An Adaptive Model for Exposing WSN as a Service platform

Thanh-Dien Tran, David Nunes*, André Gomes, and Jorge Sá Silva

Department of Informatics Engineering, University of Coimbra
Polo II - Pinhal de Marrocos, 3030-290 Coimbra, PORTUGAL
{than,asng,sasilva}@dei.uc.pt,*alphaendgame@gmail.com

Abstract. Mashing up physical resources with virtual environments provides a bigger potential for new applications using sensor and actuation networks. However, it also brings many challenges because of the limitations as well as the heterogeneity of sensor nodes and networks. As a matter of fact, a hybrid approach should be the one best suited for most needs of integration and interoperability between sensor networks and other virtual environments. This paper presents our RESTful Web Service-based model for this purpose. In addition, we also present two illustrated scenarios of mashups between sensor networks and the Facebook social network and Second Life virtual world platforms.

Keywords: Sensor Networks, Physical-Virtual world Mashup, Service Platform

1 Introduction

Wireless Sensor networks (WSNs) have been receiving a great deal of attention from the research community due to their great potential for useful applications in almost every area. Real world deployments of WSNs, however, are currently not as widespread as the amount of research would lead to believe, because of several factors. Besides the cost of sensor devices not being as cheap as most people expected (e.g., 1 dollar per sensor node), the difficulties in application development, deployment of sensor networks, and integration with existing applications and tools are factors that greatly contribute to the lack of adoption of sensor network solutions. In addition, most of the current sensor network applications are specifically designed to work with a particular type of sensor networks and it is difficult, or even impossible to apply other types.

Currently, there are several challenges when developing applications for WSNs. The first challenge is that development requires a detailed understanding of low level protocol and network, therefore requiring a great amount of effort when starting to build applications. In addition, interoperability is a difficult problem due to the independent proprietary data formats used by different sensor networks. Although it has been proved that it is possible to implement both IP protocol stack (6LowPAN) and web services on the sensor nodes, the applicability and efficiency of these technologies still needs more investigation. Because of the limitations of sensor nodes, allowing directly access to sensors from the Internet brings many challenges and may not be a suitable solution for all WSNs. Therefore, it is necessary to have an infrastructure that easily, securely, and efficiently supports the integration and interoperability of sensor networks with other environments.

In an attempt to contribute towards the seamless interoperability and integration of WSNs with Internet environments, we have proposed and implemented a solution in which sensor nodes and networks are considered as services. In our proposed model, the WSNs are data and functionality providers that are facilitated by an integration platform. The integration platform provides a set of services for consumption by external environments using web service APIs. This model facilitates the integration and interoperability of WSNs with virtual world environments, social networks and other web 2.0 applications. To illustrate the proposed approach, two simple applications that integrate sensor networks with both Facebook and Second Life were implemented. These applications allow Facebook and Second Life users to easily and instantly access information, provided by sensors, about the physical world. The remainder of the paper is organized as follows: section 2 summarizes the main related works. Then the proposed model is presented in section 3. Section 4 describes implementation, applications, and evaluation of the proposed model. The final section presents some conclusions and future work.

2 Related Work

Although sensor nodes are very limited in capability (memory, processing, communication), recent works have proved that it is possible to deploy IP protocol for sensor nodes in WSNs. Dunkels et. al [3], [4] were one of the first research groups that provided a solution for interconnecting WSNs with the Internet using variants of the TCP/IP stack. In addition, 6LoWPAN [9], [8] provides a solution for enabling IPv6 for communication over 802.15.4-based WSNs by introducing an adaptation layer, between the link layer and the network layer. 6LoWPAN applies cross-layer optimization with three primary elements: hear compression, fragmentation and layer-two forwarding. These research results represented great successes for the visions of the Internet of Things and the Web of Things, which have been attracting considerable amounts of research in recent years.

The main research approaches for combining virtual and physical resources use web services, either embedded into individual sensor nodes or via proxies. One of the first approaches used SOAP-based Web services. Tiny Web Services [10] proposed to use WSDL description for exposing sensor's data and functionality. Since the general implementation of SOAP-based web services is unlikely to be suitable for sensor nodes due to being too heavy (i.e., It comprises XML tags to describe and define data), [10] proposed methods for compressing SOAP messages using compress algorithms such as zip, LZW, and XML-specific compression techniques. In addition, [10] also proposed to use Web service Eventing (WS-Eventing [5]) for supporting cycled duty of sensor nodes. A prototype was implemented to prove that it is possible to use SOAP based Web service for resource-constrained sensor nodes. However, despite the payload of the message being compressed, it was still very large when comparing with the bare-bone protocol. Moreover, since Tiny Web Services employed the generic compression algorithms and supported cycled duty, it still required to have a HTTP proxy between the client application and sensor nodes. Another work on this direction was done by Belicato et. al in [2], which proposed to use a SOA architecture for exposing WSNs as web services in order to easily integrate them with Web mashup technologies. This proposal was a gateway-based approach that suggested the use of HTTP and XML encoding over traditional network protocols of WSNs (i.e., not TCP/IP) for communication between sensor nodes and sink nodes, where the latter acted as the proxies/gateways for the client application. Although this model did not use TCP/IP, the use of XML as the message communication between sink and sensor node still significantly increase the overhead of the packets. There are also no implementations of this model as of yet.

Another significant approach for the integration and interoperability of WSNs and virtual environments is to use RESTful Web Services. The Web of Things [12], [7], [13], [6], which inspired from the success of Web 2.0 mashup, proposed integration methods for mashing up WSNs with other web-based platforms. In this project, two methods were proposed and prototyped, one that embedded REStful Web Services into sensor nodes [7], and another one called Smart gateway [13] which interfaces with sensor nodes and exposes their functionality and data as RESTful APIs to external client applications. In these prototypes, Sun SPOT platforms were used for embedding RESTful Web Services. sMAP [1] is another RESTful-based Web Service platform for making physical data and information available and interoperable with other platforms and applications. sMAP proposed a schema to represent measurement information using JSON as the object format, and an architecture for the interoperability among heterogeneous devices. In this model, sMAP gateways act as intermediate components between the resources and the client applications. For scenarios that deal with resource constrained devices such as sensor nodes, there is a compressed and compact version of sMAP that uses UDP over 6LowPAN. This compact version was coupled with a specific sMAP server that runs Embedded Binary HTTP (EBHTTP) [14].

The above-mentioned researches and projects proved that it is feasible to expose the functionality and sensed data of sensor networks as Web Services, which makes the interoperability and integration with other applications easier. However, every approach available has its own strengths and weaknesses and there are still numerous issues that persist. We will now dedicate the following paragraphs to identifying some of the most significant issues not addressed by previous work.

In the SOAP-based Web service approaches, the overhead of the messages was too high even though they were compressed. The RESTful-based solutions are more compact but still too high for many constrained devices. As a matter of fact, the approaches proposed in [12] could not apply for other kinds of sensor nodes if it does not considered to compress the messages. The compressed JSON solution as in [1] produced a reasonable overhead but it needs a proxy (e.g., EBHTTP) to translate messages between sensor nodes and client applications. Most of the current researches for integrating WSNs with Internet environments assumed that data transmission is the main source of energy consumption. Therefore, most of them tried to reduce the overhead of the transmitted packets between sensor nodes. However, in doing so, they significantly increased the processing on the sensor nodes (e.g., compress, parser, etc), which also considerably contributes for power drain.

Most of the research towards integration and interoperability of sensor networks with the Internet environments tries to promote the nodes as the "first class citizens" of the Internet (i.e., the sensor nodes are considered as normal computers) by embedding HTTP servers and web services on them. Despite this, most of the current approaches and models still need a proxy or gateway in the middle. In addition, current approaches mainly focus on providing access to individual sensor nodes, and do not dedicate much attention towards methods for querying sensor networks (data-centric). Another lingering issue present in most of the current approaches is the lack of support for mobility and localization.

Present solutions try to solve the problem of adaptability for communication inside WSNs by adding abstract and standard layers, such as HTTP and web services, above the base network protocols. This means that these abstraction layers have to be implemented into every sensor node, thus rendering current implementations of WSNs obsolete. In order to promote reuse, an important requirement would be to allow the easy integration of already existing deployments onto the web.

With the identified limitations in mind, we envision an infrastructure that allows WSNs to be able to be mashed up with web environments, supports data-centric WSNs, can be used with currently existing sensor applications, and is energy-efficient.

3 Proposed model

The intention of our model is to provide an infrastructure for easily mashing up physical and virtual resources, while providing the necessary functionality and services for WSNs. Our solution considers services for both individual nodes and the sensor network as a whole.

3.1 General Architecture

Heterogeneity, at both the hardware and software levels, is a fact for current and possibility future WSNs. Thus, there is a need for an infrastructure that allows different types of WSNs to be mashed up with other services or content providers on the web, in order to provide more meaningful applications to users. With this purpose in mind, we propose and implement a model for integrating WSNs with external environments (Web 2.0 applications, Virtual worlds, Social Networks and other network/web-based applications). The proposed model is a hybrid gateway-based model that will support both proactive and reactive models of WSNs. In proactive sensor networks, the sensor nodes periodically collect data and send it to the sinks for processing, storing, etc. In reactive models, clients or sinks will send data or functionality requests to the sensor nodes, which will then react accordingly.

A simplified model of our approach is depicted in Fig. 1. As shown in the figure, the model uses multiple proxies/Gateways that mediate the access to the sensors. One of the main advantages of this approach is that it can easily support mobility, localization and scalability. The proxies/gateways are designed to be lightweight, so that they may be installed on modest or limited devices, such as mobile ones.

The proxy receives data packets from the sensor nodes, analyzes, and forwards them to the middleware for further processing. In addition, it also receives commands and requests from the client applications (via the middleware or gateway), processes and sends them to sensor nodes. The proxies also include a few components for supporting mobility and localization.

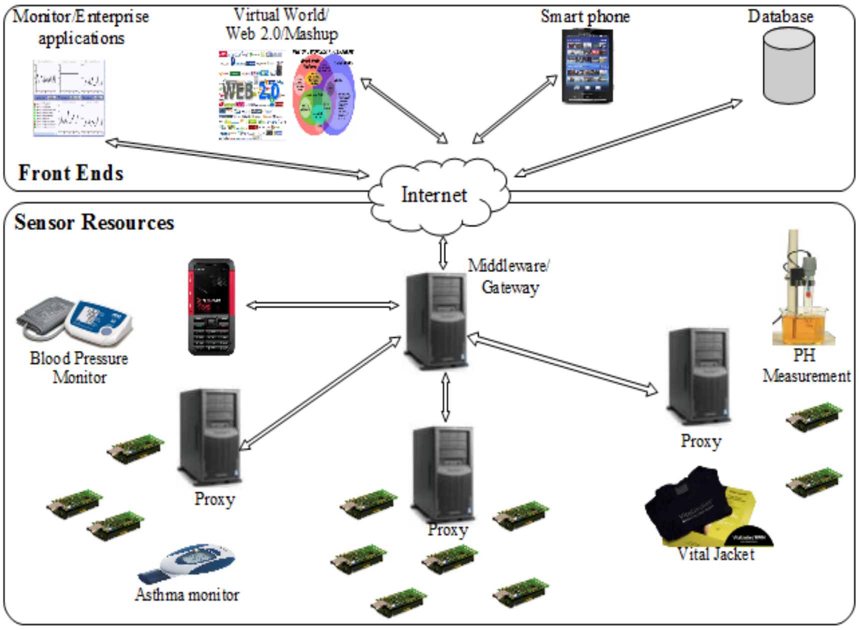

Fig. 1: The general model of the System

The communication between proxies and middleware is based on TCP/IP and Restful web services, currently supporting XML and JSON as the data formats. The communication between proxies and sensor nodes can be implemented independently and be based on any protocols that are appropriate. It can be based on open protocols such as IP or propriety ones.

The system was also thought to be able to support the use of several medical devices. In our current implementation, we use a Blood pressure monitor (also depicted in Figure 1), whose communication is based on the Bluetooth protocol. Such device needs an intermediate that also supports Bluetooth and can communicate with the middleware. In our model, both mobile phones (for increased mobility) and computers can serve as gateways for communication with these types of devices.

The middleware gateway exposes the functionality and data of sensor networks as web services. In a sense, the middleware is the basis for the integration and interoperability between sensor networks and other Web 2.0 environments. This facilitates the development of applications using sensor networks, increases the diversity of possible applications, and promotes the reuse of WSN deployments and software code. In addition, the data coming from the sensor networks could easily be stored in a data warehouse somewhere on the Internet for further processing.

In the next section we will further describe the details pertaining to each architecture component.

3.2 The Proxy Design

The proxy hardware is attached to the wireless network's BaseStation in order to get data from and forward data to it. The several components of the proxy are shown in Fig. 2. The data coming from the sensor networks is analyzed by the Packet Analyzer. The Data Processor component further processes the packets to extract the necessary data for the publishing operations performed by the Data Publishing agent, which forward the information to the Middleware for further processing.

The Proxy also includes an interface for requesting data and functionality from the sensor networks to client applications. The Query and command Processor will process the request to determine whether a request coming from outside is a query for data (e.g., data-centric query) or a command (e.g., sending command to an actuator). In order to guarantee the interoperability and integration between Proxy and Gateway (Middleware) or other client applications, the RESTfulWeb service is used.

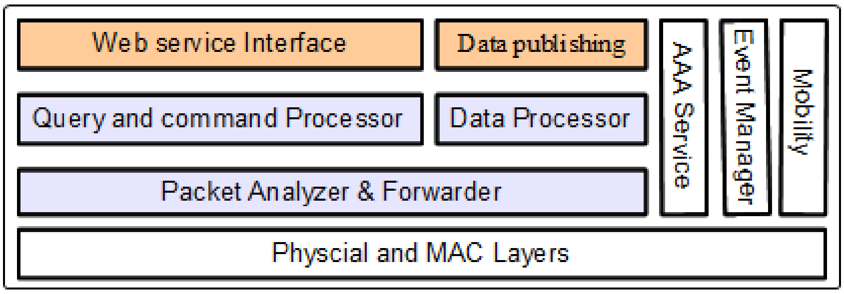

Fig. 2: The Components of Proxy

The Event Manager is based on a publisher-subscriber model and its purpose is to manage the periodicity of events sent from the framework. In our model, applications can request for periodic events to be sent at regular intervals for a certain period of time. These events can represent, for example, the sampling of values from sensors. The AAA Service ensures that only authorized users or applications can access the sensor networks. Besides the core services, the Proxy also includes support for mobile nodes, in the form of a "mobility service". The mobility service is used to manage mobile nodes. Our current mobility is a proxy-based WSN architecture proposed in [11], in which a set of Sensor Mobility Proxies (SMPs) are connected to each other via a shared SMP backbone.

Due to the heterogeneity of sensor nodes, communication protocols, and external applications, the packets coming from the sensor nodes and networks are difficult to standardize. Therefore, it was our intension to design a packet analyzer that can be customized to easily adapt to the majority of sensor protocols and applications. This means that our proxy is highly configurable and extensible for many types of applications.

In order to make it easy to manage multiple sensor networks as well as provide different services to client applications, our approach is a single entry point model in which the Gateway Middleware acts as the accessing point for outside applications. The architecture and components of the Gateway Middleware are described in the next section.

3.3 The Gateway Middleware Design

The Gateway Middleware is the entry point that allows user applications and other environments such as Virtual Worlds and Social Networks to interact with the sensor networks. Fig. 3 shows the architecture of the gateway middleware. It comprises Web service interfaces for external environments such as Web 2.0 applications and client applications to inter-operate or mashup with smart things (e.g., sensors and actuators). Through these interfaces the data from the physical world can be retrieved, stored, and visualized in a meaningful way, from everywhere.

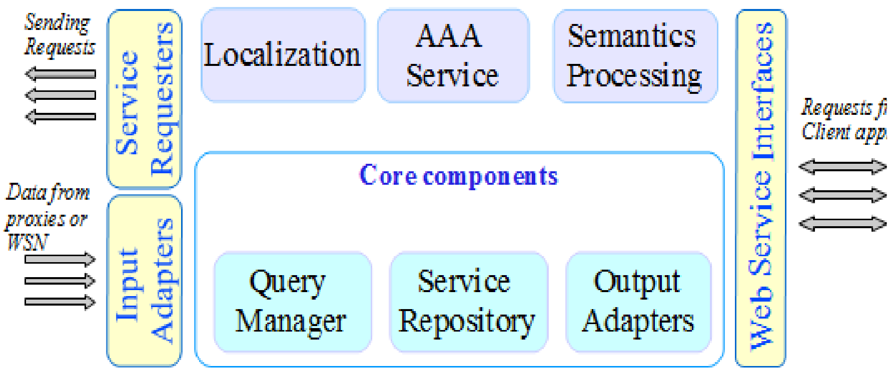

Fig. 3. The Architecture of Middleware

In order to interface with sensor networks or other the smart devices, the gateway middleware provides a set of input adapters which accept data using several different protocols, including HTTP, JMS, and XML-RPC. This component mainly interfaces with the data publishing agent of the proxy presented in section 3.2. The Service requesters provide the services for the outside applications to send requests (for data or actuation) to the sensor networks. The service requesters will communicate with the Web service interface of the proxy to realize the requests of the external applications.

The core components are comprised of the Query manager, the Service Repository and the Output Adapters. The Query Manager manages the periodical requests from the client applications and also includes mechanisms to support data-centric requests for the WSNs. It employs Advanced Message Queuing Protocol (AMQP) to provide a publish-subscribe pattern. The Service Repository is the place where client applications can ask for a set of available services provided by the physical (sensors and actuators) part. The Output Adapters allow the configuration of the formats of the output data, as well as the methods of communication in case the output data needs to be archived in a storage database.

In addition, the gateway middleware also provides useful services for sensor networks besides the basic security ones. The Localization service provides a mean to calculate the location of mobile sensor nodes. This service can be very useful for numerous applications, namely health-care and people monitoring applications, since it is possible to associate the location of a mobile node with the location of a person. In order to calculate the position of sensor nodes, it is required that the proxies collect and provide meaningful data about the nodes such as RSSIs, LQIs, or arriving times. Another important service for sensor networks is the ability to describe the semantics of data. Tagging the sensor data and networks with semantic information allows the exchange and reuse of sensory data across different applications. In our proposed approach, we intend to design semantics as a component at the gateway middleware level. Sensory semantics are helpful for reasoning and decision-making processes. The next section presents the current implementation of the proposed model and its applications.

4 Prototype, Applications and evaluation

4.1 Prototype

We have developed an implementation the proposed model for exposing WSNs as Web services. The currently implemented proxy supports 6LowPAN WSNs and retrieves the RSSIs of the received packets for the localization services at the gateway middleware. The communication between the proxy and gateway middleware is a HTTP-based Web service. The data formats for communication are JSON and XML.

Similarly, the gateway middleware exposes the data and functionality of sensor through a set of services with JSON and XML formats. In addition, the middleware also supports localization services based on RSSI and trilateration algorithm.

The current prototype uses Micaz and Telosb sensor nodes. We are currently also experimenting with the Vital Jacket [15], Blood pressure and PH measurement devices.

The following section presents the usage of the developed prototypes. The current applications mainly illustrate the use of sensor nodes with our framework. An extension of these applications with the integration of other devices is in progress.

4.2 Applications

We have implemented our model on a set of sensor nodes and three desktop computers. Two computers act as proxies and collect data from the sensors, while the other computer acts as the gateway middleware and receives data from the proxies. To illustrate the possibilities of the proposed model, two simple applications were implemented: the first application integrates sensor networks with the Facebook social networking platform; the second application inter-operates between sensors and the Second Life virtual world.

In the first application, the Facebook application Platform is employed to publish sensor data. The Facebook was chosen since it is the most common and well-known social networking platform and it provides an API that allows for the integration with other environments. In this simple demonstration, we publish the temperature data of a monitoring room in the wall of a Facebook account. This helps the account owner to instantly know the temperature of the room and act appropriately in case it unexpectedly changes. The result of this implementation is shown in Fig. 4.

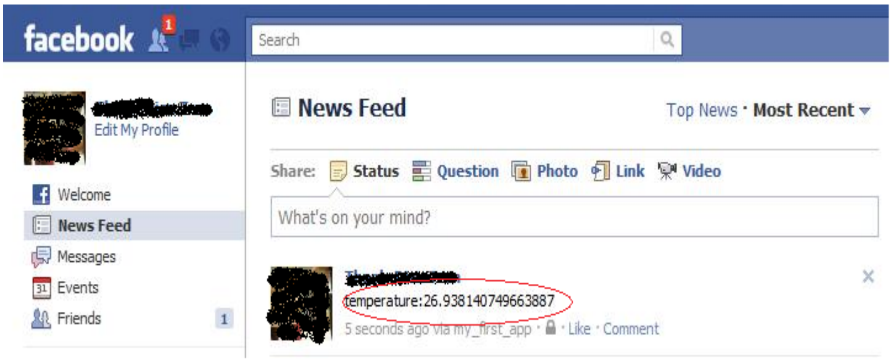

Fig. 4 Sensor Data on FaceBook

The second application that we implemented based on our model was a mashup of WSNs with the Second Life Virtual World. This application is similar to the first one but, instead of displaying sensor data as plain text, we use virtual objects. In this simple application, the temperature of a monitoring target was represented by changing the color of an object attached to an avatar. For example, if the temperature was between 10 and 19 degrees Celsius, the color of corresponding object would become green or if the temperature was greater or equal 30 degrees Celsius, the color would be red. The results of this application are illustrated in Fig. 5.

Different colors of attached object represent the different ranges of temperature of the monitoring target.

Fig. 5. Displaying temperature sensed by sensor node using object attached to avatar and colors in Second Life

The above-mentioned applications show that sensor networks can become an integral part of the virtual environments. The diversity of representation tools in 3D virtual worlds and the sheer amount of users and social connections in social networks can open the door towards new types of applications as well as promote the wide-spreading of sensor networks and smart things and their integration with the Web.

4.3 Evaluation

In our current implementation, we use UDP over 6LowPAN implemented with blip on micaz and telosb platforms. Comparing it with the other Web Service based solutions above-mentioned, our solution minimizes the overhead of communication among sensor nodes and between the sensor nodes and the outside environment by not adding the overhead of the HTTP protocol for internal communications of sensor networks. While accomplishing this, we still manage to keep the advantages of providing seamlessly integration and interoperability with external environments. In addition, it is also easy to add supporting services such as mobility and localization. The downside of our approach is that we lose the ability to be able directly integrate and connect the sensor motes without depending on proxies and the middleware. The memory fingerprint of our implementation is described in Table 1.

	MicaZ	**Telosb**
RAM (byte)	2472	3561
ROM (byte)	31536	21464

Table 1. Momory fingerprint of the proposed model

A comparison between the packet lengths of different approaches is present in Table 2. In this table, we only consider the compressed version of SOAP-based and RESTful-based Web services since the communication messages for the non-compressed versions of these approaches are very verbose and unsuitable for using in sensor networks. As we can see, embedded web service based solutions add significant overhead to packets. In addition to the IEEE 802.15.4 (22 bytes) and 6LowPAN (minimum 7 bytes) headers, Web service based solutions also comprise HTTP and encoding headers. In our case, we only use two fixed bytes that describe the type of data and type of platform that collects the data. Therefore, in order to transfer 2 bytes of payload, the total packet length is 33 bytes. The sMAP solution using packed JSON adds 4 bytes of HTTP and at least 10 more bytes for the packed JSON document. In case of compressed SOAP-based Web Services, the minimum payload for getting the temperature from a sensor node was 26 bytes and the total length was 55 bytes.

	Our model	**SMAP**	**SOAP-based**
Total length in bytes	33	43 [11]	55 [5] *

Table 2. Comparison packet length of different solutions

Although our proposed model includes several layers, it is very fast and data coming from the sensor nodes quickly reaches the client applications. For single hop communications, it takes around 60 ms for data from the sensor nodes to reach the proxy. From there, it takes approximately another 26 ms for data to reach the gateway middleware and be available for the client applications. In the case of our mashup with the Facebook platform, it takes a total of 581 ms for data from physical environments to be published and available to the users. During our tests, we could sometimes get the data from the sensor nodes to be posted on Facebook as fast as in 246 ms. An important note is that this time includes the processing time at the gateway.

In summary, from our study we observe that a hybrid Proxy/Gateway is a suitable solution for mashing up physical resources with virtual environments. Our solution preserves the major concepts of current research on sensor networks while providing an infrastructure for seamless integration and interoperability of wireless sensors with virtual social environments and other web 2.0 applications.

5 Conclusion

Exposing WSN data and functionality as web resources will make sensor networks increasingly more useful. It makes the development of applications for sensor networks easier and more flexible. Developers can employ existing applications for monitoring, controlling, and visualize the sensor data and networks. This approach also opens new ways for mashing up the physical world information with virtual worlds.

In this paper, a model using a web service-based middleware for integrating and inter-operating between physical devices and virtual environments is proposed. An important contribution is that it supports data-centric sensor networks. Localization and mobility services, which can be very useful for health-care and people monitoring applications, are also supported. Semantics will be undeniable components of sensor networks. They help sensor network data to be more meaningful and allow it to be analyzed in other domains. The paper also demonstrated two mashup applications between sensor networks and both Facebook and Second Life.

As future work, we will complete other components of the model and deploy it in real world environments. In addition, we will evaluate its scalability, reliability and performance.

References

1. S. Dawson-Haggerty, X. Jiang, G. Tolle, J. Ortiz, and D. Culler. smap: a simple measurement and actuation profile for physical information. In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, SenSys '10, pages 197-210, New York, NY, USA, 2010. ACM.
2. F. C. Delicato, P. F. Pires, L. Pirmez, and T. Batista. Wireless sensor networks as a service. In Proceedings of the 2010 17th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems, ECBS '10, pages 410-417, Washington, DC, USA, 2010. IEEE Computer Society.
3. A. Dunkels. Full tcp/ip for 8-bit architectures. In Proceedings of the 1st international conference on Mobile systems, applications and services, MobiSys '03, pages 85-98, New York, NY, USA, 2003. ACM.
4. A. Dunkels, J. Alonso, and T. Voigt. Making tcp/ip viable for wireless sensor networks, 2004.
5. D. B. et al. Web services eventing (ws-eventing), 2006.

6. D. Guinard. Towards the web of things: Web mashups for embedded devices. In In MEM 2009 in Proceedings of WWW 2009. ACM, 2009.
7. D. Guinard, V. Trifa, T. Pham, and O. Liechti. Towards physical mashups in the web of things. In Proceedings of the 6th international conference on Networked sensing systems, INSS'09, pages 196{199, Piscataway, NJ, USA, 2009. IEEE Press.
8. J. Hui, D. Culler, and S. Chakrabarti. 6lowpan: Incorporating ieee 802.15.4 into the ip architecture, 2009.
9. G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler. Transmission of ipv6 packets over ieee 802.15.4 networks (rfc 4944), 2009.
10. B. Priyantha, A. Kansal, M. Goraczko, and F. Zhao. Tiny web services for sensor device interoperability. In Proceedings of the 7th international conference on Information processing in sensor networks, IPSN '08, pages 567-568, Washington, DC, USA, 2008. IEEE Computer Society.
11. R. Silva, J. S. Silva, and F. Boavida. A proxy-based mobility solution forcritical wsn applications. In the proceedings of Second InternationalWorkshop on Medical Applications Networking, 2010.
12. V. Trifa and D. Guinard. Web of thing: Architecting the web of thing for tinkerers and hackers. "http://www.webofthings.com", 2011.
13. V. Trifa, S. Wiel, D. Guinard, and T. Bohnert. Design and implementation of a gateway for web-based interaction and management of embedded devices. In The 2nd International Workshop on Sensor Network Engineering, IWSNE'09, 2009.
14. G. Tolle. Embedded Binary HTTP (EBHTTP). IETF Internet-dradft draft-tolle-core-ebhttp-00. Mar. 2010.
15. Biodevices. Vital Jacket. http://www.biodevices.pt (2011)