

Context Aware Mobile Application Architecture (CAMAA) for Health Care Systems

Standardization and abstraction of context aware layers

Hamid Mcheick
Department of Computer Science
University of Quebec at Chicoutimi
Chicoutimi, Canada
Hamid_mcheick@uqac.ca

Hassan Sbeity, Hussein Hazimeh, Jihad Naim,
Mohamed Alameh
Open Arab University
Computer Science Department, Lebanese University
Beirut, Lebanon
hsbeity@aou.edu.lb; Hussein.Youssef.Hazimeh@edu.hefr.ch
; jnaim12@gmail.com; alameh.mhd@gmail.com

Abstract—Context awareness was introduced recently in several fields in quotidian human activities. Among context aware applications, health care systems are the most important ones. Such applications, in order to perceive the context, rely on sensors which may be physical or virtual. However, these applications lack of standardization in handling the context and the perceived sensors data. In this work, we propose a formal context aware application architecture model to deal with the context taking into account the scalability and interoperability as key features towards an abstraction of the context relatively to end user applications. As a proof of concept, we present also a case study and simulation explaining the operational aspect of this architecture in health care systems.

Keywords—Context aware systems; pervasive and ubiquitous systems; Health Care Systems; Agent systems ; Sensor Model

I. INTRODUCTION

In recent years, mobile devices were spread in an unprecedented manner. The smartness of these devices allows them to perform multiple and hard tasks. The plurality of sensors implemented in these devices gives each user a powerful, multi-feature computing entity.

Smart devices, armed with multiple sensors, can sense and react based on their environment, and become context-aware agents. The critical aspect of health care and medical stuff work makes these smart devices an ideal assistant providing an accurate, as well as the closer up-to-date state, of the patient. In addition, smart devices offer a reliable and suitable communication way between hospital's staff members.

Many applications and frameworks were built to deal with context, based on sensed data, through physical and virtual sensors. Among physical sensors we mention the global positioning system sensor (GPS), thermometer, accelerometer, Wi-Fi sensor, etc..., as well as specific sensors in health care domain like heart beats and blood pressure sensors. As an example of virtual sensors we mention the history of the context where the information is not collected directly, but extracted by analyzing a set of collected data.

Despite the many applications built to deal with collected data from the context, these applications lack of a standard architecture. The developers of context aware applications still right now taking care of handling sensed data directly using built-in interfaces provided by the operating system of the smart device. Virtual sensors data are collected and processed according to the case and in an inefficient manner or they are not used at all.

This article is based on a unified sensor interface model published recently in LAAS 2014 [7]. We define in detail a well-defined architecture to deal with contextual data, providing to context-aware applications developers an abstraction of the context. In addition, this architecture permits the subsequent extension by adding more sensors as needed, assuring its scalability. The interoperability is proven using the more popular data format: the extensible markup language (XML), which ensures the information exchange between heterogeneous systems.

After this introduction, which is the first part, we will present some related work in part 2, and then part 3 will be our architecture models, part 4 presents a case study explaining our approach. As a proof of concept, a simulation of this architecture is done and a results are given in the section 4. Finally the conclusion and future work in part 5.

II. RELATED WORKS

Many approaches propose different models to monitor the activities of patients or objects. This section gives an overview of these models.

In Hospitals of the Future [1]-Ubiquitous computing support for Medical Work in Hospitals, the authors propose a Context-aware bed with TV for patient and clinicians, Context-aware pill container, and Context-aware electronic patient record. The bed knows the nurse, patient and medicine tray and display relevant information according to this information such as medical scheme or patient Record.

In Context-Aware Multimedia Computing in the Intelligent Hospital [2], this architecture was proposed based on the needs

of the Royal London Hospital, based on remote consultation such as Audio-video call based on notification. Intelligent hospital has a system that sends notification of awareness and patient data to doctors and specialists.

In Context-Aware Mobile Communication in Hospitals [3], Empower mobile devices in which hospital workers perform their task, and Contextual elements used location, delivery timing. For example: “a message for room 226 to any physician, delivery time today after 2 p.m.”.

A Flexible, Low-Overhead Ubiquitous System for Medication Monitoring [4], the authors proposed a system based on RFID tags for Flexible, Low-Overhead Ubiquitous System for Medication.

Vocera [5] communication system-experimentation in St. Vincent hospital, Birmingham, USA: Based on wearable badge with many capabilities like voice-dialing based on speech recognition, hands free conversation, hands free answer to calls, and voice message if there is no answer. It delivers information directly to user.

Shahriyar et al. [9] proposed three levels architecture of telemonitoring system: i) sensor network, ii) patient server to collect the information, and iii) hospital server that processes information and makes them available to physicians.

Thomasse et al. [10] developed a medical remote monitoring systems at home (project TISSAD). These systems monitor and prevent accidents and aggravation of elderly and chronic health conditions. This project focused on the user by consolidating data into four classes: identification, historical requirements, medical history and medical data.

Stoicu-Tivadar et al [11] proposed a remote monitoring system of elderly people. This system proposes a strategy to implement an alarm component for this system, which consists of many units in the homes of people monitored for collecting from sensors (medical and environmental) and sending data, and a call center with a server for recording and tracking data. Alerts are modeled in XML.

Dey et al. [6] consider the Context as "any information that can be used to characterize the position of an entity. An entity is a person, place or object. That is considered relevant to the interaction between a user and an application, the user and applications including themselves." This definition generalizes the parameters of context.

According to the previous overview, we observed that the preceding approaches have treated the problem as a specific case point of view, where most of them focused on modeling the patients, hospitals and bedrooms [1,3], leaving the issues of collecting and modeling sensor data. In our approach, we try to propose a standard unified model to treat the problem of modeling and collecting sensor data regardless of the type of

the application, where in this case we can instantiate our model and employ it to serve separate domains, as in this paper we deal with patients’ health situation, the same model can be applied for monitoring devices status, and so forth.

III. NEW CONTEXT AWARE MOBILE APPLICATION ARCHITECTURE (CAMAA) MODEL

Data will be collected continuously from a wide range of sensors. This data should be well organized, formatted, and structured by an Agent before forwarding it to an upper layer, where it should be processed in a later stage. Our new architecture is decomposed as follows, from bottom to top (Fig. 1) A) Sensors layer, B) Agents layer, and C) Application Layer.

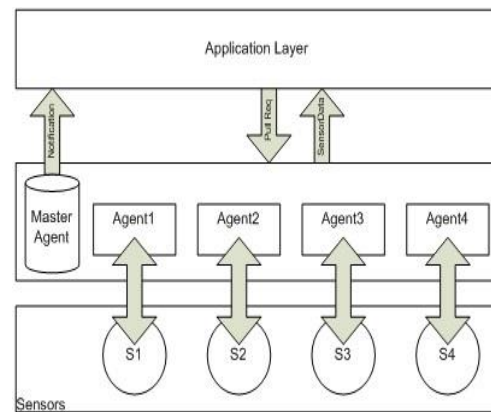


Fig. 1. Context Architecture Layers.

A. Sensors Layer

Contains the sensors available in the user context, which are not limited to the sensors used by the user’s application, it can be physical sensors (GPS, temperature, etc..) or virtual sensors, data is collected continuously and forwarded to the upper Layer “Agents Layer”.

B. Agents Layer

There is an agent associated to each sensor. The agent’s role is to collect data from the sensor, sensors’ data are stored in XML format (Fig. 2), and decides whether to forward it or not to upper layer, according to certain threshold. This threshold measures the difference between the last sent value and the new value (Fig. 3). Also, agents forward sensor data in case of a Pull request coming or scheduled from the upper layer.

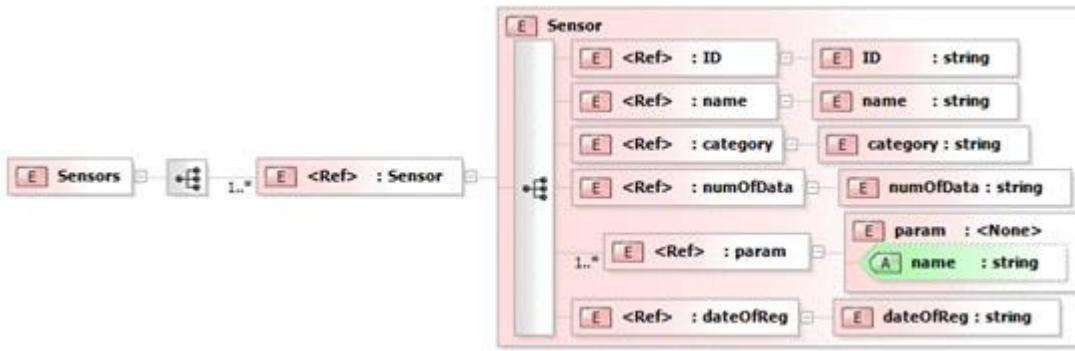


Fig. 2. XML Schema.

The Master Agent is responsible of:

- 1) Registering sensor's observers (subscribers), and their associated threshold.
- 2) Sending notifications for the appropriate subscribers.
- 3) Listing the available sensors: in an XML format which serves as a directory containing all available sensors with their IDs, names and type whether it is physical or virtual (Fig. 4).

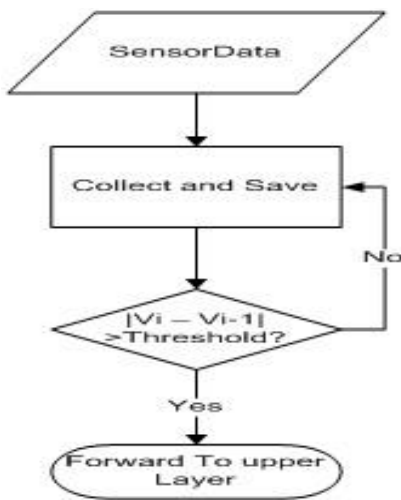


Fig. 3. Threshold test.

Regarding the data sensors, figure 4 describes the details of each sensor. We have four sensors: heart beat, blood pressure, HDL, and sugar volume. For each sensor, we consider five characteristics : name, value, date, time, and GPS. For example, heart beat rate is 90 at 10:12.

ID	name	value	date	time	GPS
s1	heart_beats	90	12/1/2012	10:12	GPS
s2	blood_pressure	9	12/1/2012	10:12	GPS
s3	HDL	120	12/1/2012	10:12	GPS
s4	suger_volume	210	12/1/2012	10:12	GPS

Fig. 4. Sensors Data Sample.

C. Application Layer

Here the client requests subscription from the Agents layer, and pulls data when a notification is received, or by initiating a pull request, also he can discover the available sensors.

IV. CASE STUDY

In this section we are going to demonstrate a case study that conforms to our model mentioned in section 3. We will prove our work on a patient suffering from heart disease and diabetes. Different sensor types (wearable sensors, GPS sensor, etc.) are used to track his medical state and resuscitate him as soon as possible in emergency case.

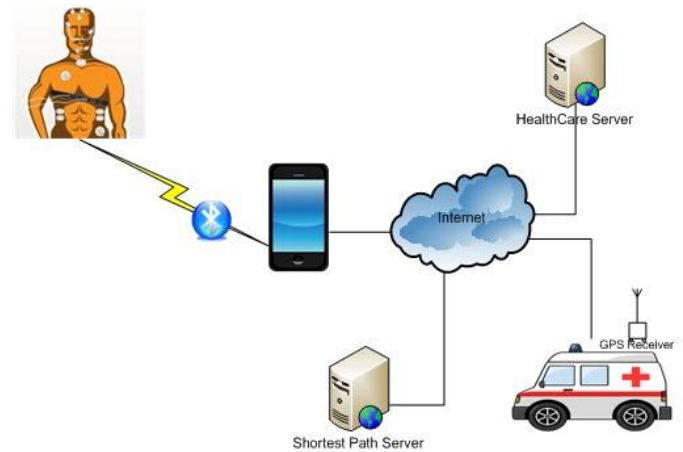


Fig. 5. Case Study illustrations.

Four main sensors are used in the monitoring process:

- Heart beats sensor
- Blood pressure sensor
- Blood sugar volume sensor
- HDL, LDL volume sensor

These sensors are linked to a mobile device (smart phone) already equipped with sensors like (GPS, thermometer ...), where an implementation (middleware) of our model is installed. Sensors' information will act as an input to the

layered model in form of raw data, passing through the three layers.

In the first layer, the data from each sensor is collected by the agents in the second in form of XML data (see Appendix). The second layer checks the new sensed data by comparison with previous data to ensure that the new data represents a new real state. This is accomplished by fixing a threshold for each sensor according to the sensor type and the targeted state. For example, if we fix the threshold of the heart beats sensor to five, then any sensed value with difference less than five will be rejected. This approach has some benefits in decreasing the processing and power consumption without affecting the accuracy of sensed data. Thereafter, a notification of value change is sent to the upper layer.

The third layer receive the notification and then pull the sensed XML data to take the decision for requesting an ambulance or not, using a mathematical model (regression model). If "yes", a request, containing the location of the patient acquired from GPS sensor, is sent to a central health care system requiring an ambulance vehicle to resuscitate the patient quickly. In case of no notifications received, this layer pulls the sensed XML data periodically to guarantee that data is up-to-date.

After which, the ambulance will arrive to the patient home passing by the shortest path, which is calculated instantly using a web service taking as input the location received previously. Here too, the shortest path is determined depending on crowd detection using sensors and the geographic shortest path using the current location of the ambulance sensed via its GPS sensor [12].

Subsequently, the ambulance brought the patient to the nearest hospital following the shortest path.

A. Simulations of the Context aware Architecture Model

We decided to build our simulation around a Publish Subscribe architecture using the Observer design pattern. In our case, applications represent observers and agents represent subjects. It is possible for an application to subscribe to a set of heterogeneous agents and to receive their associated notifications. The exchange protocol is located at the Master Agent level that stores subjects-observers couples (applications / agents) also acting as directory agents for applications. Specifically, we used the concept of separation of concerns through AspectJ [8] to model the spread of a notification of an agent subject to observer applications of this agent.

We have created generic interfaces to represent the concepts of agent application and sensors, to create an implementation of these interfaces as necessary. For example, temperature sensors or the heart rate are two different implementations of the sensor interface. Virtual sensors are used for simulating activity and sending data at regular intervals which are associated with a reference to the corresponding agent. And according to the same logic it is possible to create several different implementations of agents for the same sensor. The interest is to easily change the logical processing of the data for a given sensor, such as the activation threshold.

Finally, we integrated inversion of control (IoC) mechanism such as configuring sensors, agents and applications from an XML file. It is therefore possible to determine the parameters related to the processing logic agents and the behavioral characteristics of virtual sensors (amplitude variations, frequency of updating, unit of measure). Before making the whole more consistent and it is possible to test the interaction between the different layers, we have also developed simple applications that highlight the notification of the master agent through journaling.

We consider two applications: one control the temperature received from sensor temperature agent and another control the heart rate agent. These agents monitor the sensors using aspectJ and injection of control (IoC) mechanisms. Figure 6 and figure 7 show two scenarios and results of this simulation.

In the appendix A and B, a part of configuration XML file is given to illustrate these results.

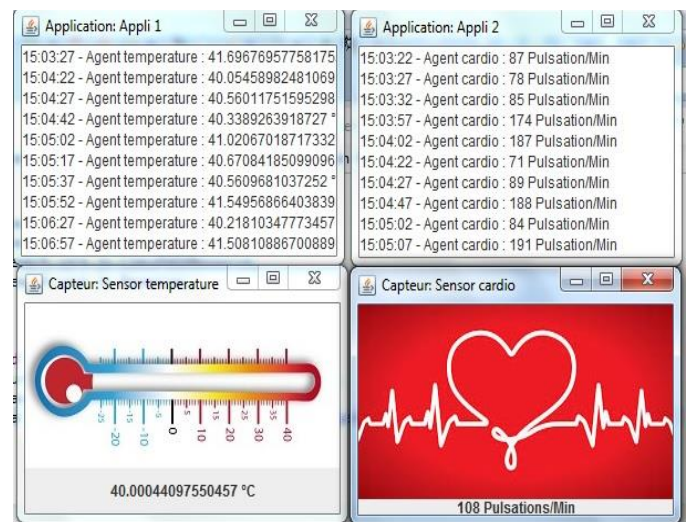


Fig. 6. Result of the simulation of Context aware Architecture Model

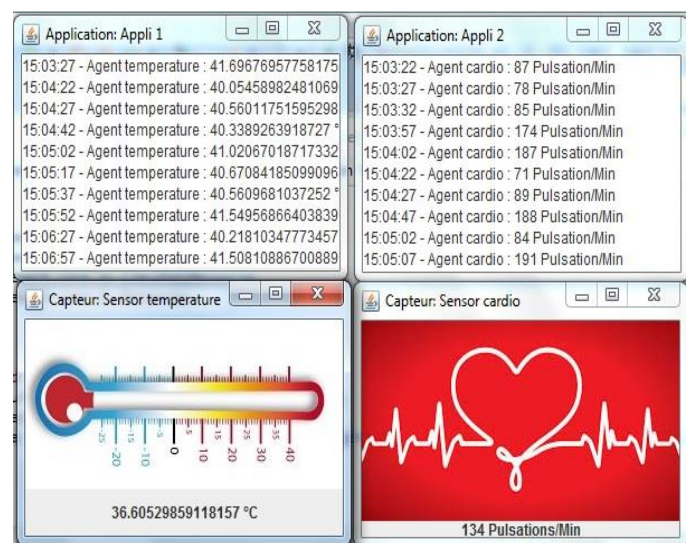


Fig. 7. Another result of the simulation of Context aware Architecture Model

V. CONCLUSION AND FUTURE WORK

In this article, we propose a way to standardize the dealing with sensed data regardless of type of the sensor. A new context architecture model is introduced representing an interface between the application layer and sensors and providing an abstraction of the sensors details that simplify dealing with sensed data. We release a case study, as a proof of concept, to explain and prove the effectiveness of such architecture in health care domain.

Although interactions between the different entities are functional, we would also have treated the communication aspect network. A common approach in the architecture involves the use of a message-oriented middleware that can manage the queues of messages and priorities. Web services are also often mentioned as easily integrated in heterogeneous environments.

In the simulation, we have even been able to create our own data modeling in the context of these exchanges agents, master agents and applications, in order to standardize the structure of the information collected by sensors, for example using an XML format.

More generally, our research has shown that contextual applications aroused interest increasingly important not only in the medial environment but in many research projects worldwide.

Further research could be on the integration of this architecture in the operating system of smart devices dealing with sensors, making sensor data processing as transparent as possible. Also, an experimentation with different kinds of sensors (physical and virtual) is one of future research issue to be done in order to improve this architecture model.

ACKNOWLEDGMENT

This work is supported by the Department of Computer Science at the University of Quebec at Chicoutimi, Quebec Canada; Arab Open University and the Faculty of Science (computer science department) at the Lebanese University.

REFERENCES

- [1] J. Bardram, "Hospitals of the future—ubiquitous computing support for medical work in hospitals," Workshop Ubihealth 2003.
- [2] S. Mitchell, M. Spiteri, J. Bates, and G. Coulouris, "Context aware multimedia computing in the intelligent hospital," in: Proceedings of the Ninth ACM SIGOPS European Workshop, Denmark, September 2000.
- [3] A. Miguel, I. Ana, and M. Victor. "Context-aware mobile communication in hospitals," IEEE Computer 36.9 (2003):38-46.
- [4] Fishkin, M. Wang, "A Flexible, Low-Overhead Ubiquitous System for Medication Monitoring," Intel Research Seattle Technical Memo IRS-TR-03-011, October 25, 2003.
- [5] V. Stanford, "Beam me up, Dr. McCoy," IEEE Pervasive Computing Mag. 2 (3) (2003) 13–18.
- [6] A. K. Dey, 2001. "Understanding and using context," Personal and ubiquitous computing, vol. 5, pp. 4-7.
- [7] Fatima Abdallah, Hassan Sbeyti, Ali Masri, Malak Khreis, Hussein Hazimeh, Mohamed Dbouk, and Ahmad Fadlallah. "Unified Sensor interface for Context-Aware Mobile applications (SCAMP)," Poster,

LAAS 2014.

- [8] The AspectJ™ Programming Guide, Copyright (c) 1998-2001 Xerox Corporation, 2002-2003 Palo Alto Research Center, Incorporated. <http://www.eclipse.org/aspectj/doc/next/progguide/index.html>
- [9] R. Shahriyar, F. Bari, G. Kundu, S. Iqbal Ahamed, and M. Akbar. Intelligent Mobile Health Monitoring System (IMHMS). International Journal of Control and Automation, Vol.2, No.3, 13-28, 2009.
- [10] Jean-Pierre Thomesse et l'équipe du projet TIISSAD. Les technologies de l'information intégrées aux services de soin à domicile : Le projet TIISSAD. Informatique et santé, pp.27-34, 2002.
- [11] V. Stoicu-Tivadar, L. Stoicu-Tivadar, V. Topic and D. Berian. A. WebService based Alarm Solution in a TeleCare System. 5th International Symposium on Applied Computational Intelligence and Informatics, pp.117-121, 2009.
- [12] Hamid Mcheick et Malak Khreis, Mohammad Al-kalla, Hala Sweidan, CARE Framework: Context-Aware Reliable Engine Health Focus On Traffic Monitoring System, UKSIM 16 the International Conference on Modeling and Simulation UKSIM-2014 (IEEE), March 26-28, 2014, UK.

Appendix A

Data collected from sensors encoded in XML format:

```
<?xml version="1.0" encoding="UTF-8"?>
<Sensors>
<Sensor>
  <ID>s1</ID>
  <name>heart_beats</name>
  <value>90</value>
  <date>12/1/2012</date>
  <time>10:12</time>
  <GPS x="10" y="20"></GPS>
</Sensor>
<Sensor>
  <ID>s2</ID>
  <name>blood_pressure</name>
  <value>9</value>
  <date>12/1/2012</date>
  <time>10:12</time>
  <GPS x="10" y="20"></GPS>
</Sensor>
<Sensor>
  <ID>s3</ID>
  <name>HDL</name>
  <value>120</value>
  <date>12/1/2012</date>
  <time>10:12</time>
  <GPS x="10" y="20"></GPS>
</Sensor>
<Sensor>
  <ID>s4</ID>
  <name>suger_volume</name>
  <value>210</value>
  <date>12/1/2012</date>
  <time>10:12</time>
  <GPS x="10" y="20"></GPS>
</Sensor>
</Sensors>
```

Appendix B

```
<bean name="AgentTemp"
class="layer.agents.TemperatureAgentImpl">
```

```
<constructor-arg index="0" value="Agent
temperature"/>
<constructor-arg index="1" value="36"/>
<constructor-arg index="2" value="4"/>
<constructor-arg index="3" value="°C"/>
</bean>

<bean name="AgentCardio"
class="layer.agents.CardioAgentImpl">
<constructor-arg index="0" value="Agent cardio"/>
<constructor-arg index="1" value="130"/>
<constructor-arg index="2" value="40"/>
<constructor-arg index="3" value="Pulsation/Min"/>
</bean>
```

```
<property name="applicationsList">
<list>
  <ref bean="Application1" />
  <ref bean="Application2" />
</list>
</property>
<property name="agApMap">
<map>
  <entry key-ref="AgentTemp" value-
ref="Application1" />
  <entry key-ref="AgentCardio" value-
ref="Application2" />
</map>
</property>
```