SenseBox – A Generic Sensor Platform for the Web of Things

Arne Bröring[1,2,3], Albert Remke[1,3], and Damian Lasnia[1]

[1] Institute for Geoinformatics, University of Münster,
Weseler Str. 253, Münster, Germany
{arneb,a.remke,d.lasnia}@uni-muenster.de
http://swsl.uni-muenster.de
[2] ITC Faculty, University of Twente,
Enschede, The Netherlands
[3] 52°North Initiative for Geospatial Open Source Software GmbH,
Martin-Luther-King-Weg 24, Münster, Germany

Abstract. Applications of the Web of Things reach from smart shoes posting your running performance online, over the localization of goods in the production chain, to computing the insurance cost of cars based on the actually driven kilometers. Thereby, Web of Things applications follow the REST paradigm, i.e. access to things and their properties is offered via REST APIs. This allows an easy meshing of web-enabled things into existing Web applications. This work introduces the SenseBox, a small computing device equipped (1) with different sensors to perceive its environment and (2) with a Web server and an according REST API which makes it available as a first class citizen on the Web. In an example use case of this generic sensor platform, the SenseBox is deployed next to a road and its in-built ultra sonic sensor is used to detect the number of bypassing cars and eventually determine the traffic density.

Keywords: Web of Things, Geosensors, Sensor Integration.

1 Introduction

The Web of Things (WoT) is about connecting real world objects to the Web. These objects have an identity represented by a URI and they are able to communicate with people, machines or other objects, e.g. by responding to an http GET request. They have a memory as to store and to provide information both about themselves and about the context they are living within.

Objects may have additional capabilities. They may act as sensors, taking up certain stimuli and transform them into useful observations. Or they are able to do something, e.g. open and close valves or move to a certain location. Depending on the degree of control by operators these objects act more or less autonomously. All activities require a certain amount of intelligence, i.e. some kind of situational awareness, the ability to conclude and to solve problems. This general pattern of an intelligent object is depicted in Figure 1.

A. Puiatti et al. (Eds.): MobiQuitous 2011, LNICST 104, pp. 186–196, 2012.

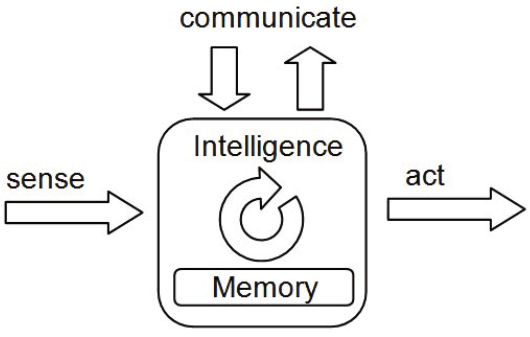

Fig. 1. The intelligent object patten

We are particularly interested in web enabled sensor objects, which are able to provide aggregated spatiotemporal information about some environmental phenomena. The deduced research question is: Does the WoT paradigm facilitate the provision and use of spatiotemporal sensor data?

The SenseBox project investigates certain real world use cases for web enabled sensor objects as to explore interaction patterns and practical requirements. Aim is to develop a generic sensor platform which is easily deployable in an on-the-fly manner and available as a thing on the Web. After introducing the concept of the Web of Things and the research field of integrating sensors with the Web (Section 2) this paper outlines the Traffic SenseBox case study (Section 3) and describes in detail the SenseBox architecture and its communication interface (Section 4). In Section 5, we compare the developed approach to related work. The paper ends with conclusion and an outlook to future work (Section 6).

2 Background

The vision of the two related research fields of Internet of Things [1] and Web of Things [2] is on integrating general, real-world things with the Internet or Web, respectively. Examples for such things are household appliances, embedded and mobile devices, but also smart sensing devices. Often, the user interaction takes place through a cell phone acting as the mediator within the triangle of human, thing, and Internet / Web. The application fields of the Internet of Things are influenced by the idea of ubiquitous computing [3]. They reach from smart shoes posting your running performance online, over management of logistics (e.g., localization of goods in the production chain), to insurance (e.g., car insurance costs based on the actually driven kilometres).

For technically realizing the Internet of Things, research topics include protocol stacks for the Internet Protocol (IP) standard optimized for smart things (e.g., IPv6, 6LoWPAN) [4], naming services for things [5], or the unique identification of objects (e.g., RFID). The Web of Things can be seen as an evolvement

of the Internet of Things. It leverages existing Web protocols as a common language for real objects to interact with each other. HTTP is used as an application protocol rather than a transport protocol as it is generally the case in web service infrastructures. Things are addressed by URLs and their functionality is accessed through the well-defined HTTP operations (GET, POST, PUT, etc.). Hence, Web of Things applications follow the REST paradigm [6]. Specific frameworks (e.g. [7], or [8]) offer REST APIs to enable access to things and their properties as resources. These REST APIs may not only be used to interact with a thing via the Web, also website representations of things may be provided to display dynamically generated visualizations of data gathered by the thing. Then, the mash-up paradigm and tools from the Web 2.0 realm can be applied to easily build new applications. An example application may use Twitter to announce the status of a washing machine or may let a fridge post to an Atom feed to declare which groceries are about to run out.

An already established approach for integrating sensors with the Web is the Sensor Web Enablement (SWE) framework defined by the Open Geospatial Consortium (OGC); its current state is described in [9]. SWE specifications are very generic and powerful since intended use cases are broad and often complex (e.g. disaster management or early warning systems). However, use cases, which do not need the full functionality on data filtering, sensor discovery, tasking and event handling as provided by SWE, may be easier to realize with the Web of Things approach by considering sensor objects as things. Hence, this work investigates the application of Web of Things principles to a simple sensor platform – the SenseBox.

3 The *Traffic* SenseBox Case Study

Strassen.NRW[1] is a state government enterprise, which is responsible for the construction and maintenance of about 20,000 kilometers of the road network in North-Rhine Westfalia, Germany. One of its tasks is the traffic management, i.e. monitoring, forecasting, and controlling the traffic flow as to assure mobility and safety on its road network. Strassen.NRW maintains a comprehensive telematics infrastructure as to gather real time information on e.g. traffic density and weather conditions and to control special devices such as electronic road signs or route guiding displays. The metering of traffic flow is based mainly on the use of statically mounted induction loops, which count the passing vehicles. But this sensor network is not dense enough for real time monitoring of the traffic situation in case of accidents and daily moving road works. That is the reason why Strassen.NRW is interested in additional means for collecting real time information on the traffic flow. Thus, we evaluate the following use case:

A specialized Traffic SenseBox is mounted to the safety truck securing mobile road works (e.g. for maintaining batters and guardrails). This SenseBox is able to count passing vehicles by utilizing an ultrasonic sensing unit capable of

[1] `http://strassen.nrw.de`

measuring distances to objects in front of it (Figure 2). Once switched on, the preconfigured SenseBox automatically connects through the mobile communication network to the Web. Once available on the Web, it can register as a new traffic sensor at the registry of the telematics infrastructure of Strassen.NRW.

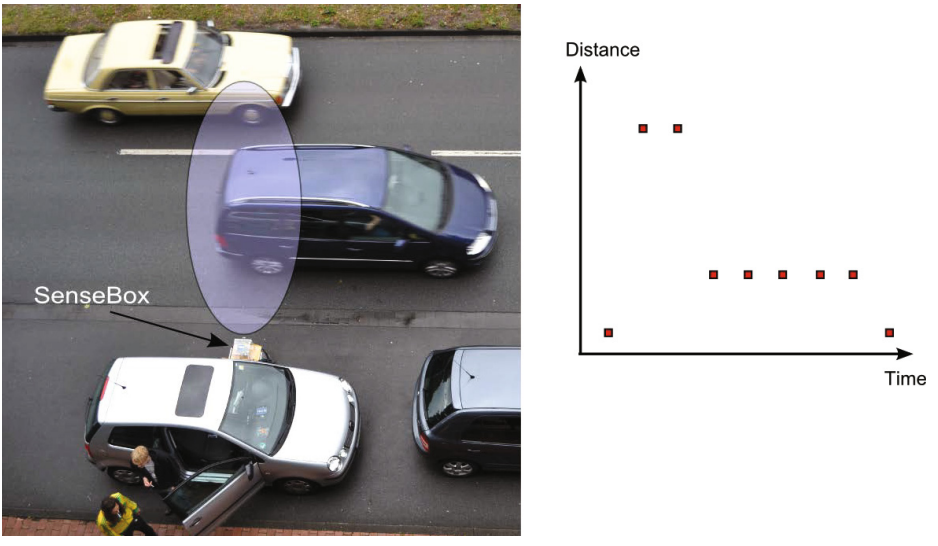

Fig. 2. The Traffic SenseBox test environment

The squad leader of the mobile road work has than the option to access the SenseBox through his mobile device, e.g. to check if it is working properly. He may even add further information on the situational context (e.g. the type of mobile road work, operator of the SenseBox). At the same time the operator of the Strassen.NRW traffic management system notices a new sensor item popping up on his map display, symbolizing the type of sensor and the current traffic flow. He is able to select this item to get further information such as the number of vehicles currently passing by. Also, Police cars could be equipped with Traffic SenseBoxes. In case of an accident, SenseBoxes could be placed together with warning signs in a certain distance to the accident location.

The organization, which owns a Traffic SenseBox decides about granting access to this device and thus about sharing its capabilities. Configuring and tasking sensors will be handled restrictively. Requesting information from sensors may be open to other institutions such as local authorities, automobile associations or private companies, which maintain their own traffic information systems. It has to be investigated in further detail, if direct access to sensors is valuable enough or if aggregated and quality assured information is preferred, which would be rather a service of the sensor owning organization.

4 The SenseBox - Requirements and Design

The technical design of the SenseBox needs to be generic and flexible so that it can be utilized in different use cases. Outdoor deployments, e.g. the detection of traffic density (Section 3), require the SenseBox to be robust against external influences such as mechanical stress, dust, temperature, humidity. Furthermore, it needs to make reliable observations under varying environmental conditions concerning for example weather, traffic, or position of the sensor. Since non-technicians deploy a SenseBox, this process needs to be easy. Together, those requirements shall be achieved with low costs.

The SenseBox is conceptualized as a physically and logically autonomous unit. As to be discoverable it has to self-register at predefined registries. It has to allow for changes both to its configuration and to its resource/data model as to be adaptable to various use cases. As for supporting privacy issues securing access is mandatory for most of the use cases. While the application protocol for web enabled objects is HTTP by definition, the SenseBox has to support a variety of communication patterns (e.g. pull, push) and low level protocols (e.g. UMTS, IEEE 802.11) as to support communication under various conditions.

4.1 The SenseBox Hardware and Deployment Setup

The SenseBox prototype is designed to operate on 12V and 24V batteries or car lighter sockets, as used in case of the Traffic SenseBox. To achieve a compact design a Mini-ITX low-power motherboard[2] is chosen which allows an integration of the needed hardware components within 15x15x15 centimeters. The motherboard is equipped with a 1.66 GHz Intel Atom processor, 2GB RAM and 8GB CompactFlash mass storage for data processing, data storage and as web server platform. The web connectivity is established by a UMTS/3G USB device so that all collected and processed data is available on the web in real time.

The web accessible data within our prototype setup is mostly derived from the two sensor outputs, GPS and ultrasonic sensor. The Arduino framework[3], consisting of microcontroller board hardware and the firmware programming language, is utilized to integrate the sensor components with the web enabled computing component. The Arduino Uno microcontroller board provides the capability to connect and configure a variety of different sensors in an easy way, and to preprocess the sensor outputs and send them to the main computing module. The ultrasonic range finder is configured to measure the distance between the near side lane and passing vehicles while the GPS component provides location and time awareness, e.g., to georeference the other sensor's outputs. Figure 3 shows the assembled SenseBox hardware prototype which has been tested in the car counting scenario as shown in Figure 2.

[2] http://www.mini-itx.com/
[3] http://www.arduino.cc/

Fig. 3. The SenseBox hardware prototype

4.2 The SenseBox Software Architecture

The design of the SenseBox software architecture is shown in Figure 4. In its center, the integrator component triggering sub-components to provide communication interfaces as well as interfaces to sensors and the database. The interface to sensors is established by a well-defined and generic message protocol for data coming from the Arduino board to which the sensors are connected. The transformator component retrieves these raw sensor data and is able to translate them to observations with more descriptive metadata (e.g. unit of measure or links to definitions of observed phenomena). Further, the transformator can apply specialized processing steps to certain incoming sensor streams to derive higher-level information. For example in the use case of this work (Section 3), the number of cars passing by the SenseBox is calculated from the distances measured by the ultrasonic sensor. Once assessed, this event of a new observation is broadcasted to the other components of the business logic; among them, the database handler receives the event and stores the observation in a database. Henceforth, the observation can be accessed via the REST interface as described in the following (Section 4.3).

4.3 A REST API for SenseBoxes

In order to provide data gathered by the SenseBox to other Web applications, a REST API for accessing measured observations is designed based on previous work we have done on meaningful URI schemes for sensor data [10].

A representation of a SenseBox can be accessed by following the URI *http://my.authority.org/boxes/<id>*. A list of observations gathered by the

sensors of this SenseBox can be retrieved by appending the URI segment */observations*. To represent single observations as XML we rely on the established Observations & Measurements standard [11]. Single observations are accessed by following the URI scheme *http://my.authority.org/boxes/<id>/observations/<observation id>*.

When dealing with sensor data for different thematic properties which are varying over time and space, it is important to offer according filter mechanisms to enable the retrieval of certain data subsets. For example, a reference to all car count observations of a SenseBox is given by appending an according query selecting this property filter: *http://my.authority.org/boxes/7/observations ?property=car_count*. Multiple property identifiers can be appended. By adhering to the proposal of [12] for a sound URI scheme, these multiple identifiers are separated by semicolons, as their order does not matter.

To refer to observations from a particular time instant or period, the time query token can be appended to the URI followed by two comma-separated time strings encoded according to the ISO 8601 specification [13]. For example, the URI *http://my.authority.org/boxes/7/observations ?time=2011-01-10T14:00, 2011-02-11T16:00 &property=car_count* points to the observation collection with all car count observations taken by SenseBox '7' from January 1st 2011 at 2pm until February at 4pm. The first time string represents the start date of the time period for which observation should be returned, the second indicates the end of the time period. The second time string can also be omitted when a link to observations of a particular time instant has to be specified.

As a spatial filter, a bounding box can be appended to the URI. We use commas to separate the ordered parameters forming a bounding box. The first four values are the coordinates defining a two dimensional rectangle, while the fifth value is the identifier of their coordinate reference system: *<minCoord1>, <minCoord2>,<maxCoord1>,<maxCoord2>,<crsURI>*. This spatial filter is either applied to the last position of the SenseBox or if the time parameter is specified as well, is applied to the position of the SenseBox at the defined time. An example of a URI using a bounding box filter is specified as follows: *http://my.authority.org/ boxes/7/observations ?bbox=3,6,23,36, urn:ogc:def:crs:EPSG:6.5:4326*.

5 Related Work

Already in 2002, Delin [14] defined a concept with similar characteristics to the Web of Things paradigm, the so-called Sensor Web, a heterogeneous pool of distributed web accessible sensor nodes. His approach described a set of micro-power, micro-bandwidth, and micro-cost sensor pods that are inter-connected and designed and manufactured along well-defined use cases. Today, these characteristics are provided by mainstream wireless sensing networks utilizing the *Motes* concept. Motes are small-sized sensor nodes with limited computing capabilities and able to connect with each other within a certain connectivity range to monitor phenomena of interest. Recent research and manufacturing activities

focused on the advancement of such sensor network technology by enhancing intelligent nodes that are equipped with more processing-power [15]. For example, Oracle offers a ready-to-use development kit consisting of hardware and software components for sensor nodes, called *Sun Spots*[4], with a 10 meter connectivity range. These generic nodes are capable of transforming Wireless Sensor Networks to Wireless Sensor Actuator Networks, because their nodes are able to change the behavior of the observing system [16], while former sensor networks were primarily designed to solely monitor their environment [17].

The SenseBox is similar to the approaches above; however, the hardware design is not primarily addressing bandwidth or processing limitations, hence, more expensive computations and storing capabilities (e.g., database for long-term sensor data storing) are available. Further, the SenseBox focuses on completely embedded web functionality and therefore provides a well-defined REST API offering a light-weight sensor data access protocol based on HTTP. This is for example contrary to the approach designed by Resch [15] that utilizes OGC's Sensor Web Enablement standards. Those standards [9] are richer in functionality but also more complex and hence more difficult to integrate with client applications.

Although, the Traffic SenseBox, as a first use case, has been primarily served as an example to demonstrate the generic SenseBox concept, related approaches for traffic control shall also be compared here. Table 1 gives an overview and comparison of traffic control systems currently in use. Besides object tracking based solutions which make use of the smartphone capabilities of car drivers, such as Waze[5], systems exist that need an additional physical infrastructure in form of sensor platforms mounted beyond or next to the street network. The latter once are either based on subsurface sensors (e.g. Canoga[6]) or are camera-based traffic control systems (e.g. Sensor Technologies[7]). Compared to those systems, the current design of the Traffic SenseBox with its single ultrasonic sensor has a clear drawback of poor measurement detail; it can only count cars, while the other systems are able to observe speed at all lanes simultaneously. However, due to the generic design of the SenseBox, future designs may include additional sensors (e.g. a second ultrasonic sensor) which would allow determining the speed of by-passing cars.

The SenseBox approach has advantages in terms of flexibile control over the deployed infrastructure. Road network administrators can decide in an ad-hoc manner where new SenseBoxes need to be deployed, e.g. in case of an accident or construction. This is not the case for subsurface sensor systems and also cameras are usually persistently mounted. Vehicle tracking via Web connected mobile phones and associated GPS sensors appears on first sight as the best solution. However, it lacks flexibility since road network managers have no control over the underlying infrastructure and are dependent on users who provide their

[4] http://www.sunspotworld.com
[5] http://world.waze.com/
[6] http://www.gtt.com/Products/CanogaTrafficSensingSystem
[7] http://www.sensor-tech.eu/sensor_tv_en.htm?Lang=EN

positional data. In addition, such object (or human) tracking based approaches are critical in terms of privacy, just like camera based approaches are. Thus, the Traffic SenseBox can be seen as a flexible and privacy-aware real time traffic control system.

Table 1. Comparsion of real time traffic control systems

	Physical Infrastructure needed	Measurement detail	Flexible control over infrastructure	Privacy
Object tracking (GSM, GPS)	Reusable (+)	+	o	-
Subsurface sensors	Yes (-)	+	-	+
Camera-based	Yes (-)	+	o	-
SenseBox	Yes (-)	-	+	+

6 Conclusions and Outlook

This work presents the SenseBox - a generic sensor platform focusing on an easy accessibility via the Web by following the Web of Things approach. The SenseBox's REST interface enables its integration as a first class citizen on the Web and includes thematic, spatial, as well as temporal filtering of observations gathered by its sensors. By utilizing the Observations & Measurements standard from OGC's SWE framework, we showed that the WoT approach can be integrated with the Sensor Web concept. We can conclude that the integration of sensors is facilitated by following the WoT paradigm, since HTTP is utilized by many applications. However, most of the logic is decentralized, meaning that sensors need to become rich in functionality - which is realizable for platforms such as the SenseBox, as it is particularly designed for those tasks.

Applying the SenseBox in the described use case of determining the traffic density by counting cars with an ultrasonic sensor has already shown satisfying results. However, the restricted range of the chosen ultrasonic sensor allows only counting cars on the nearby lane. In future, this use case setup will be refined, but also other use cases will be investigated (e.g., precision agriculture, air pollution monitoring, or noise assessment) to prove the genericness of the SenseBox. The full benefit of the SenseBox concept can be achieved as soon as such sensor resources are shared across organizations (e.g. Strassen.NRW, municipalities, or automobile clubs). Whether this is practically possible concerning political policies, business models, trust, etc. has to be investigated.

In future, we plan to evolve the SenseBox design by separating the platform logic from the specific sensor interfaces. To achieve this, the SenseBox will be combined with our previously developed Sensor Interface Descriptor concept [18] which allows the declarative description of sensor protocols to enable plug & play of sensors [19]. Furthermore, the REST interface will be extended in future so that the configuration of the SenseBox via HTTP PUT will be possible as well

as a push-based observation delivery. Also, other representations of observation resources shall be supported, such as the RDF format to make SenseBoxes available on the Linked Data Cloud [20], similar to our previous work on a RESTful proxy [21] for the Sensor Observation Service [22]. Another direction of research to be tackled is the inter-communication between SenseBoxes for more reliable observations. Based on the on-board GPS sensor, near SenseBoxes may be identified autonomously for initiating interaction via the Web.

Acknowledgment. This work is financially supported by the project *Flexible and Efficient Integration of Sensors and Sensor Web Services* funded by the ERDF program (grant N 114/2008) of the European Union, as well as the 52°North Sensor Web community (`http://52north.org/sensorWeb`) which envisions a real time integration of heterogeneous sensors into a coherent information infrastructure. Further, the authors thank the students of the University of Muenster who have contributed to the development of the SenseBox project: Dustin Demuth, Kristina Knoppe, Maurin Radtke, Arthur Rohrbach, Raimund Schnürer, Christopher Stephan, Umut Tas, and Jan Wirwahn.

References

1. Gershenfeld, N., Krikorian, R., Cohen, D.: The Internet of Things. Scientific American 291(4), 76–81 (2004)
2. Guinard, D., Trifa, V.: Towards the Web of Things: Web Mashups for Embedded Devices. In: International World Wide Web Conference. ACM, Madrid (2009)
3. Weiser, M.: The Computer for the 21st Century. Scientific American 265(9), 94–104 (1991)
4. Hui, J.W., Culler, D.E.: IP is Dead, Long Live IP for Wireless Sensor Networks. In: SenSys 2008: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, pp. 15–28. ACM, New York (2008)
5. EPCglobal: EPCglobal Object Name Service (ONS) 1.0.1. EPCglobal Inc. (2008)
6. Fielding, R.T., Taylor, R.N.: Principled Design of the Modern Web Architecture. ACM Transactions on Internet Technology 2(2), 115–150 (2002)
7. Pinto, J., Martins, R., Sousa, J.: Towards a REST-style Architecture for Networked Vehicles and Sensors. In: 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 745–750. IEEE, Mannheim (2010)
8. Ostermaier, B., Schlup, F., Romer, K.: WebPlug: A Framework for the Web of Things. In: 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops), pp. 690–695. IEEE, Mannheim (2010)
9. Bröring, A., Echterhoff, J., Jirka, S., Simonis, I., Everding, T., Stasch, C., Liang, S., Lemmens, R.: New Generation Sensor Web Enablement. Sensors 11(3), 2652–2699 (2011)
10. Janowicz, K., Bröring, A., Stasch, C., Everding, T.: Towards Meaningful URIs for Linked Sensor Data. In: Devaraju, A., Llaves, A., Maue, P., Kessler, C. (eds.) Towards Digital Earth: Search, Discover and Share Geospatial Data, Workshop at Future Internet Symposium, vol. 640. CEUR-WS, Berlin (2010)

11. Cox, S.: OGC Implementation Specification 07-022r1: Observations and Measurements - Part 1 - Observation schema. Open Geospatial Consortium, Wayland, MA, USA (2007)
12. Richardson, L., Ruby, S.: RESTful Web Services. O'Reilly Media, Inc. (2007)
13. ISO: ISO 8601:2004 - Data elements and interchange formats - Information interchange - Representation of dates and times. International Organization for Standardization (ISO), Geneva, Switzerland (2004)
14. Delin, K.: The Sensor Web: A Macro-Instrument for Coordinated Sensing. Sensors 2, 270–285 (2001)
15. Resch, B., Mittlboeck, M., Lippautz, M.: Pervasive Monitoring - An Intelligent Sensor Pod Approach for Standardised Measurement Infrastructures. Sensors 10(12), 11440–11467 (2010)
16. Xia, F., Tian, Y., Li, Y., Sung, Y.: Wireless sensor/actuator network design for mobile control applications. Sensors 7(10), 2157–2173 (2007)
17. Heidemann, J., Govindan, R.: Embedded sensor networks. In: Handbook of Networked and Embedded Control Systems, pp. 721–738 (2005)
18. Bröring, A., Below, S., Foerster, T.: Declarative Sensor Interface Descriptors for the Sensor Web. In: Brovelli, M., Dragicevic, S., Li, S., Veenendaal, B. (eds.) WebMGS 2010: 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services, vol. 38. ISPRS, Como (2010)
19. Bröring, A., Maué, P., Janowicz, K., Nüst, D., Malewski, C.: Semantically-enabled Sensor Plug & Play for the Sensor Web. Sensors 11(8), 7568–7605 (2011)
20. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story so far. Journal on Semantic Web and Information Systems 5(3), 1–22 (2009)
21. Janowicz, K., Bröring, A., Stasch, C., Schade, S., Everding, T., Llaves, A.: A RESTful Proxy and Data Model for Linked Sensor Data. International Journal of Digital Earth (2011) (in press)
22. Bröring, A., Stasch, C., Echterhoff, J.: OGC Interface Standard 10-037: Sensor Observation Service (SOS) 2.0 Interface Standard. Open Geospatial Consortium, Wayland, MA, USA (2010) (candidate standard)