

Engineering Large-Scale Body Area Networks Applications

Giancarlo Fortino, Raffaele Gravina, Antonio Guerrieri
DIMES – University of Calabria
Via P. Bucci, cubo 41C
87036 Rende (CS), Italy
g.fortino@unical.it, rgravina@deis.unical.it,
aguerrieri@deis.unical.it

Giuseppe Di Fatta
School of Systems Engineering
University of Reading
Reading, RG6 6AX, UK
G.DiFatta@reading.ac.uk

ABSTRACT

This paper presents an approach to design and deploy large-scale applications for body area networks (BANs). The approach is based on BodyCloud, which is a Cloud-based multi-tier application-level architecture. BodyCloud integrates a Cloud computing platform and the SPINE BAN middleware. It specifically provides programming abstractions, such as group, modality, workflow and view, which support the rapid and effective development of community BAN applications. This work provides an overview of the general architecture of BodyCloud and discusses several large-scale application services that can be deployed on BodyCloud.

Categories and Subject Descriptors

D.2.6 [Software Engineering]: Programming Environments—Integrated environments; C.3 [Computer-communication Networks]: Special-purpose and Application-based Systems—Real-time and embedded systems.

General Terms

Design, Experimentation, Languages.

Keywords

Body Area Networks, Cloud Computing, SaaS, Wearable Computing, Large-Scale Distributed Applications.

1. INTRODUCTION

Wireless Sensor Networks (WSNs) [1] are used in pervasive systems aimed at the collection, processing and archiving of data generated by sensing natural and/or artificial phenomena. WSNs specifically consist of spatially distributed sensor nodes and are typically coordinated by one or more base stations. Sensors cooperatively monitor physical, environmental, or human parameters such as temperature, sound, vibration, pressure, pollution, motion, heart rate, and blood pressure.

Body Sensor Networks or Body Area Networks (BANs) [2] are a class of WSNs that is specifically applied to the monitoring of humans. BANs represent an emerging technological platform for many and diversified human-centered applications, ranging from medical to sports performance monitoring, gaming, mass event management and social networking.

BANs can currently be formed by a wide variety of wearable sensor nodes providing important physiological measurements

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

BODYNETS 2013, September 30-October 02, Boston, United States

Copyright © 2013 ICST 978-1-936968-89-3

DOI 10.4108/icst.bodynets.2013.253721

such as body temperature, heart rate, limb motions, blood glucose, blood oxygen saturation, electrocardiogram (ECG), electromyography (EMG), galvanic skin response (GSR), electroencephalogram (EEG), photoplethysmography (PPG). Data streams from body sensors are processed to provide higher level information for complex tasks, such as activity recognition, emotion detection, sickness prevention and monitoring.

The effective and efficient management of a large number of cooperative and non-cooperative BANs is a critical task in supporting a wide range of pervasive applications for communities of users. In fact, the huge amount of data that networks of BANs may generate, requires a scalable and flexible infrastructure for the collection, storage and processing, including the online and offline analysis of data streams.

The management of networks of BANs and their gathered data cannot be autonomously accomplished with their limited resources, BodyCloud tackles the problem by exploiting a Cloud computing infrastructure and providing an integrated platform, namely a Cloud-enabled BAN infrastructure, that offers:

- capabilities of using heterogeneous sensors through mobile devices acting as gateways,
- scalability of processing power for different kinds of analysis,
- scalability of data stream storage,
- ubiquitous and global access to the processing and storage infrastructure,
- easy sharing of results and
- pay-as-you-go pricing for using BAN services.

This paper presents an approach for the rapid prototyping of large-scale BAN applications that is based on BodyCloud [3, 4]. BodyCloud is designed as a SaaS architecture to support the storage and management of sensor data streams and the processing and analysis of the stored data using software services hosted in the Cloud. In particular, BodyCloud endeavors to support several cross-disciplinary applications and specialized processing tasks. It enables large-scale data sharing and collaborations among users and applications in the Cloud, and delivers Cloud services via sensor-rich mobile devices. BodyCloud also offers decision support services to take further actions based on the analyzed BSN data.

The rest of the paper is structured as follows. Section 2 discusses related work about the integration of WSNs (and specifically BANs) and Cloud Computing. Section 3 provides an overview of the BodyCloud architecture. Section 4 presents the process of engineering BodyCloud application services. Finally, in Section 5 conclusive remarks are drawn and directions of future work are briefly outlined.

2. RELATED WORK

Recently researchers have investigated the integration of WSNs with a large-scale distributed computing infrastructure [5]. Examples include an integration architecture of Cloud computing and WSNs [6], Sensor-Web [7], the Sensor-Cloud infrastructure [8], SAaaS [9], Cloud-enabled wearable system [10], and Cloud-assisted ECG service [11].

In [6], a SaaS architecture for sensor network analytical services is proposed. It is implemented atop a PaaS layer (e.g. Google App Engine, Microsoft Azure) and is organized in three layers: (i) sensor data management, which collects sensor data streams coming from the WSN gateway; (ii) run time for filter analysis, which supports the execution of processing workflows for sensor data according to the pipe-and-filter paradigm; (iii) filter management, visualization and notification, which are three components that respectively allow for the definition and management of the processing filter chain, for the visualization of analyzed data, and for the notification of events.

Authors in [7] propose the Open Sensor Web Architecture (OSWA). OSWA is an OGC (Open Geospatial Consortium) Sensor Web Enablement standard-compliant software infrastructure for providing service-oriented based access and management/ integration of sensors. OSWA also integrates distributed computing platforms such as SOA and Grid computing. OSWA is designed around the conventional Grid layers: Fabric, Services, Development and Application. Specifically, the OSWA-based platform provides a number of sensor services such as sensor notification, collection and observation; data collection, aggregation and archive; sensor coordination and data processing; faulty sensor data correction and management; sensor configuration and directory service.

In [8], authors propose a new infrastructure, called Sensor-Cloud Infrastructure, which can manage physical sensors on IT infrastructure. The Sensor-Cloud Infrastructure virtualizes a physical sensor as a virtual sensor on the Cloud computing platform. Dynamically grouped virtual sensors can be automatically provisioned when the users require them through a portal server interacting with the provisioning server, performing resource management, and a monitoring server, monitoring real/virtual sensors.

SAaaS [9] is a Cloud-enabled SaaS architecture aiming at the management of wireless sensor and actuator networks (WSAN). SAaaS is a software stack that implements the following main functionalities: involvement of (W)SNs, smartphones or other devices endowed with sensors and/or actuators, and their enablement for interoperation and management in a Cloud environment; exploitation of Volunteer-based methods for node involvement; functions and interfaces for federating SAaaS Clouds, either volunteer-based or commercial/institutional.

Specific research efforts involving integration between BANs and Cloud computing include a Cloud-enabled system based on an integrative gaming paradigm [10], a Cloud-based ECG data analysis service [11], and BodyCloud [3, 4].

In [10] the author present a Cloud-enabled system based on an integrative gaming paradigm. The system is designed to integrate multiple activities that involve physical exercises and cognitive skills by means of a game-based storyline. The Cloud-enabled system is configurable and allows researchers to easily create new games that can be driven by different activity types. Using the Cloud-based system, data are archived in an on-line datastore that

is easily accessible by a website. This enables clinicians to remotely access the stored data and to easily integrate such data into electronic medical records. Moreover, Cloud-based tools for reporting and data analysis tools allow for effective analysis of data and enable integration of the physiological information into biosignatures and clinical repositories. The system is meant to provide clinicians with continuous information on their patients. This is an important feature of the system proposed in [10]. However, the system is special-purpose and its architecture is not reusable and extensible as basis for the development of different Cloud-enabled BSN applications.

A more general approach is proposed by Pandey et al. [11] that report the development of an autonomic Cloud environment for hosting an ECG data analysis service. In particular, the service collects people's health data, stores them into a Cloud-based information repository and facilitates the analysis on the data using software services hosted in the Cloud. A prototype system is used as an experimental testbed on a specific use case, namely, the collection of electrocardiogram (ECG) data obtained at real-time from volunteers to perform basic ECG beat analysis. The ECG software is hosted as a web-service such that any client-side implementation can simply call the underlying functions (analyze, upload data, etc.) without having to go through the complexities of the underlying application. The PaaS layer controls the execution of the software using three major components: (i) Container scaling manager, (ii) Workflow Engine, and (iii) Aneka Cloud middleware.

Although the proposed approach is more general than the one presented in [10] as it proposes a layered software architecture that could be extended to accommodate for different BSN-oriented application services, it is still not flexible enough for rapid prototyping of community BSN applications. Moreover, it is based on Aneka that is a proprietary Cloud middleware.

While such systems are special-purpose or focus on specific aspects of assisted living monitoring (physical activities, ECG, etc), our BodyCloud approach (see Section 3) is different in that it provides real-time and off-line BSN data stream processing and analysis to support many BSN applications, using a general-purpose SaaS approach based on a widely available PaaS infrastructure, the Google App Engine.

3. AN OVERVIEW OF BODYCLOUD

BodyCloud is an architecture for the integration of BANs and a Cloud PaaS infrastructure. The architecture, shown in Figure 1, consists of four main subsystems (or sides):

- *Body*, which is the system side that monitors the assisted living by means of a BAN and sends the collected data to the Cloud through an Android-enabled mobile device. In particular, data acquisition is currently based on Android-SPINE, the Android version of the SPINE middleware [12]. It allows Android-enabled smartphones and tablets to be used as coordinator of the BAN. Data collected through Android-SPINE are then streamed up to the Cloud-side by using the real-time data feed modality (see below). In Android-SPINE communication of wearable sensors with the BAN coordinator is based on Bluetooth. The following functionalities are provided by the application-level SPINE protocol [13]: sensor discovery, sensor configuration, in-node processing, BSN activation/deactivation, data collection, and logging. Finally, the current prototype implementation fully supports the Bluetooth-based Shimmer sensor nodes.

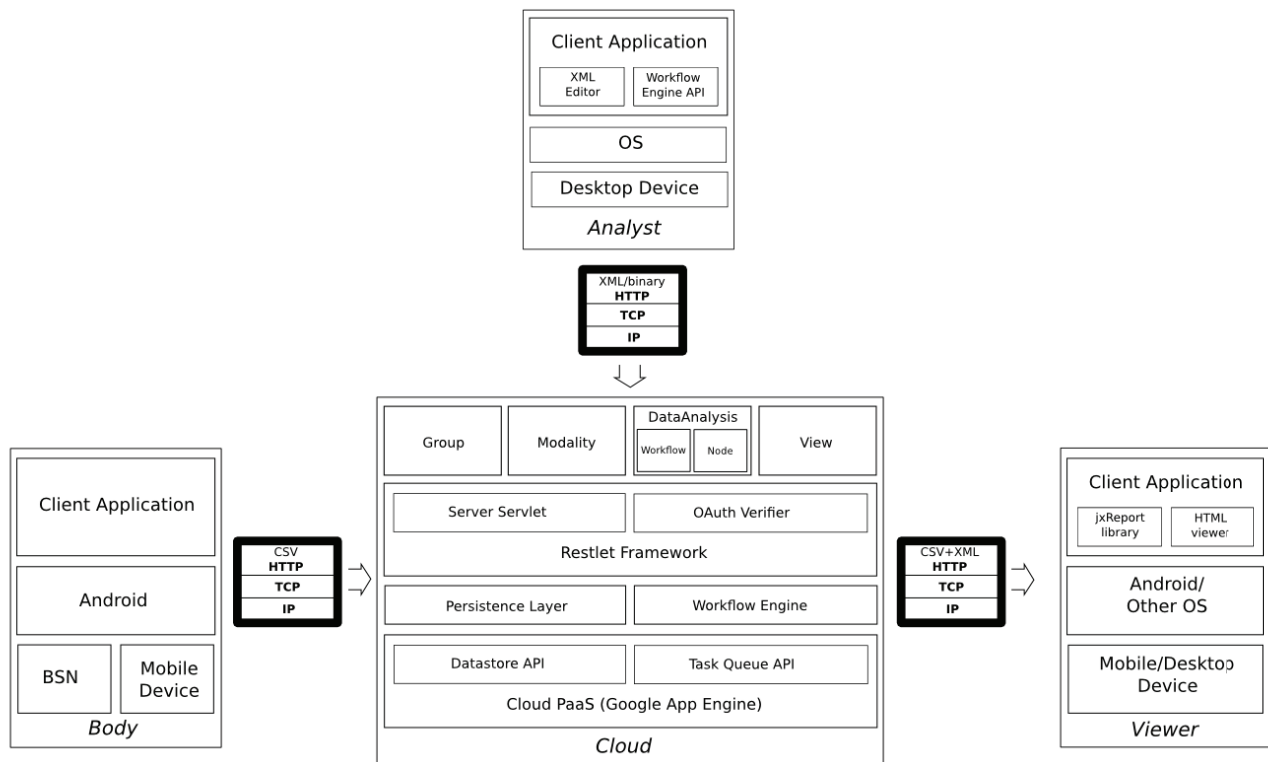


Figure 1. BodyCloud architecture.

- *Cloud*, which is the system side providing full support for specific applications through data collection, processing/analysis and visualization. In particular, each specific application can be defined through the following programming abstractions: *Group*, *Modality*, *Workflow/Node*, and *View*. Such abstractions are supported by a RESTful web service (Server Servlet), implemented using the Restlet Framework, making the interaction with the Cloud-side fully based on HTTP methods (get, put, post, delete). The interactions are authenticated by the OAuth Verifier component based on OAuth 2.0. The Cloud-side is supported by the Google App Engine PaaS [14] that provides the Datastore API, atop which the Persistence Layer managing the collected BAN data is built, and the Task Queue API, which enables asynchronous execution of tasks triggered by requests.

In the following, the Web-based programming abstractions are described in more detail:

- o *Group* is a HTTP resource formalizing a specific application which manipulates a specific BAN data source. In particular, it is composed of three correlated sub resources: *Collector*, *Data*, and *Contributor*. *Collector* is intended to gather BAN data which comply to the same data specification. *Data* represents the actual data collected by the Group. In particular, *Data* is grouped on a per user basis. Moreover, *Data* can be encoded in different formats such as CSV, Arff, and JSON. The *Contributor* is a sub-resource that contains the users who uploaded data to the Group.
- o *Modality* is a HTTP resource formalizing a specific interaction between Body, Cloud and Viewer, within a Group. In particular, it encodes a Body-Cloud or a Viewer-Cloud interaction and is intended to be interpreted and executed by a client application. A Modality models a specific service, such as BAN data feeds, data analysis tasks, single-user or community applications. A Modality defines the specifications of input and output data formats, protocols for data transfer, the flow of processing tasks to transform input data into output data and the specifications of output data visualization. Finally, a Modality can be activated individually and in groups to provide a service to the user.
- o *Workflow* is a HTTP resource formalizing a data-flow process that analyzes input data to generate output data. A Workflow is composed of one or more Nodes organized in a directed acyclic graph. Nodes represent specific algorithms, which can be developed as Java code according to the Workflow Engine library (see Figure 1), and links between nodes are data flows. Once implemented, a Node can be packed within a jar file and uploaded to the Cloud-side where it can be exploited in different workflows.
- o *View* is a HTTP resource formalizing the visualization layout of the output data for Viewers.

- *Analyst*, which is the side of the system that supports the development of new BodyCloud application services. In particular, users can create new BodyCloud services by defining the aforementioned entities: groups, modalities, workflows, and views. Each entity can be created with an HTTP PUT request to the corresponding Cloud-side resource, thus requiring only a simple HTTP client as Analyst-side supporting application. As the workflow requires new nodes to be developed, the Analyst-side also requires an appropriate development environment. Once developed, new nodes are also uploaded with a HTTP PUT request to the corresponding Cloud-side resource. A predefined set of nodes is typically available, depending on the adopted implementation of the Workflow Engine. In Section 4 the process of defining a new BodyCloud service is presented and exemplified for several case studies.

- *Viewer*, which is the system side able to visualize the output produced by the data analysis through advanced graphical reporting facilities. The graphical view is automatically generated by applying the View specification to the data. Specifically, as part of the current prototype, a Java library, named jxReport, has been developed and integrated into the client application. jxReport provides the functionalities to generate HTML reports from an XML schema and a data model and provides the desirable separation between the data model and the view. During the report generation jxReport reads the model, from a CSV file for example, and draws the graphic elements specified in the XML document based on the model data. The jxReport library can be used in any Java-based environment (e.g. mobile or desktop).

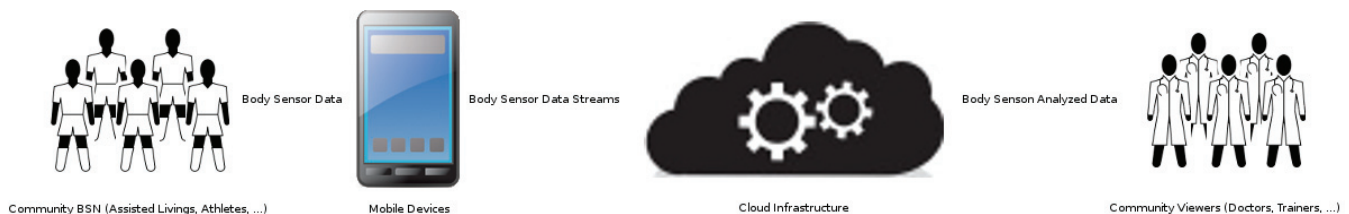


Figure 2. Reference Schema of Cloud-based BAN services.

4. LARGE-SCALE BAN SERVICES

BodyCloud supports an effective approach for the rapid prototyping of large-scale applications based on BANs according to the schema shown in Figure 2.

A BAN service definition based on the BodyCloud approach can be carried out according to the following five phases:

1. *Development and upload of the processing algorithms*: design, implementation and upload of any custom processing/analysis algorithms as nodes, if required. All uploaded nodes are stored into the Cloud-side and can be exploited by any BodyCloud user.
2. *Definition of the Data Source*: definition of a Group containing the specification of data that can be collected by the BAN and then processed by the algorithms defined in phase 1.
3. *Definition of the Workflow*: definition of the data analysis process, through the combination of the uploaded nodes and their static parameters into a workflow. The starting node of the workflow should read the input data from the Data Source.
4. *Definition of the View*: definition of one or more graphical formats (or views) of the data output produced by the workflow.
5. *Definition of the Modalities*: at least a Body-side specific modality and a Viewer-side specific one must be defined. The Body-side modality should have an input data specification similar to the Data Source, an action that will upload the data to the group defined in phase 2 and no output specification. The Viewer-side modality should perform the workflow execution as action, the parameters of which must be defined accordingly to the node documentation. Its output specification must match with the workflow output and contains the correlated reference to the view.

In the following subsections, we provide three examples of large-scale BAN services that can be easily built atop BodyCloud.

4.1 ECGaaS

The ECG as a Service (ECGaaS) has been developed with the BodyCloud approach and allows monitoring (collect, process, store, analyze and visualize) the ECG data coming from individuals or group of people (e.g. assisted livings, athletes, emergency teams). The electrocardiogram (ECG) is the standard method for measuring the electrical and functional activity of the heart. Traditionally, the ECG is used to diagnose cardiovascular diseases and cardiac abnormalities [15]. In particular, in the case study, the ECG signal is captured by the Body-side, through a Shimmer sensor node equipped with the ECG board, and sent to the Cloud-side in which the R-R intervals and heart rate (HR) [16] are extracted through QRS-complex detector algorithms [17] deployed as nodes in the BodyCloud system.

The specific entities (group, modality, workflow/node, view) defining the ECGaaS are:

- The ECGMonitoring group, which represents the group of monitored users.
- The DataFeed, SingleAnalysis, and GroupAnalysis modalities. The former allows to transmit ECG data from the Body-side onto the Cloud-side, whereas the second and third respectively perform single and group analysis of the ECG data, specifically the extraction of the R-R signals (from which the heart rate can also be straightforwardly computed). The specification of the DataFeed modality is reported in Figure 3 whereas the GroupAnalysis modality is portrayed in Figure 5. The DataFeed, which collects sampled data from the Body-side and sends them to the Cloud-side, is performed each 60s. The GroupAnalysis gets all the contributors (i.e. the identifiers of the involving participants)

and executes the workflow on their data so providing the tachogram of all participants.

- The EcgToRR workflow (see Figure 4), which models a workflow composed of two sequential nodes able to read the ECG collected user data through the data reader node and extract the R-R signal from the ECG data through the RR node.
- The Tachogram View, which is the graphical format through which the R-R signal will be rendered at the Viewer-side.

```
<modality>
  <inputSpecification>
    <data>
      <name>ECGShimmerSample </name>
      <type>INTEGER</type>
      <source>ECGShimmerSensor</source>
    </data>
  </inputSpecification>
  <init-action>
    <uri>/group/ecg-monitoring/data</uri>
    <method>DELETE</method>
  </init-action>
  <action>
    <uri>/group/ecg-monitoring/data</uri>
    <method>PUT</method>
    <repeat>true</repeat>
    <trigger after="60"/>
  </action>
</modality>
```

Figure 3. ECGMonitoring DataFeed modality.

```
<workflow>
  <node>
    <type>UserDataReader</type>
  </node>
  <node>
    <type>RR</type>
  </node>
</workflow>
```

Figure 4. EcgToRR workflow.

```
<modality>
  <init-action>
    <uri>/group/ecg-monitoring/contributors</uri>
    <method>GET</method>
  </init-action>
  <action>
    <uri>/engine/workflow/ecg</uri>
    <method>POST</method>
    <parameter>
      <name>sourceUser</name>
      <reference xpath="//users/user"/ type="MAP">
    </parameter>
    <parameter>
      <name>sourceGroup</name>
      <value>ecg-monitoring</value>
    </parameter>
    <repeat>false</repeat>
  </action>
  <outputSpecification>
    <data>
      <name>rr</name>
      <type>DOUBLE</type>
    </data>
    <view>/view/tachogram.xml</view>
  </outputSpecification>
</modality>
```

Figure 5. ECGMonitoring GroupAnalysis modality.

4.2 Fear Detection

Apart from its fundamental use in health-care for the diagnosis of assisted livings' cardiac status, the ECG signal can be exploited to detect emotions, as it is very reactive to physiological responses due to emotions and other external factors. The advantage of using the ECG signal for detecting basic emotions is that a person can be monitored using non-invasive wearable cardiac sensors. In contrast, emotion recognition methods based on facial recognition are more invasive as they require the placement of electrodes and cameras to detect subtle changes in the person's face. In particular, basic fear status could be detected by analyzing the ECG signal. Fear is the physiological response when a person is in danger. The basic cardiac physiological response that generates the state of fear is the Cardiac Defense Response (CDR) [17]. On the basis of an algorithm for the CDR detection, as the one proposed in [17], a basic fear detection service (FEARaaS) can be easily developed atop BodyCloud, by also reusing some system components and entities defined for the ECGaaS.

The specific entities (group, modality, workflow/node, view) defining the FEARaaS are:

- The FearDetection group, which represents the group of monitored users.
- The ECGDataFeed (see Figure 3), SingleCDRAnalysis (see Figure 6), and GroupFearDetectionAnalysis (see Figure 7) modalities. ECGDataFeed is the same modality as in the ECGaaS (see previous subsection). SingleCDRAnalysis performs the CDR detection on a single subject and provides true if the CDR is detected, false otherwise. GroupFearDetectionAnalysis performs the CDR detection on a group of assisted livings and provides a positive result if the number of people having a CDR in a given time period overtakes a given threshold.
- The SingleCDR workflow (see Figure 8), which models a three sequential nodes workflow able to read the ECG collected user data through the data reader node, extract the R-R signal from the ECG data through the RR node, and apply on the R-R signal the CDR detection algorithm.
- The GroupCDR workflows, which is based on the SingleCDR workflow to which the node processing the group fear detection algorithm is added.
- The CDR View, which is a simple graphical format through which the True/False results provided by the (single or group) CDR detection algorithm will be presented at the Viewer-side.

```
<modality>
  <init-action>
    <uri>/group/fear-detection/contributors</uri>
    <method>GET</method>
  </init-action>
  <action>
    <uri>/engine/workflow/cdr</uri>
    <method>POST</method>
    <parameter>
      <name>sourceUser</name>
      <reference xpath="//users/user"/>
    </parameter>
    <parameter>
      <name>sourceGroup</name>
      <value>cdr-monitoring</value>
    </parameter>
    <repeat>false</repeat>
  </action>
  <outputSpecification>
```

```

<data>
  <name>cdr</name>
  <type>BOOLEAN</type>
</data>
<view>/view/cdrplot.xml</view>
</outputSpecification>
</modality>

```

Figure 6. SingleCDRAnalysis modality.

```

<modality>
  <init-action>
    <uri>/group/fear-detection/contributors</uri>
    <method>GET</method>
  </init-action>
  <action>
    <uri>/engine/workflow/cdr</uri>
    <method>POST</method>
    <parameter>
      <name>sourceUser</name>
      <reference xpath="//users/user"/ type="MAP">
    </parameter>
    <parameter>
      <name>sourceGroup</name>
      <value>cdr-monitoring</value>
    </parameter>
    <repeat>false</repeat>
  </action>
  <outputSpecification>
    <data>
      <name>cdr</name>
      <type>BOOLEAN</type>
    </data>
    <view>/view/cdrplot.xml</view>
  </outputSpecification>
</modality>

```

Figure 7. GroupFearDetectionAnalysis modality.

```

<workflow>
  <node>
    <type>UserDataReader</type>
  </node>
  <node>
    <type>RR</type>
  </node>
  <node>
    <type>CDR</type>
  </node>
</workflow>

```

Figure 8. SingleCDR workflow.

4.3 Remote Rehabilitation

The remote rehabilitation application service (REHABaaS) involves remote rehabilitation of the limbs (e.g. elbow and knee) of assisted livings. The service is based on the Body-side including two sensor nodes equipped with 3-axial accelerometers. Sensors are placed in specific positions of the limbs for collecting accelerometer data, which are then processed by the BAN coordinator to provide specific rehabilitation information such as extension angles of elbow and knee [18].

The specific entities (group, modality, workflow/node, view) defining the REHABaaS are:

- The Rehab Group, which represents the group of monitored users to be rehabilitated.
- The RehabDataFeed Modality (see Figure 9), which allows transmitting the rehabilitation data from the Body-side onto the Cloud-side.
- The RehabDataAnalysis Modality, which, based on the RehabDataAnalysis workflow (see Figure 10), performs analysis of the single subject (through a specific workflow node) and provides (statistical) information about the evolution of the rehabilitation.
- The RehabData View, which is the graphical format through which the rehab data will be rendered at the Viewer-side.

```

<modality>
  <inputSpecification>
    <data>
      <sensor1Data>
        <name>AccXSample</name>
        <type>INTEGER</type>
        <source>ECGShimmerSensor1</source>
        <name>AccYSample</name>
        <type>INTEGER</type>
        <source>ECGShimmerSensor1</source>
        <name>AccZSample</name>
        <type>INTEGER</type>
        <source>ECGShimmerSensor1</source>
      </sensor1Data>
      <sensor2Data>
        <name>AccXSample</name>
        <type>INTEGER</type>
        <source>ECGShimmerSensor2</source>
        <name>AccYSample</name>
        <type>INTEGER</type>
        <source>ECGShimmerSensor2</source>
        <name>AccZSample</name>
        <type>INTEGER</type>
        <source>ECGShimmerSensor2</source>
      </sensor2Data>
      <extensionAngle>
        <name>AngleSample</name>
        <type>INTEGER</type>
        <source>BSN</source>
      </extensionAngle >
    </data>
  </inputSpecification>
  <init-action>
    <uri>/group/rehab-monitoring/data</uri>
    <method>DELETE</method>
  </init-action>
  <action>
    <uri>/group/rehab-monitoring/data</uri>
    <method>PUT</method>
    <repeat>true</repeat>
    <trigger after="1"/>
  </action>
</modality>

```

Figure 9. RehabDataFeed modality.

```

<workflow>
  <node>
    <type>UserDataReader</type>
  </node>
  <node>
    <type>Stats</type>
  </node>
</workflow>

```

Figure 10. RehabDataAnalysis workflow.

5. CONCLUSIONS

In this paper we have proposed an approach based on the BodyCloud architecture for the engineering of large-scale BAN application services. The approach allows for rapid prototyping of BAN applications involving the management of large networks or communities of individuals. In particular, BodyCloud supports the development of BAN services through the integration of (i) a flexible BAN framework, namely Android SPINE, which allows for the implementation of the on-body sensor-based application, and (ii) a SaaS-level Cloud computing framework supported by Google App Engine. The Cloud computing framework provides four effective programming abstractions to build Cloud-based BAN services: Group, Modality, Workflow, and View. After providing an overview of BodyCloud by describing its main functionalities, we have described the process for engineering BodyCloud applications and elucidated such engineering process by showing three case studies (ECGaaS, FEARaaS, and REHABaaS).

Future work will be devoted to (i) extending the proposed engineering approach into a full-fledged development methodology supported by a CASE tool and (ii) integrating well-established workflow-based data mining/analysis tools (such as KNIME, RapidMiner, Weka) into BodyCloud to provide access to a wide range of already developed algorithms for data analysis and mining.

6. ACKNOWLEDGMENTS

Our thanks to Daniele Parisi and Vincenzo Pirrone for their efforts in implementing BodyCloud and the ECGaaS application service.

7. REFERENCES

- [1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292-2330, 2008.
- [2] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. C. Leung, "Body area networks: a survey," *Mobile networks and applications*, vol. 16, no. 2, pp. 171-193, 2011.
- [3] G. Fortino, M. Pathan, and G. Di Fatta, "BodyCloud: Integration of Cloud Computing and Body Sensor Networks," In *Proc. of CloudCom'12*, 2012.
- [4] G. Fortino, D. Parisi, V. Pirrone, G. Di Fatta, "BodyCloud: A SaaS Approach for Community Body Sensor Networks," *Future Generation Computer Systems*. 2014. to appear.
- [5] A. Cuzzocrea, G. Fortino, O. Rana, "Managing Data and Processes in Cloud-Enabled Large-Scale Sensor Networks: State-Of-The-Art and Future Research Directions," In *Proc. of IEEE CCGrid 2013*.
- [6] W. Kurschl, and W. Beer, "Combining Cloud Computing and Wireless Sensor Networks," In *Proc. of iiWAS'09*, 2009.
- [7] X. Chu, and R. Buyya, "Service-Oriented Sensor Web," In: *Sensor Networks and Configuration*. Springer, 2007.
- [8] M. Yuriyama, and T. Kushida, "Sensor-Cloud Infrastructure-Physical Sensor Management with Virtualized Sensors on Cloud Computing," In *Proc. of NBIS'10*, 2010.
- [9] S. Distefano, G. Merlino, and A. Puliafito. SAaaS: a Framework for Volunteer-Based Sensing Clouds. *Parallel and Cloud Computing* 1(2), 2012.
- [10] K. Kahol, "Integrative Gaming: A Framework for Sustainable Game-Based Diabetes Management," *Journal of Diabetes Science and Technology*, Volume 5, Issue 2, March 2011.
- [11] S. Pandey, W. Voorsluys, S. Niu, A. Khandoker, and R. Buyya, "An Autonomic Cloud Environment for Hosting ECG Data Analysis Services," *Future Generation Computer Systems*, 2011.
- [12] G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, R. Jafari, "Enabling Effective Programming and Flexible Management of Efficient Body Sensor Network Applications", *IEEE Transactions on Human-Machine Systems*, vol. 43, no. 1, pp. 115-133, Jan. 2013. DOI: 10.1109/TSMCC.2012.2215852.
- [13] F. Bellifemine, G. Fortino, R. Giannantonio, R. Gravina, A. Guerrieri, M. Sgroi, "SPINE: A domain-specific framework for rapid prototyping of WBSN applications" *Software Practice and Experience*, Wiley, 41(3), 2011, pp. 237-265, DOI:10.1002/spe.
- [14] Google App Engine. <https://cloud.google.com/products/>
- [15] G.S. Wagner, *Marriott's Practical Electrocardiography*, Lippincott Williams & Wilkins, Ed., 2007.
- [16] A. Andreoli, R. Gravina, R. Giannantonio, P. Pierleoni, and G. Fortino, "SPINE-HRV: A BSN-Based Toolkit for Heart Rate Variability Analysis in the Time-Domain," in *Wearable and Autonomous Biomedical Devices and Systems for Smart Environment*, ser. *Lecture Notes in Electrical Engineering*. Springer Berlin Heidelberg, 2010, vol. 75, pp. 369-389.
- [17] R. Covello, G. Fortino, R. Gravina, A. Aguilar, J.G. Breslin, "Novel method and real-time system for detecting the Cardiac Defense Response based on the ECG," *IEEE Symp. MEMEA 2013*, Ottawa, Canada, 2013.
- [18] S. Iyengar, F. Tempia Bonda, R. Gravina, A. Guerrieri, G. Fortino, A. Sangiovanni-Vincentelli, "A Framework for Creating Healthcare Monitoring Applications Using Wireless Body Sensor Networks", In the *Proc. of the 3rd International Conference on Body Area Networks (BodyNets'08)*, Tempe (AZ), USA, Mar. 13-15, 2008.