OPEN ACCESS

International Electronic Conference on Sensors and Applications
1 - 16 June 2014

www.mdpi.com/journal/sensors

Conference Proceedings Paper – Sensors and Applications

Virtualization of event sources in Wireless Sensor Networks for the Internet of Things

Néstor Lucas Martínez[1,*], José-Fernán Martínez [1] and Vicente Hernández Díaz [1]

[1] Universidad Politécnica de Madrid (UPM), Centro de Investigación en Tecnologías Software y Sistemas Multimedia para la Sostenibilidad (CITSEM), Edificio La Arboleda, Campus Sur, Carretera de Valencia Km.7, 28031 Madrid, Spain; E-Mails: {nestor.lucas, jf.martinez, vicente.hernandez}@upm.es

***** Author to whom correspondence should be addressed; E-Mail: nestor.lucas@upm.es; Tel.: +34 91 452 4900 Ext. 20791; Fax: +34 91 336 7817.

Published: 1 June 2014

Abstract: Sensor networks, and more specifically wireless sensor networks (WSN), are generally used to collect information from the environment. The gathered data are mainly delivered to sinks or gateways that become the endpoints where applications can retrieve and use such data. But applications would also expect from a WSN an event-driven operational model, so that they can be notified whenever occur some specific environmental changes instead of analyzing continuously the data provided periodically.In either operational model, wireless sensor networks represent a collection of objects interconnected, in a similar way that is outlined by the Internet of Things vision. In following years sensors will become more capable and resourceful. But in the meantime, they lie into the definition of constrained devices. In addition, to fulfill the vision of the Internet of Things, they must have a virtual representation that allows indirect access to their resources, a model that should also include the virtualization of event sources in a WSN. Thus, in this paper we propose a model for a virtual representation of event sources in a WSN. The event sources are modeled as internet resources that are accessible by any internet application, following an Internet of Things approach. The model has been tested in a real implementation where a wireless sensor network has been deployed in an open neighborhood environment. Different event sources have been identified in the proposed scenario, and they have been represented following the proposed model.

Keywords: wireless sensor networks; internet of things; event-driven; virtualization

---

1. Introduction

The Internet of Things paradigm [1] aims at supporting *smart objects* connectivity so that any physical object (home appliances, cars, products in a mall, smartphones) can interact each other unmanned-wise and provide humans with a better daily experience. Existing technologies like WSN or RFID, among others, are envisioned as foundation technologies for IoT. For accomplishing that, standards are required as they will encourage interoperability among devices and solutions from different stakeholders. The IoT-A European research project [2] is specifying an architectural reference model [3] that will provide a common framework for IoT-A solutions, overcoming interoperability challenges. Briefly, the proposed model in [4] virtualizes devices to obtain their computational representation. The so called virtual device exposes its resources, semantically described, to any other actor by means of services, accomplishing a SOA [5] approach. Different SOA technologies are being used and worldwide accepted, but the one that is becoming preferred for its simplicity and low overload whenever constrained devices are involved is RESTful Web Services, based on the REST [6] approach. In a ROA (Resource Oriented Architecture) solution, system entities can only create, read, update and delete resources hosted by any other system entity.

WSN based solutions are usually event-driven systems to save network resources. Sensors node go to sleep mode until a significant event is triggered, notifying subscribers about that only when it is necessary. That will reduce the energy consumption and the number of messages across the network, improving the overall performance of the WSN. The design of such systems has to comply with IoT reference models in order to be integrated in any IoT solution. Therefore, the elements in an event-driven system like in a WSN have to be properly modelled to match the REST approach as well as the IoT reference model (e.g. IoT-A).

This paper proposes a model that enables REST compliance of event-driven system, virtualizing the corresponding devices as it is being proposed in IoT reference models like IoT-A. The model has been developed and successfully integrated in a pilot in WoO (Web of Objects), a European research project labelled by the ITEA2 research program and funded by the Spanish Ministry for Industry, Energy and Tourism. WoO aims at providing a model for developing a Web of Objects, comprised of objects in an IoT that cooperates smartly to arrange and provide a web of services and complex virtual devices.

The different elements of the models have been integrated in nSOM (nano Service Oriented Middleware), a middleware that is being developed by the Universidad Politécnica de Madrid for deploying WSN solutions on any hardware platform.

The event sources in the WSN are registered and published in a repository, either statically or dynamically by discovering them, as REST resources. The nSOM event manager will send a subscription message to every event-source (sensor node). Any external subscriber will also register in the nSOM repository and will be exposed as a resource. Whenever an event is triggered, the sensor node will notify the event manager that will create a new resource in the repository representing the new event and will notify the appropriate subscriber about that new resource. The event sources have only one subscriber, the event manager, thus avoiding storing large lists of subscribers and reducing the number of messages across the WSN when notifying subscribers.

The repository and the event manager are nSOM elements running in the gateway that connects the WSN to a other network.

2. Proposal

When dealing with the design of any architecture that involves the use of a WSN, there are a set of challenges that must be taken into consideration. Other authors have identified in the past several factors that can influence in the design of the network, like the fault tolerance, the scalability, the hardware constraints, the topology, the communication and so on [7]. In an event-driven model there is also the consideration of whether an event notification must be guaranteed or not, as well as the delivery time. In our scenario, as will be described in the results section, a maximum delay of the delivery time of one minute was imposed by platform requirements.

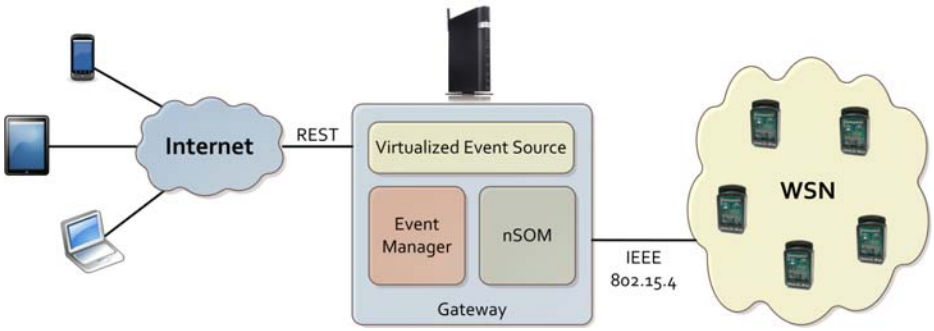
In this model, besides the response time, the limitation of resources has also been taken into account. The message interchange between the entities participating in an event notification has been reduced to a minimum in the domain of the WSN.

2.1. The virtualization model

Every node in a WSN can be considered as a set of resources that can be exposed to other entities, including humans. The same approach can be used to describe the events detected by the sensor nodes. As with any WSN, a gateway that translates and routes the messages from and to such network provides the connectivity to the Internet community.

The proposal is to provide, in the gateway, a virtual representation of network nodes capabilities and functionalities by means of exposed resources. Through this virtual representation, a set of services can be defined to access node resources, and even create new ones to manage new events as they are triggered. This complies with the domain model described in IoT-A [3]. **Figure 1** shows a graphical depiction of the proposed model.

Figure 1. Virtualization model scenario



The model can be defined in three stages. In the first one the gateway detects that a node in the WSN provides events. This detection is done either actively or passively using one of the two methods that has been developed in the nSOM middleware in the scope of the WoO project. Once an event source is identifed, the event manager in the gateway subscribes itself as an event consumer, exposing the event source as a REST service. The REST services are modeled following the design patterns described in [8]. This step corresponds to messages 1.X **Figure 2**.

In the second step the users and other entities that want to be notified when an event source generates an event, subscribes themselves to the event manager in the gateway using also a REST service. Both these resources and the ones related to the sources are shown in **Table 1**. In this way the

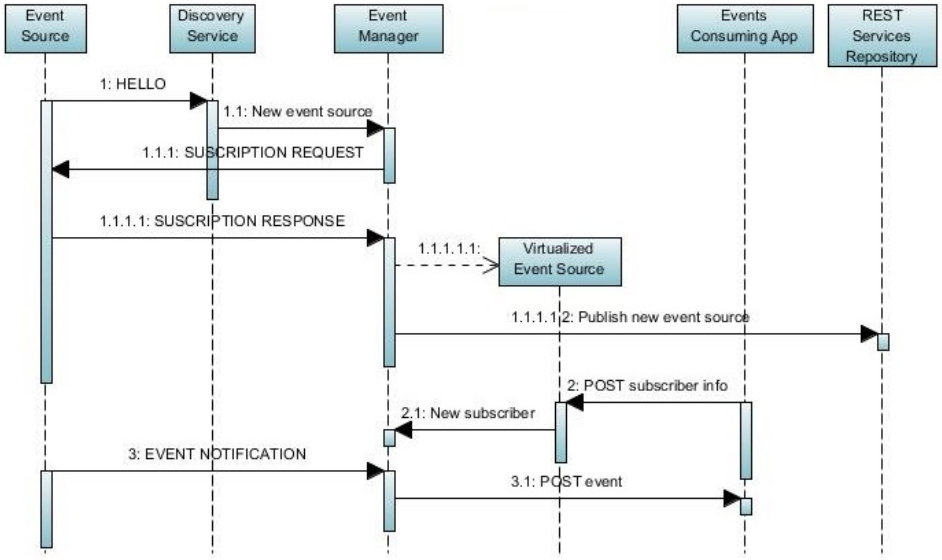
event manager acts as an event broker to the event consumers. The subscribers provide the event manager with a callback function that is also modeled as a REST service. This step corresponds to messages 2.X in **Figure 2**.

Table 1. Event manager resources description

URI	Method	Description
Sources	POST	Register a new source.
	GET	Retrieve the list of registered sources.
sources/{sourceID}	GET	Retrieve the information of "sourceID".
	PUT	Update the information of "sourceID".
	DELETE	Remove the registered entry of "sourceID".
subscribers	POST	Register a new subscriber.
	GET	Retrieve a list of registered subscribers.
subscribers/{subscriberID}	GET	Retrieve the information of "subscriberID".
	PUT	Update the information of "subscriberID".
	DELETE	Remove the registered entry of "subscriberID".

Finally, when an event is generated in the event source, it is notified first to the event manager by a lightweight message using again the nSOM middleware. The event manager then resends the notification to the event consumers using the callback function, which is also a REST service, using a POST method. This step corresponds to messages 3.X in **Figure 2**. All the information interchanged using REST is JSON based.

Figure 2. Event source virtualization and event notification processing



In the above figure, *Event Source* refers to an event source in a WSN and *Discovery Service*, and *REST Services Repository* refer to nSOM elements running in the gateway.

The representation of the event source as a REST service in the gateway is part of the whole virtualization model that has been developed by the UPM in WoO for integrating WSN as objects in the IoT domain using nSOM.

3. Results and Discussion

The model has been tested successfully in a real scenario that is based on an open smart neighborhood. SunSPOT devices equipped with a sensor board capable of providing temperature and luminosity readings, as well as interfacing with external sensors, have been used. One of these sensors were programmed to virtualize a heat detector for use in fire detection, composing it from temperature services running in other nodes, taking as a reference the operational specifications recommended by the European Normative EN 54-5:2000 [9]. Other SunSPOT node was provided with an external proximity sensor. It was programmed as a presence detector. Both were defined as event sources, and notified the appropriate notifications to the event manager on the gateway whenever they detect an event condition. Besides these event sources, a set of other sensor nodes where deployed to test the validity of the scenario.

The gateway was a PC with Ubuntu 12.04 LTS, and a SunSPOTbasestation connected to an USB port to provide connectivity with the deployed SunSPOT nodes by means of an IEEE 802.15.4 radio interface. The nSOM middleware was integrated into an ESB platform, Fuse ESB Enterprise 7.1, translating the messages incoming from the WSN to the corresponding REST services, and if required, the REST invocations to the corresponding nSOM messages to be casted to the WSN. The elements follow the model previously described. In the ESB two bundles are provided for the basic functionalities of interacting with nSOM capable devices in a WSN, and to act as an event manager. The information regarding event sources, consumers and events are managed dynamically and stored in a MySQL database.

This deployment has been integrated with other WoO partners' contributions in the scope of the open smart neighborhood scenario. In the tests that were conducted, the event manager was able to detect correctly the two types of event sources, and to represent them as REST resources. All the event notifications from the WSN were also captured succesfully by the event manager, and redirected to the event subscribers. All this process was done in a small fraction of the time requirements for the project.

4. Conclusions

An event-driven system has been successfully modelled and integrated in a REST solution for an IoT application, in a real pilot in the WoO project framework. The capabilities of different pilot devices from different partners were modelled as resources, according also to IoT-A recommendations, and applications and devices interacted readily. The results show that devices virtualization, using REST and ontologies for semantically annotating their description, leverages the IoT development, covering also event-driven devices and minimizing the negative impact on the network performance of the messages concerning subscription and event notification.

A virtual event-source is an interesting research issue we are working on now. A node in the WSN runs a small orchestrator that will look up for specific event sources, subscribe to its notification service, and compose a new kind of event that will be triggered whenever a certain sequence of events arises, becoming an event-source therefore.

Acknowledgments

The European project "Web of Objects" (WoO) (project code ITEA2-10028) and the Spanish ministry "Ministerio de Industria, Energía y Turismo" (project code TSI-020400-2011-29), support this work.

Author Contributions

Néstor Lucas, José-Fernán Martínez and Vicente Hernández were responsible for the theoretical analysis of the virtualization middleware. Néstor Lucas and Vicente Hernández designed and implemented the event source virtualization and other key components of the virtualization middleware. The platform for WSN virtualization is being developed by the UPM research group of Next-Generation Networks and Services (GRyS), where the authors belong to.

Conflicts of Interest

The authors declare no conflict of interest.

References and Notes

1. Agrawal, S.; Das, M.L. In *Internet of Things — A paradigm shift of future Internet applications*, Engineering (NUiCONE), 2011 Nirma University International Conference on, 8-10 Dec. 2011, 2011; 2011; pp. 1-7.
2. IoT-A. Internet of Things Architecture. Available online: http://www.iot-a.eu/public (10 January 2014),
3. Bauer, M.; Boussard, M.; Bui, M.; Carrez, F.; Jardak, C.; De Loof, J.; Magerkurth, C.; Meissner, S.; Nettsträter, A.; Olivereau, A.; Thoma, M.; Walewski, M.W.; Stefa, J.; Salinas, A. *Deliverable D1.5 — Final architectural reference model for the IoT v3.0*; IoT-A: 2013; p.^pp.
4. Bauer, M.; Boussard, M.; Bui, M.; Carrez, F.; Jardak, C.; De Loof, J.; Magerkurth, C.; Meissner, S.; Nettsträter, A.; Olivereau, A.; Thoma, M.; Walewski, M.W.; Stefa, J.; Salinas, A., Deliverable D1.5 — Final architectural reference model for the IoT v3.0. In IoT-A: 2013.
5. Reference Architecture Foundation for Service Oriented Architecture Version 1.0. Available online: http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/csprd02/soa-ra-v1.0-csprd02.pdf (July 2013),
6. Fielding, R.T., Architectural Styles and the Design of Network-based Software Architectures. In *Doctoral dissertation*, University of California: Irvine, 2000.
7. Akyildiz, I.F.; Su, W.; Sankarasubramaniam, Y.; Cayirci, E. Wireless Sensor Networks: A Survey. *Computer Networks* **2002**, *38*, 393-422.
8. Li, L.; Chou, W. In *Design Patterns for RESTful Communication Web Services*, Web Services (ICWS), 2010 IEEE International Conference on, 2010; 2010; pp. 512-519.
9. AENOR *Fire detection and fire alarm systems - Part 5: Heat detectors - Point detectors.*; European Committee for Standardization: 2000; p.^pp.