Towards a Middleware for Context-Aware Health Monitoring

Eduardo A. Oliveira[1(✉)], Fernando Koch[2], Michael Kirley[1],
and Carlos Victor G. dos Passos Barros[3]

[1] Department of Computing and Information Systems,
The University of Melbourne, Parkville, VIC 3010, Australia
{eduardo.oliveira,mkirley}@unimelb.edu
[2] SAMSUNG Research Institute, São Paulo, Brazil
fernando.koch@samsung.com
[3] Recife Center for Advanced Studies and Systems, Recife, Brazil
carlos.gomes@cesar.org.br

Abstract. The surge of commodity devices, sensors and apps allows for the continuous monitoring of patient's health status with relatively low-cost technology. Nonetheless, current solutions focus on presenting data and target at individual health metrics and not intelligent recommendations. In order to advance the state-of-the-art, there is a demand for models that correlate mobile sensor data, health parameters, and situational and/or social environment. We seek to improve current models by combining environmental monitoring, personal data collecting, and predictive analytics. For that, we introduce a middleware called Device Nimbus that provides the structures to integrate data from sensors in existing mobile computing technology. Moreover, it includes the algorithms for context inference and recommendation support. This development leads to innovative solutions in continuous health monitoring, based on recommendations contextualised in the situation and social environment. In this paper we propose a model, position it against state-of-the-art, and outline a proof-of-concept implementation.

Keywords: Intelligent agent · Context aware · Health · Middleware

1 Introduction

Wearable Health-Monitoring Systems is receiving large attention by both industry and academic research [7,11,18]. Current solutions focus on collecting mobile sensor data and presenting data and target at individual health metrics. They fail in proposing intelligent recommendations and correlating with situational and/or social environment. For instance, an application that measures heartbeat rate issues an alarm if a threshold is reached and the user is running. However, it does not take into consideration that the user is running at a park with a colleague. In this case, it could issue the warning to his running mate to slow the pace, thus enhancing the recommendation efficiency. Hence, there is

a demand for a solution able to integrate data from multiple sources and support advanced decision making in automated health recommendation, based on contextual and social aspects.

Context awareness provides the tools for personalised health monitoring. It provides techniques to implement noise filtering, information selection, and service adaptiveness [8]. For that, accurate inference of context parameters is paramount to support alarming and intelligent recommendation in health monitoring. However, we identified a lack of techniques to infer context parameters based on social health aspects. We seek to improve current models of data aggregation from multiple sources, social data, situation data, and predictive analytics. This development will support innovative solutions in health monitoring that relate situational and/or social environment to provide recommendations and decision support.

We introduce a middleware called "Device Nimbus" that provides the structures to integrate data from diverse sensors in commodity mobile computing technology and execute the models of context and predictive analysis. The solution is being designed to fulfilling the requirements of Internet of Things, namely: heterogeneity, e.g. different sensors, protocols and applications; dynamicity, e.g. arrival and departure of devices and sensors; analysis, e.g. contents personalization, recommendations and prediction, and; evolution, e.g. support for new protocols, devices and sensors. The proposal encompasses three main components: Data Collectors, Data Integration and Intelligent Modules. The resulting solution addresses the requirements of the target scenario by providing context-awareness, adaptivity, flexibility and extensibility to the proposed middleware.

This paper is organised as follows. Section 2 details the proposal for "Device Nimbus", presenting requirements and expected results. Section 3 provides an overview of the state-of-the-art and comparative analysis. The paper concludes with Sect. 4 by providing our perspective on technology development and future work.

2   Proposal

Device Nimbus provides a stepping-stone towards solving many of the problems currently found in health domain such as: tracking users based in lots of different mobile devices/sensors/protocols, providing personalized feedbacks and getting connected with clinics and hospitals and, was designed to deal with big amounts of data. Many people agree that middleware plays a vital role in hiding the complexity of distributed applications. Middleware typically operate in an environment that may include heterogeneous computer architectures, operating systems, network protocols, devices and databases [15]. Device Nimbus middleware will progress the state of the art supporting the design of health systems and applications composed of a large number of independent, autonomous, heterogeneous and interacting sub-systems, sensors and mobile devices. Developers will be able use this middleware when developing new apps, which can collect and analyze personal metrics/ data in a variety of pre-determined ways.

Network communication, scalability, reliability, coordination and heterogeneity have been common requirements of traditional middleware such as Remote Procedure Calls (RPC), Message Oriented Middleware (MOM), Distributed Computing Environment (DCE), Transaction Processing Monitors (TPMON), and Object Oriented Middleware (ORB) [15]. Therefore, some requirements different from traditional middleware have to be considered for the middleware supporting applications and services in the ubiquitous environment. We propose the following new requirements for the platform:

– Context-Awareness: context should include device characteristics, user's activities/behavior/routine, and services.
– Adaptivity: adaptivity should enhance significantly the security and trustworthiness of the middleware and of the large number of independent, autonomous, heterogeneous and interacting sub-systems by incorporating novel technologies that promote their autonomous/autonomic management when addressing attacks and operational failures. The system should be able to recognize unmet needs within its execution context and to adapt itself to meet those needs.
– Lightweight: minimum range of functionality used by most applications.
– Flexibility: all middleware layers will be easily configurable through an administration API that will be accessed through management consoles. Properties such as the routing, conversion and storage of data, will be capable to be configured at runtime.
– Extensibility: on top of the middleware, it will be possible to easily add new smart services that aggregate on top of the gathered data, as well as to plug data consumers. Both approaches allow generating relevant information on top the integrated data that was collected by the integrated systems.
– Standards-compliance: this project will utilize open standards for interfaces definition, network communication, and data representation, also allowing the extensibility of the middleware by facilitating the integration of additional sub-systems and services.

Figure 1 depicts the conceptual middleware purpose. At the core, it will provide the mechanism to integrate mobile devices, social networks and health sensors; to derive a general architecture enabling general interoperability and is based in the use of an intelligent agent. Figure 2 depicts the proposed middleware architecture. The proposed context-aware system can be represented as a layered middleware composed from bottom to top by sensors, raw data retrieval, preprocessing, storage or management, and an application layer. This approach will allow for the identification of common concepts in both context-aware and prediction computing frameworks, allowing us to devise a general concept for smarter device development. It will be possible to pool data from smart devices in terms of context awareness: for text mining, sentiment analysis, node classification in the context of this application domain. Individual users will be able to automatically convert "units" from smart devices and export and send data/reports to the physicians, health groups, hospitals and even social networks. In short, the

Fig. 1. The high-level view of the Device Nimbus concept

proposed middleware aims to collect and process data from multiple sources, as well as to infer events or patterns that suggest more complicated circumstances.

Data Collectors. The Data Collectors are the part of the middleware that is responsible for collecting data from different devices and sensors. The objective of this layer is to allow for the easy integration of sub-systems that collect data from the external world. This data collection can be achieved via sensors that provide data, through people that feed the system with data (e.g. through mobile devices), or through systems that are able to gather non-structured data from the Web (e.g. social networks, web pages, documents - intelligent agents should be used as consumers of Third-party applications - private protocols). A key aspect is data from this layer may come from different domains (e.g. fitness, illness, weather), which ultimately will allow the Data Integration Layer to extract cutting-edge information for supporting more advance smart services. Due to the dynamic nature of sensors/systems that may enter or leave the middleware in an unpredictable way, we decided to use a dynamic services platform in order to bring SOC to this layer. Both dynamicity and flexibility that allow the evolution of components and services at runtime, among other reasons, made the OSGi the platform of choice for constructing this layer. Built on top, an ESB is responsible for receiving data from different sensors and systems and delivering them into the middleware.

Data Integration. The second key block in the design, is the part of the middleware that is responsible for persistence and data integration. The data collected from the Data Collection Systems and persisted in an environment that relies e.g. on a Cloud Computing infrastructure for guaranteeing the provisioning of
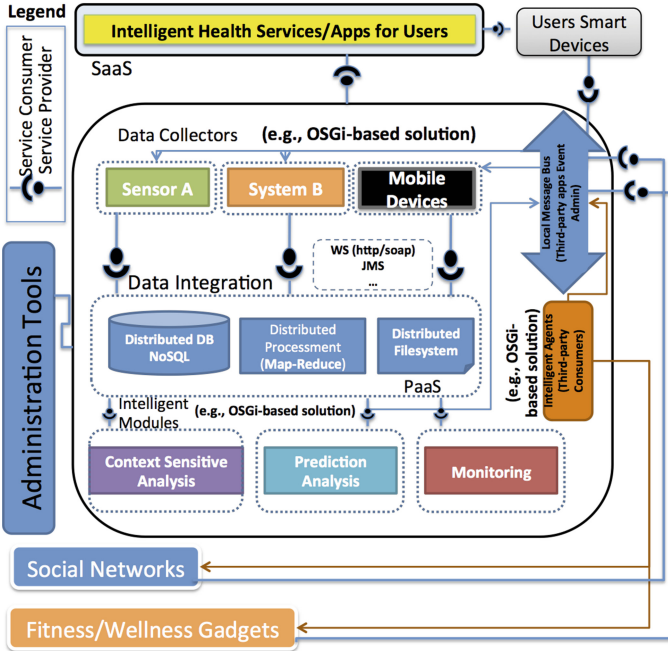
Fig. 2. The high-level view of the Device Nimbus architecture

the necessary storage space. Moreover, the middleware is able to handle several communication protocols thanks to bridge mechanisms we provide: Java message service (JMS) and web services (HTTP/SOAP). Bridges to other protocols, such as XMPP, could be easily added even during execution. The ESB and the intelligent agent of the middleware manage the input data from the different devices in a consolidated NOSQL database.

A significant challenge when developing smart applications, as well as interoperability of distributed systems, is the design of techniques for the integration of distributed data on the Web and from sensors. Processing and analysing acquired data, associated with concepts of pervasive and ubiquitous computing, among others, supports smart applications and context-sensitive systems development. The proposed data integration module is developed for ensure big data processing, classification and organization to support the development of applications on top. Additionally, a service layer providing access to the processed data allows the construction of smart applications and services that reuse functionality of the platform.

The data integration module can be developed with third party components and engines for processing the data and inferring information and knowledge from it. Mechanisms for data analysis and data mining must be used in this module.

Intelligent Module. The third block in the proposed design is the part of the middleware that is responsible for data mining and data analysis. The data-reasoning engine, which would be an engine that can use a large number of techniques (e.g., data correlation, decision tree, information gain, computational context, predictive analysis) transparent to the applications, extracting information from the massive data that are stored. This component combines context aware capability and predictive analysis, using state-of the-art machine learning models. Due to the dynamic nature of artificial intelligent modules that may enter or leave the middleware (additional analysis new modules), the intelligent module was designed to be integrated with OSGi. The middleware considered dynamic services platform in order to bring SOC to this layer. The Intelligent Module is being built on top of Data Integration layer. To support more sensors and systems in Device Nimbus, the ESB in the middleware must be updated with new and different components.

As a strategy to collect data from more web environments, the proposed middleware was also designed based on an intelligent agent architecture-based use. The intelligent agent was designed and added as part of the Middleware to collect data from users, based on their provided logins in web environments (e.g. Twitter, Facebook, Skype, Gtalk, other). The intelligent agent was strategically designed to track and monitor #hashtags and, to work as a chatterbot. Through Natural Language Processing (NLP), the agent can interact with users in different environments. In the same way that sensors and systems can communicate with Device Nimbus to provide data (through the ESB), users can provide data (logins in social networks, devices that they want to be tracked and monitored, other), to the middleware through simple and natural chats with the intelligent agent.

To better explain how the intelligent agent works in the middleware, we present an example scenario that describes the interaction between one user and the Device Nimbus. In this scenario, we assume that the middleware will be able to collect data from heterogeneous and distributed **Sensors (S)**, considering **Context Elements (CE)** to answer **Questions (Q)**:

– **S** = {Humidity, Temperature, NFC, Luminosity, Facebook, Twitter};
– **C** = {{New posts in Facebook and Twitter, from the middleware users', using nikeplus or runkeeper apps},{Interactions between users' and the intelligent agent of the middleware through Facebook or Twitter - NLP},{Big climatic changes},{Holidays and special dates}};
– **Q** = {{Identify runners in a specific location, based on data collected from Twitter or Facebook (#hashtag) posts}; {Identify the relationship between runners in a specific location and environmental data (temperature/humidity/ date/time)}; {Identify whether the same runner visited different locations using Facebook or Twitter data}; {Identify the main running locations in the city (city mapping)}; {Identify the main running locations in the city and what are the most empty/crowd date/time}; {Identify the main running locations in the city, what are the most empty/crowd date/time and the relationship between these locations and environmental data (temperature/humidity)}; {Identify

what individuals has to say about the running locations of the city by tracking Twitter and Facebook hashtags}.

To answer the Questions (Q), the Intelligent Agent must be able to merge inputs from the distributed and heterogenous sensors (S), considering the different CE and routine of each user. There are people that like running in the rain, while others love running in sunny days, for example. To identify the main fitness locations in the city, the Intelligent Agent must be able to track the users that decide to be tracked by the middleware. By merging their fitness location, day of the week, time of the day and frequency that they run/exercise, the intelligent will be able to provide rich and personalised feedbacks to each user, based on their needs. If a user loves to run with friends, maybe it's better to go to a crowded location instead of trying to meet people in an empty location.

To ensure the quality of data collected from Twitter and Facebook, the intelligent agent and the ESB are both looking for the same #hashtags and users (logins were provided as input). A single instance of an intelligent agent, which is provided by Device Nimbus middleware, can be available in lots of different environments (such as a contact on Skype and GTalk or as an user in Twitter or Facebook). Despite the fact that the intelligent agent tracks special #hashtags from Device Nimbus users and appears in many different environments, the middleware provides a single agent to them all, which. In other words, the user can chat about his health or routine across different environments with the same bot. If a user starts communicating with the intelligent agent in GTalk, asking him about good spots to run: "where can I run in Melbourne?" he will get an answer about it, as requested. In parallel, the intelligent agent will be monitoring lots of different users on Twitter and Facebook, and will be able to identify where most of them are running in the city, days of the week that people most run, time of the day, and others. To provide best answers, the data collected from environmental sensors and other data sources will be also considered. By providing an interface of Device Nimbus intelligent agent as a chatterbot, more data can be collected and analysed from different users in different environments.

The concept of an intelligent agent monitoring users in different environments was presented in [13,14], and was here adapted to the fitness and wellness domain. To be tracked/monitored by Device Nimbus, each user can just use special commands such as "#addEnvironment Twitter oliveiraeduardo" to set a new login in the middleware (user is in Facebook adding a Twitter login, for example - teaching the bot his others logins distributed in the Web). The advantage to share logins with Device Nimbus is because the middleware with provide health support and assistance to the user. Only the owner of each login recorded in the middleware have access to their personal information and feedback.

For the Natural Language Processing, used by the intelligent agent of the middleware to communicate with the users in the various integrated Web environments, we used the ProgramD library and Drools inference engine (rule-based reasoning). Drools is responsible for integrating users distributed data and for considering context while users are interacting with the intelligent agent of Device Nimbus. The knowledge-api, drools-core, drools-compiler and

drools-decisiontables modules are working with OSGi. ProgramD is a fully functional Artificial Intelligence Markup Language (AIML) bot engine that is implemented with Java. It supports multiple bots, it is easy to configure and runs in a GUI application and also under a J2EE environment. AIML is an XML dialect for creating natural language software agents. When the AIML markup language is loaded for the bot, users can chat with the bot.

The advantage of providing a single intelligent agent in the middleware lies in the fact that with only one agent, Device Nimbus can also have a single integrated database. If a user interacts with the intelligent agent through Facebook, the agent will know, referring to the historical database of the user that he has already communicated with him through Twitter and Skype, and that s/he has demonstrated interest in running spots. At the same time, the intelligent agent is able to integrate these data with data that comes from #hashtags or other different wearable devices, modelling every user based on their routine and unique needs. Device Nimbus middleware provides also an interface to help in configuring, monitoring and managing the Middleware and to get connected with Third-Party apps, as described below:

Administration Tools. The final components of the model that must be addressed are administration tools. The proposed middleware provides an array of administration tools that allow users configuring, monitoring and managing of the subsystems Fig. 2. This basically includes (i) a system management console for visualizing the nodes that participate in the system's architecture (which may vary over time) at runtime, and eventually reconfiguring system parameters on them; and (ii) a tool for visualizing and configuring the system monitoring and adaptation policies. The goal is to provide a sort of administration view (i.e. a control panel) for the people that will be in charge of the system administration.

Intelligent Healthcare Services. User interaction with the middleware is via intelligent health services/apps. Applications based on service-oriented computing will benefit from the middleware, which will provide many services on top of the data that is pre-processed Fig. 2. Examples of such applications are a command and control centre that visualizes the data and analytic information about what is being collected from the data collection systems, and an application that shows trends/predictions about health domain.

In summary, Device Nimbus is designed in order to achieve three major measurable objectives: (1) definition and implementation of components for the data collection, (2) definition and implementation of components for the data integration, (3) definition and implementation of a layer for processing and analyzing the acquired data.

In order to test the proposed middleware and validate the three main components of Device Nimbus, a minimum viable product (MVP) is under development and will be detailed, with results, in the future. A series of tests is being strategically planned to measure the efficacy of the MVP implementation of Device Nimbus. Given constraints, we are nominating fitness and wellbeing apps as our primary source of data from the wider health domain. As a second step, experiments will be conducted using some of the health apps listed in Sect. 3 instead

of the fitness and wellbeing apps. Such experiments will require collaboration with medical professionals, clinics and hospitals.

Each of the components of the proposed middleware will tested sequentially:

- Collecting heterogeneous data: (i) Test data collection from Twitter, Nikeplus, RunKeeper, Gtalk and Skype; (ii) Test data collection from environmental sensors (temperature, humidity, noise and luminosity).
- Integrating data: (i) Test the ability to the middleware to integrate the heterogeneous data into the NoSQL database; (ii) Test the ability of the intelligent agent of the middleware to integrate the users data into the NoSQL database.
- Analyzing data: (i) Test the ability to the middleware to analyse the integrated data (Context Sensitive Analysis).

By collecting, integrating and analysing data, the proposed middleware will be able to answer the Questions (Q) presented before in this Section.

3   Related Work

Many of the existing ICT solutions for smart health device are proprietary, usually provided by large services vendors, e.g. the likes of IBM, Microsoft, Google, Samsung, Apple and others [2]. These solutions/products are designed as a unified, distributed and real-time control platform, adding cloud computing, sensing, simulation, analysis services and applications. Integrated sensor networks, mobile devices and people power these systems, which are able to combine, aggregate, analyse and inspect for deriving knowledge from health settings. Current developments focus on wearable technologies like smart watches instrumented to collect health data, such as: *physiological sensors* to collect heart rate, blood pressure, respiration rate, electrocardiogram, and others; *environmental sensors* that collect external temperature, velocity, acceleration, and others, and; *light reflection sensors* to collect health parameters like oxygen saturation, skin temperature, blood pressure, and others. These efforts supersede early works based on designed instrumentation, such as [3,16], and offer commodity solutions for experimentation with advanced health monitoring.

The combination of sensors, devices and systems from different modalities and standards makes it necessary to develop hardware independent software solutions for efficient application development [9,17]. In this context, there are numerous studies focusing on middleware/platform design [5,10,19]. Middleware can help health sensor/mobile networks to manage their inherent complexity and heterogeneity. The idea is to isolate commons behaviour that can be reused by several applications and to encapsulate it as system services [1].

As a way of avoiding proprietary solutions, the Open Health Tools was created as an open source community with a vision of enabling an ecosystem, where members of Health and IT professions collaborate. This collaboration is based on building interoperable systems (platform) that enable patients and their care providers to have access to vital and reliable information at the time and place it is needed. However, interoperability benefits are highly dispersed across many

Table 1. State-of-the-art comparison

References x requirements	Context	Adaptivity	Lightweight	Flexibility	Extensibility	Open softwares
Microsoft HealthVault	X	X	X	X		
Samsung S Health	X	X	X	X		
Apple iHealth	X	X	X	X		
Open Health Tools		X	X			X
Magic Broker [4]		X	X	X	X	
SOFIA [6]		X	X	X	X	
Xively		X	X	X	X	
Device Nimbus	X	X	X	X	X	X

stakeholders, and early adopters are penalised by negative network externalities and first-mover disadvantages, e.g. faced barriers and challenges that have resulted in partial success, slow progress and outright failure [12].

Although other initiatives like Magic Broker [4], SOFIA [6] and Xively provide part of their infrastructure as free and open source software, they describe system architectures that are just centred on the smart environments concept, for creating interacting objects ecosystems (sensors, devices, appliances and embedded systems). They are not concerned with simultaneously dealing with different vertical axes (e.g., health, weather, fitness and feeding) and with closed or proprietary wearable solutions. In addition, all of those experimental efforts are custom-tailored solutions whose components or subsystems were not modelled as interchangeable parts, nor were conceived to be integrated with other subsystems. In health domain, a big challenge is to collect, integrate and analyse data from proprietary solutions (mobile devices from Nike, Garmin, Google, Apple and others) and from lots of proprietary different sensors and, tracking the same user based in lots of different input devices to provide them contextual feedbacks/suggestions, based on their particular needs. Based on the requirement analysis presented in Sect. 2, we position "Device Nimbus" against the state-of-the-art as described in Table 1. It becomes evident that the proposal covers a gap in current technology not previously contemplated by the analysed technology.

4   Conclusion

We propose an end-to-end solution for continuous health monitoring enhanced with support to contextualised recommendation. The proposal combines environmental monitoring, personal data collecting, and predictive analytics. For that, we introduce a middleware called Device Nimbus. It provides the structures to integrate data from sensors in existing mobile computing technology. Moreover, it includes the algorithms for context inference and recommendation support. An important design feature of the middleware is the inclusion of an active intelligent agent and an environment for the integration and aggregation of services, applications and fitness/wellness data. The rationale, on which our middleware design is based, is that it should be flexible and extensible.

We demonstrate how the propose improves the state-of-the-art by supporting the requirements for Context-Awareness, Adaptivity, Lightweight, Flexibility, Extensibility and Standards-compliance. The middleware encapsulated key blocks (or components) for data collection from heterogeneous sensors and devices; data retrieval; pre-processing; storage or management, and an application layer. The intelligent agent module allows for prediction capabilities and context awareness, allowing for the dissemination of useful and timely information for users in dynamic environments. As future work, we intend to develop and test the MVP in the fitness and wellness domain. Further implementation and partnership with clinics and hospitals will allow Device Nimbus to provide users' with personalized preventive care, providing early warning signs for serious illnesses, by collecting physiological inputs such as heart rate, blood pressure, body and skin temperature, oxygen saturation, respiration rate, electrocardiogram and others.

The middleware will also provide an environment for the integration and aggregation of services and applications in the health domain. The Device Nimbus platform also aims to develop a business ecosystem that targets Health related domains, enabling the participation of different solution providers and thus stimulating the economy in the ICT sector. Third party services will be able to built systems and apps on top of the platform thus forming an ecosystem around the platform. This proposed work is currently under development in a pilot project in The University of Melbourne - Australia, in collaboration with the University of Campinas and the CESAR.

References

1. Alemdar, H., Ersoy, C.: Wireless sensor networks for healthcare: a survey. Comput. Netw. **54**(15), 2688–2710 (2010)
2. Oliveira, E.A., Kirley, M., Vanz, E., Gama, K.: HSPY: an intelligent framework for context and predictive analysis for smarter health devices. In: 2014 International Conference on Information and Communication Technology Convergence (ICTC), pp. 53–58. IEEE (2014)
3. Black, J., Koch, F., Sonenberg, L., Scheepers, R., Khandoker, A., Charry, E., Walker, B., Soe, N.L.: Mobile solutions for front-line health workers in developing countries. In: 2009 11th International Conference on e-Health Networking, Applications and Services (Healthcom), pp. 89–93 (2009)
4. Blackstock, M., Kaviani, N., Lea, R., Friday, A.: Magic broker 2: an open and extensible platform for the internet of things. In: Internet of Things (IOT), pp. 1–8. IEEE (2010)
5. Chatzigiannakis, I., Mylonas, G., Nikoletseas, S.: 50 ways to build your application: a survey of middleware and systems for wireless sensor networks. In: IEEE Conference on Emerging Technologies and Factory Automation, 2007 ETFA, pp. 466–473. IEEE (2007)
6. Filipponi, L., Vitaletti, A., Landi, G., Memeo, V., Laura, G., Pucci, P.: Smart city: an event driven architecture for monitoring public spaces with heterogeneous sensors. In: 2010 Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM), pp. 281–286. IEEE (2010)

7. Gatzoulis, L., Iakovidis, I.: Wearable and portable ehealth systems. IEEE Eng. Med. Biol. Mag. **26**(5), 51–56 (2007)
8. Koster, A., Koch, F., Kim, Y.B.: Serendipitous recommendation based on big context. In: Bazzan, A.L.C., Pichara, K. (eds.) IBERAMIA 2014. LNCS, vol. 8864, pp. 319–330. Springer, Heidelberg (2014)
9. López, T.S., Kim, D.: Wireless sensor networks and rfid integration for context aware services. Artigo publicado no Web Site do Auto-ID Labs (2008). http://www. autoidlabs.org/uploads/media/withhold_AUTOIDLABS-WP-SWNET-026.pdf
10. Lu, H.F., Chen, J.L.: Design of middleware for tele-homecare systems. Wireless Commun. Mob. Comput. **9**(12), 1553–1564 (2009)
11. Lymberis, A., Dittmar, A.: Advanced wearable health systems and applications-research and development efforts in the European union. IEEE Eng. Med. Biol. Mag. **26**(3), 29–33 (2007)
12. Manzaroli, D., Roffia, L., Cinotti, T.S., Azzoni, P., Ovaska, E., Nannini, V., Mattarozzi, S.: Smart-m3 and OSGI: the interoperability platform. In: 2010 IEEE Symposium on Computers and Communications (ISCC), pp. 1053–1058. IEEE (2010)
13. Oliveira, E.A., Tedesco, P.: i-collaboration 3.0: um framework de apoio ao desenvolvimento de ambientes distribuídos de aprendizagem sensíveis ao contexto. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação, vol. 2 (2013)
14. Oliveira, E.A., Tedesco, P.: i-collaboration: um modelo de colaboração inteligente personalizada para ambientes de ead. Revista Brasileira de Informática na Educação **18**(1), 17–31 (2010)
15. Park, N.S., Lee, K.W., Kim, H.: A middleware for supporting context-aware services in mobile and ubiquitous environment. In: 2005 International Conference on Mobile Business, ICMB 2005, pp. 694–697. IEEE (2005)
16. Rolim, C., Koch, F., Black, J., Geyer, C.: Health solutions using low cost mobile phones and smart spaces for the continuous monitoring and remote diagnostics of chronic diseases. In: The Third International Conference on eHealth, Telemedicine, and Social Medicine, eTELEMED 2011, pp. 72–76 (2011)
17. Triantafyllidis, A., Koutkias, V., Chouvarda, I., Maglaveras, N.: an open and reconfigurable wireless sensor network for pervasive health monitoring. In: 2008 Second International Conference on Pervasive Computing Technologies for Healthcare, PervasiveHealth 2008, pp. 112–115. IEEE (2008)
18. Tröster, G.: The agenda of wearable healthcare. In: IMIA Yearbook of Medical Informatics, pp. 125–138 (2005)
19. Waluyo, A.B., Ying, S., Pek, I., Wu, J.K.: Middleware for wireless medical body area network. In: 2007 IEEE Biomedical Circuits and Systems Conference, BIOCAS 2007, pp. 183–186. IEEE (2007)