A Citizen Science Sensor Platform as a Live Link from GIS to the Internet of Things

Dustin Demuth
d.demuth@52north.org

Arne Bröring
broering@52north.org

Albert Remke
remke@52north.org

52° North GmbH, Martin-Luther-King-Weg 24, 48155 Münster, Germany

Abstract

Making citizen science sensor platforms and their measured data accessible to GIS users can be a challenging task. Data has to be discovered, downloaded, and converted before adding it to a GIS project. To facilitate integration and enhance interoperability between sensor platforms and GIS, this work proposes a method for accessing sensor platforms directly through GIS. Our approach implements selected parts of the OGC GeoServices REST API directly on a sensor platform, thus making it a first class node on the Internet of Things. Users can get a live view of what is measured in the field, in their GIS.

Keywords: Internet of Things, Citizen Science, Data Acquisition, Sensor Web, GeoServices

1 Introduction

In the past years, communities of amateur enthusiasts have created more and more technical solutions for environmental monitoring, environmental control and data acquisition. A plethora of solutions already exists based on open hardware and software platforms (e.g., Arduino[1] or Wiring[2]) as well as the recent advances in the mobile devices market. The members of such communities who systematically collect data to foster additional knowledge on a certain subject or region can be called *citizen scientists* [9]. They use open hardware systems and smartphones as the basis for building sensor platforms to observe and measure their environment. Data generated by such sensor platforms is often pushed to Sensor Web portals [4], such as *cosm*[3] or *Thingspeak*[4], which offer functions for storing, sharing, visualizing, as well as discovering sensor data. Those Sensor Web platforms, which expose sensors as Web-accessible resources, form a part of the *Internet of Things* [1].

Despite existing, manifold functions of Sensor Web platforms, those services do not include Geographic Information System (GIS) functionalities, such as the overlay with map layers or the implementation of spatial operators. To carry out such analyses, the stored data needs to be converted for subsequent integration with a GIS. The integration is challenging. Due to insufficient meta data and the usage of non-standardized data formats, interoperability between the citizen scientists' sensor platforms and GIS tools is not existent. If addressing this challenge and improving interoperability, we could significantly increase the availability of sensor data collected by citizen scientists and foster its usage by GIS experts and decision makers.

In order to tackle the challenge described above, this paper proposes an approach to provide *direct* and *live* access to citizen science sensor platforms. The approach does not require a third party brokering platform which collects and gives access to sensor data. Instead, the sensor platforms become first class nodes on the Internet of Things and directly serve interfaces, standardized by the Open Geospatial Consortium (OGC).

The use cases for sensor platforms offering direct and live access through the Web are manifold. They include, but are not limited to, the detection of urban environmental phenomena such as changes of wind fields or higher pollutant concentrations, traffic monitoring and management [5], civil protection (e.g. radiation detection), forest fire and bush fire detection and their analysis and prevention, or precision agriculture [8]. In all cases amateurs or experts can contribute sensor platforms.

[1] http://www.arduino.cc
[2] http://www.wiring.org.co
[3] http://www.cosm.com
[4] http://www.thingspeak.com

The remainder of this paper is organized as follows; Section 2 is illuminates Citizen Science and the Web of Things, whereas Section 3 is depicts the implementation of a Feature Service on an internet enabled sensor platform. After a short discussion of the limitations of the proposed approach in Section 4, the paper closes with an outlook as Section 5.

2 Background

Today, we observe an increasing involvement of citizens into research projects [7]. Reasons for this can be found in the rising interest in the environment [7] and in the advancements of information and communication technologies. Affordable technical equipment (such as computers and smartphones), as well as open or freely available software enable citizens to take part in studies more easily, or even motivate them to contribute to professional studies, e.g. by donating processing time for a project such as Seti@home[5]. The increasing possibilities of global communication allow citizens to share their insights with a large number of people. Technologies, such as the *Internet of Things*, make it easy to share sensor data at a global scale.

The Internet of Things uses unique identification of things (e.g. based on barcodes or Radio Frequency Identification, RFID). The Cluster of European Research Projects on the Internet of Things states that the Internet of Things is a dynamic, global, self configuring and interoperable structure. Physical and virtual objects are identifiable and physical attributes have virtual personalities [6]. In summary, the Internet of Things can be understood as a network of physical things and their virtual representations which use the Internet protocols as transport mechanisms [24].

An extension of the Internet of Things is the *Web of Things* [17]. It focuses on enabling interaction between things and integrating them into the Web. It relies on established protocols and principles, such as HTTP (Hypertext Transport Protocol) [13] and Representational State Transfer (REST) [14]. In the Web of Things each thing is represented by a resource and is identifiable by a Uniform Resource Identifier (URI) [2]. This representation can be achieved by integrating lightweight Web servers into each thing [16]. Using Web technologies, things can be used like any other Web resource [21]. Both the Internet of Things and the Web of Things are structures influenced by users [18]. Emerging communities are creating hardware and software applications that fit their special needs and integrate them into the Internet/Web of Things.

This way, the Internet of Things also influences citizen science. While citizen science has formerly only been associated with activities such as habitat mapping or wildlife tracking, emerging projects now make use of environmental monitoring technologies. Affordable hardware solutions and good documentation of open hardware projects in particular drive this movement. New sensor platforms emerge, which can be used to measure for example weather related phenomena, air quality [10], or noise [19]. Those projects invite users to build their own measurement applications and share their design with the community.

An example for a citizen science project is the crowd-funded *Air Quality Egg*[6]. It is solely based on open hardware and software. By sharing the collected data publicly through Sensor Web portals, citizen scientists increase the amount of volunteered geographic information (VGI) [15]. User contributed geographic content leads to a wikification of GIS [3] and can help to see phenomena from a different perspective. Nevertheless, sharing data with professionals often requires extra efforts for citizen scientists, since expert tools are not available, crucial information on the acquired data is missing, scientific processes are unclear for the citizen, professionals simply considered data insignificant.

3 An Approach for *Live* and *Direct* Linking of GIS and the Internet of Things

In order to link experts, citizen scientists, their sensor platforms, and the data gathered by those platforms, mechanisms are required to enable an access to the data. This section proposes an approach to tackle this challenge, by identifying suitable hardware and proposing the use of standards-based methods to exchange information and to bind citizen platforms to the expert's GIS tools.

There are three options to consider for enabling citizen scientists to share gathered data.

First, data could be pushed to Sensor Web portals hosted by a third party, such as *cosm*. Although this is an easy-to-use and simple approach, many users have concerns about this option due to questions of data ownership, data availability, and potential costs. They would also prefer an interoperable, standards-based access to data to enable platform independency. Hence, we suggest the usage of OGC-compliant services for an integration with GIS tools.

The second option for citizen scientists is to publish their acquired data to OGC services which would be op-

[5] `http://setiathome.berkeley.edu/`

[6] `http://airqualityegg.com/`

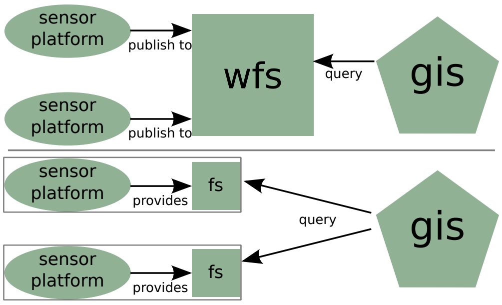

Figure 1: *Top*: each sensor platform publishes to an OGC-compliant Web Feature Service. *Bottom*: each sensor platform implements a feature service

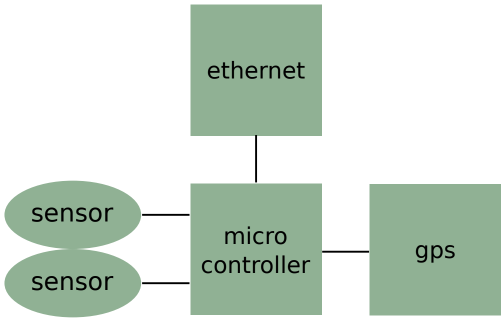

Figure 2: Hardware Components

erated by a group of volunteers or some organization (Figure 1 - top). Unfortunately, the operation of OGC services and the setting up of a publishing mechanism can be challenging and cost-intensive tasks for a small community.

A third option is the deployment of OGC-compliant services directly on each sensor platform (Figure 1 - bottom). Sensor platforms become directly accessible for GIS, since there is no detour through third party systems. A GIS is equipped with live access to sensors out in the field. Despite obvious disadvantages, such as lower performance, such an approach solves the issues of the two options sketched out above. The community keeps the ownership of the acquired data and there are no additional costs for third party systems.

3.1 Sensor Platform

The proposed implementation of a citizen science sensor platform consists of a micro controller board, a GPS, as well as a chipset providing Internet connectivity and sensors. The hardware components are depicted in Figure 2. The central component of the sensor platform is an Arduino Mega 2560[7] compatible micro controller board. Arduino is an electronic prototyping platform, consisting of the micro controller board, a boot loader that runs the board, and an IDE. The whole platform is open source, making clones and variations of the micro controller board possible. Well known amongst the enthusiasts of electronics, the Arduino architecture is already used in citizen science projects. The board provides interfaces on which sensors and actuators can be attached; its micro controller can be programmed in a flexible manner and runs a lightweight Web server. It also controls and evaluates the sensors and their readings. A GPS receiver is connected to the micro controller, so that information on the current time and position can be acquired. The platform can connect to the Internet through a chipset, which provides Ethernet functionality and a TCP/IP stack.

Once running, the sensor platform starts collecting data by reading the sensors in defined intervals. Each measurement is stored in a file which is identified by the sensor ID. Location and time derived from the GPS are added to each measurement.

3.2 Implemented Service

In late 2010, Esri[8] released its Open GeoServices REST API to the public [11, 12] and requested its standardization within the Open Geospatial Consortium (OGC) in early 2011 [20]. OGC formed a Standards Working Group (SWG)[9] to modify and implement the white paper as an OGC standard. Currently, the draft is not supported by most GIS applications, nevertheless the Esri JavaScript API[10] supports the RESTful GeoServices in their unmodified form.

Although the specification is designed to be implemented on a mature GIS Server, this paper shows how to implement its essential parts in order to offer the measurements of sensors connected to a sensor platform.

The platform only integrates the *Feature Service*, since the hardware is limited in processing power and sophisticated storage for images is not available. In contrast to the established OGC Web Feature Service [22], the GeoServices REST specification's default response is formatted in the JavaScript Object Notation (JSON). JSON is a lightweight alternative to the eXtensible Markup Language (XML). It claims to be less processing-intensive and to consume less bandwidth.

The GeoServices REST Feature Service specification

[7] `http://arduino.cc/en/Main/ArduinoBoardMega2560`

[8] `http://www.esri.com/`

[9] `http://www.opengeospatial.org/projects/groups/gservrestswg`

[10] `http://www.esri.com/getting-started/developers/get-started/javascript`

supports both tables and layers. Since a GPS is connected, we add a spatiotemporal data tag to each measurement and represent it as a feature. Measurements from the same sensor share the same feature type, which makes it possible to collect all measurements of this sensor in the same layer. As a result, it is not necessary to implement tables for this sensor platform.

The Feature Service has four different types of responses, if we omit error messages. These responses can be the collection of available layers (each layer represents one sensor), the detailed view of a layer's attributes (such as the layer's extent or the field definitions of the contained features), the collection of features belonging to a specified layer, or a single feature from a specified layer. In order to gain access to the measurements provided by the Feature Service, the service has to be queried. Querying is done by sending an HTTP GET request to the service URL. The service generates a response to the client's request. To query the service for its available sensors, the service's root URL has to be called. The service responds with the requested collection. Listing 1 depicts a possible response to the request `http://example.org/geoservices/`. The response contains the description of the requested service and an array of layers.

```
{
  "serviceDescription": "RESTful GeoServices
      SenseBox",
  "layers": [
    {
    "id": "1",
    "name": "carbonmonoxide"
    }
  ]
}
```

Listing 1: The service response

To receive information on a layer, the layer can be queried. This is done by adding the layer's id (in this case `1`) to the services root URL. Then, the requested URL for the layer is `http://example.org/geoservices/1/`. If the service knows the layer identified by the ID, it responds with a JSON object. This object is shown in listing 2.

```
{
  "id": "1",
  "type": "Feature Layer",
  "displayField": "value",
  "capabilities": "Query",
  "geometryType": "GeometryPoint",
  "minScale": 0,
  "maxScale": 0,
  "spatialReference": {
      "wkid": 4326
  },
  "objectIdField": "objectid",
  "fields": [
    {
      "name": "objectid",
      "type": "FieldTypeOID",
      "alias": "Object ID"
    },
    <abbreviated>
  ]
}
```

Listing 2: The layer response

The response contains an array of the layer's field definitions, transmitted geometry's definition and the spatial reference system. It also includes additional information which is valuable for the GIS to display the layer correctly.

The layer's features can be listed by adding the term `query` to the URL. The request `http://example.org/geoservices/1/query` would lead to the response shown in listing 3.

```
{
  "objectIdFieldName": "objectid",
  "geometryType": "GeometryPoint",
  "spatialReference": {
    "wkid": 4326
  },
  "fields": [
    {
      "name": "objectid",
      "type": "FieldTypeOID",
      "alias": "Object ID"
    },
    <abbreviated>
  ],
  "features": [
    {
      "geometry": {
        "point": {
          "x": 7.652118,
          "y": 51.934969
        },
        "spatialReference": {
          "wkid": 4326
        }
      },
      "attributes": {
        "ObjectID": "20000101000003",
        "Time": "2000-01-01T00:00:03Z",
        "Value": "5"
      }
    },
    <abbreviated>
    }
  ]
}
```

Listing 3: A collection of features

The response contains an array of field definitions of the layer and an array of features, as well as the definition of the transmitted geometry and the spatial reference system.

The last possible request is the request for a certain feature. This is done by adding the feature ID to the layer URL. A GET request to `http://example.org/geoservices/1/20000101000018` would result in the response shown in listing 4.

```
{
  "feature": {
    "geometry": {
      "point": {
        "x": 7.652118,
        "y": 51.934969
      },
      "spatialReference": {
        "wkid": "4326"
      }
    },
    "attributes": {
      "ObjectID": "20000101000018",
      "Value": "5"
    }
  }
}
```

Listing 4: A feature

The specification defines possibilities to narrow down the collection of features delivered by letting the Feature Service filter the response before sending it to the client. Such filters can be, for example, spatial relations (e.g. intersects, contains, ...), geometries (e.g. point: x,y), or filters on attributes (e.g. 'where value=5'). In addition, it specifies parameters to enable spatial reference system transformations. This implementation is not capable of doing such filtering and transformations, due to processing power constraints. It will always respond with a complete set of features.

4 Discussion

Due to the low-cost hardware used by our approach, the proposed solution has certain limitations.

The hardware used is currently not capable of multithreading and the program flow is linear. Missing multithreading can cause that some measurements can not be made. This especially happens when the platform has to deliver a long response and is occupied with processing and sending data to the client. No measurement can be taken until the client's request is fulfilled. The platform is also limited in processing speed and memory, thus it only allows simple queries. Due to the persistent internet connection, the sensor platform requires a higher amount of energy than platforms without an internet connection.

Nevertheless, due to the high pace in hardware development, it is to be expected that those challenges can be dealt with in the future. Similar hardware with much higher performance is already on the market[11,12]. Current advances in energy harvesting, the generation of energy from the environment (renewable energy, harvesting electromagnetic radiation from radio services [23]), can support our approach in the future.

5 Conclusions and Outlook

This work has shown that sensor platforms can easily be exposed as nodes within the Internet of Things. By implementing the OGC GeoServices REST API, citizen scientists can make their sensor platforms available to GIS users without having to deal with sophisticated data publishing infrastructures. The proposed direct access enables a live view from GIS into the physical world.

Although we have discussed our current prototype's limitations in performance and energy consumption, it is expected that future hardware improvements will foster our approach. The implemented solution is easy to deploy on sensor platforms by citizen scientists. If necessary, the flexible Arduino infrastructure allows the user to carry out a simple adaptation to his own needs.

Additional efforts have to be made to create possibilities to remotely configure the sensor platform. These efforts include the evaluation of methods for securing access to the platform's data and configuration. In addition, future platforms should offer functions to support remote sensor calibration.

Acknowledgments

This work has been partially supported by the GLUES project funded by the German Federal Ministry of Education and Research, the Google Summer of Code 2012 programme, as well as the 52° North Initiative for Geospatial Open Source Software.

References

[1] L. Atzori, A. Iera, and G. Morabito. The Internet of Things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.

[2] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. RFC 3986 (Standard), Jan. 2005.

[11] `http://arduino.cc/en/Main/ArduinoBoardDue`
[12] `http://www.raspberrypi.org/`

[3] M. Boulos. Web gis in practice iii: creating a simple interactive map of england's strategic health authorities using google maps api, google earth kml, and msn virtual earth map control. *International Journal of Health Geographics*, 4(1):22, 2005.

[4] A. Bröring, J. Echterhoff, S. Jirka, I. Simonis, T. Everding, C. Stasch, S. Liang, and R. Lemmens. New Generation Sensor Web Enablement. *Sensors*, 11(3):2652–2699, 2011.

[5] A. Bröring, A. Remke, and D. Lasnia. Sensebox - a generic sensor platform for the web of things. In *8th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous 2011)*, Springer LNICST, 2011.

[6] CERP-IoT. Internet of things - strategic research roadmap. Technical report, GRIFS-project, 09 2009.

[7] C. Conrad and K. Hilchey. A review of citizen science and community-based environmental monitoring: issues and opportunities. *Environmental Monitoring and Assessment*, 176:273–291, 2011. 10.1007/s10661-010-1582-5.

[8] D. Demuth. A Web of Things integrated Sensor Platform for Precision Agriculture, March 2012. Bachelors Thesis.

[9] J. Dickinson, R. Louv, and R. Bonney. *Citizen Science: Public Participation in Environmental Research*. Cornell University Press, 2012.

[10] P. Dutta, P. M. Aoki, N. Kumar, A. Mainwaring, C. Myers, W. Willett, and A. Woodruff. Common sense: participatory urban sensing using a network of handheld air quality monitors. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 349–350, New York, NY, USA, 2009. ACM.

[11] ESRI. Esri releases the open geoservices rest specification. *ArcNews Online*, 2010. Online available: `http://www.esri.com/news/arcnews/fall10articles/esri-releases.html`; last accessed 04/2013.

[12] ESRI. Geoservices rest specification version 1.0. *Esri White Paper*, September 2010. Online available: `http://www.esri.com/library/whitepapers/pdfs/geoservices-rest-spec.pdf`; last accessed 04/2013.

[13] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard), June 1999. Updated by RFCs 2817, 5785, 6266.

[14] R. T. Fielding and R. N. Taylor. Principled design of the modern web architecture. In *Proceedings of the 22nd international conference on Software engineering*, ICSE '00, pages 407–416, New York, NY, USA, 2000. ACM.

[15] M. Goodchild. Citizens as sensors: the world of volunteered geography. *GeoJournal*, 69:211–221, 2007. 10.1007/s10708-007-9111-y.

[16] D. Guinard and V. Trifa. Towards the web of things: Web mashups for embedded devices. In *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences)*, Madrid, Spain, Apr. 2009.

[17] D. Guinard, V. Trifa, F. Mattern, and E. Wilde. From the internet of things to the web of things: Resource oriented architecture and best practices. In D. Uckelmann, M. Harrison, and F. Michahelles, editors, *Architecting the Internet of Things*, chapter 5, pages 97–129. Springer, New York Dordrecht Heidelberg London, Apr. 2011.

[18] G. Kortuem and F. Kawsar. Market-based user innovation in the internet of things. In *Internet of Things (IOT), 2010*, pages 1 –8, 29 2010-dec. 1 2010.

[19] N. Maisonneuve, M. Stevens, M. E. Niessen, P. Hanappe, and L. Steels. Citizen noise pollution monitoring. In *Proceedings of the 10th Annual International Conference on Digital Government Research: Social Networks: Making Connections between Citizens, Data and Government*, dg.o '09, pages 96–103. Digital Government Society of North America, 2009.

[20] Open Geospatial Consortium. OGC seeks comments on geoservices rest api candidate standard. 2011. Online available: `http://www.opengeospatial.org/standards/requests/89`; last accessed 04/2013.

[21] V. Stirbu. Towards a restful plug and play experience in the web of things. In *Semantic Computing, 2008 IEEE International Conference on*, pages 512 –517, aug. 2008.

[22] P. A. Vretanos. *OGC Implementation Specification 04-094: Web Feature Service Implementation Specification*. Open Geospatial Consortium, Wayland, MA, USA, 2005.

[23] R. Vullers, R. van Schaijk, I. Doms, C. V. Hoof, and R. Mertens. Micropower energy harvesting. *Solid-State Electronics*, 53(7):684 – 693, 2009. Papers Selected from the 38th European Solid-State Device Research Conference – ESSDERC'08.

[24] E. Wilde. Putting things to rest. UCB iSchool Report 2007-015, 2007.