF	Presence Information Data Format
PGM	Presence Group Manager
PUA	Presence User Agents
QoS	Quality of Service
RFC	Request For Comments
RPID	Rich Presence Extension for PIDF
S-CSCF	Serving-CSCF
SIG	Special Interest Group
SIMPLE	SIP Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
SpO2	Peripheral capillary oxygen saturation
UBIMON	Ubiquitous Monitoring Environment for Wearable and Implantable Sensors
UE	User Equipment
URI	Universal Resource Identifier
Wi-Fi	Wireless Fidelity
XCAP	XML Configuration Access Protocol
XML	Extensible Markup Language

REFERENCES

[1] J. A. Fraile1, J. Bajo and J. M. Corchado, "Context-aware and Home Care: Improving the quality of life for patients living at home", *Actas de los Talleres de las Jornadas de Ingeniería del Software y Bases de Datos*, vol. 3, no. 6, 2009.

[2] *Tunstall*, Tunstall Healthcare (UK) Ltd, [Online]. Available: http://www.tunstall.co.uk (current February 2012)

[3] *QuietCare*, Intel-GE Care Innovations™, [Online]. Available: http://www.careinnovations.com/Products/QuietCare/Default.aspx (current July 2011)

[4] *Project Mithril at MIT University*, [Online]. Available: http://www.media.mit.edu/wearables/mithril/index.html (current October 2004)

[5] J. E. Bardram, "Applications of context-aware computing in hospital work: examples and design principles", In Proc. of the ACM symposium on Applied computing, New York, 2004, pp. 1574-1579.

[6] S. Mitchell, M. D. Spiteri, J. Bates, G. Coulouris, "Context-aware multimedia computing in the intelligent hospital", In Proc. of the 9th workshop on ACM SIGOPS European workshop, pp. 13-18, 2000.

[7] P. Hu, J. Indulska, R. Robinson, "An Autonomic Context Management System for Pervasive Computing", In Proc. 6th of the Annual IEEE International Conference on Pervasive Computing and Communications, March 2008.

[8] *Project Ubimon at Imperial College London*, [Online]. Available: http://www.doc.ic.ac.uk/vip/ubimon/home/index.html (current January 2005)

[9] *MobiHealth funded by the European Commission*, [Online]. Available: http://www.mobihealth.org (current May 2004)

[10] *HealthService24*, Ericsson Enterprise AB – project coordinator, [Online]. Available: http://www.healthservice24.com (current December 2006)

[11] Basuma funded by the German Federal Ministry for economics and labor (BMWA), [Online]. Available: http://www.basuma.de (current January 2006)

[12] *Project CodeBlue at Hardvard University*, [Online]. Available: http://fiji.eecs.harvard.edu/CodeBlue (current March 2007)

[13] U. Varshney, "Pervasive healthcare and wireless health monitoring", *Journal: Mobile Networks and Applications, Springer-Verlag*, New York, vol. 12, pp. 113-127, March 2007.
[14] M. Carmen Domingo, "A Context-Aware Service Architecture for the Integration of Body Sensor Networks and Social Networks through the IP Multimedia Subsystem", *IEEE Communications Magazine*, vol. 50, pp. 102-108, January 2011.
[15] F. Toutain, A. Bouabdallah, R.. Zemek, C. Daloz, "Interpersonal Context-Aware Communication Services", *IEEE Communications Magazine*, vol. 50, pp. 68-74, January 2011.
[16] R. Latuske, ARS Software GmbH, "Bluetooth Health Device Profile (HDP)", München, 2009. [Online]. Available: http://www.ars2000 .com /Bluetooth_HDP.pdf (current January 2012)
[17] *ZigBee Aliance*, "ZigBee Wireless Sensor Applications for Health, Wellness and Fitness", March 2009. [Online]. Available: https://docs.zigbee.org/zigbee-docs/dcn/09-4962.pdf (current February 2012)
[18] M. Poikselk¨a, G. Mayer, "*The IMS: IP Multimedia Concepts and Services (3rd ed.)* ": The Complete Book, John Wiley & Sons, 2009.
[19] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, M. Handley,E. Schooler, "Session Initiation Protocol (SIP)", RFC 3261, [Online]. Available: http://www.ietf.org/rfc/rfc3261.txt (current June 2002)
[20] M. Day, J. Rosenberg, H. Sugano, "A Model for Presence and Instant Messaging", RFC 2778, [Online]. Available: http://www.ietf.org /rfc/rfc2778.txt (current February 2000)
[21] J. Rosenberg, "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP) ", RFC 4825, [Online]. Available: http://www.rfc-editor.org/rfc/rfc4825.txt (current May 2007)
[22] P. Calhoun,J. Loughney, E. Guttman, G. Zorn, J. Arkko, "Diameter Base Protocol", RFC 3588, [Online]. Avaliable: http://tools.ietf. org/html/rfc3588 (current September 2003)
[23] H. Sugano, S. Fujimoto, G. Klyne, A. Bateman, W. Carr, J. Peterson, "Presence Information Data Format (PIDF)", "RFC 3863, [Online]. Available: http://www.faqs.org/rfcs/rfc3863.html (current August 2004)
[24] H. Schulzrinne, V. Gurbani, P. Kyzivat, J. Rosenberg, "RPID: Rich Presence Extensions to the Presence Information Data Format (PIDF)", RFC 4480, [Online]. Available: http://tools.ietf. org/html/rfc4480 (current July 2006)
[25] P. Gutheim, "An Ontology – Based Context Inference Service for Mobile Applications in Next – Generation Network", *IEEE Communications Magazine*, vol. 50, pp. 60-66, January 2011.