Short Paper: Citizen Sensing within a Real-Time Passenger Information System

David Corsar, Peter Edwards, Chris Baillie, Milan Markovic,
Konstantinos Papangelis, and John Nelson

dot.rural Digital Economy Hub,
University of Aberdeen, Aberdeen, UK
{dcorsar,p.edwards,c.baillie,m.markovic,
k.papangelis,j.d.nelson}@abdn.ac.uk
http://www.dotrural.ac.uk

Abstract. *GetThere* is a real-time passenger information system (RTPI) for rural areas that uses a citizen sensing approach to acquire information from public transport users. This paper describes the use of ontologies in *GetThere* to represent and integrate citizen sensors with data required to provide RTPI (e.g. timetable and route descriptions). The service architecture used to manage semantic sensor data is also described.

Keywords: Citizen sensing, semantic sensors, ontology, quality, provenance, transport

1 Introduction

We are developing an information ecosystem within the Informed Rural Passenger project[1] to support *GetThere*, a real-time passenger information (RTPI) system for rural areas. This ecosystem is based on an ontological framework that describes the datasets necessary to support the provision of RTPI, such as estimated vehicle arrival times or notification of delays. A key aspect is the integration of sensor data (e.g. vehicle locations from GPS) with other data (e.g. timetable and route descriptions). However, it is not uncommon in rural areas to experience situations in which appropriate sensors are unavailable (e.g. rural buses are not typically equipped with GPS devices). To address this, we have adopted a citizen sensing approach [7], i.e. enabling humans (in our case, public transport users) to act as data sensors using the *GetThere* smartphone app[2]. However, this introduces new issues associated with the quality of such observations due to malicious users, inaccurate devices, and erroneous observations.

This paper discusses the citizen sensing aspects of *GetThere*. Section 2 describes the ontologies used to represent and integrate citizen sensor data, and the service architecture built to manage semantic sensor data; section 3 discusses its deployment within the *GetThere* system; section 4 discusses the performance of the architecture; and section 5 outlines conclusions and future work.

---

[1] http://www.dotrural.ac.uk/irp
[2] http://www.gettherebus.com

2   Supporting RTPI via Citizen Sensing

During the initial design of *GetThere* and from experience of trialling the system, a number of requirements were identified to enable citizen sensing to be used to support RTPI provision using the ecosystem. These include: semantic integration of sensor descriptions and observations with other data within the ecosystem; following best practise and reusing existing (sensor) ontologies to describe (citizen) sensor data; recording the provenance of observations; management of observations and sensor descriptions via a RESTful API (create, retrieve, update, and delete); and assessing the quality of user observations.

2.1   Ontologies

Fig. 1 outlines the ontological framework used to integrate citizen sensors within the ecosystem. This framework is designed to support a range of transport applications in different geographic areas. The W3C Semantic Sensor Network (SSN) Incubator Group ontology[3] forms the basis of the framework. This ontology provides a generic model for describing *Sensor*s, the *Sensing* methods they implement, *Observation*s (value for a property of a phenomenon), *Sensor Output*s generated by sensors, *Observation Value*s, and *Feature Of Interest*s (real world phenomena being observed) [5].

The Travel Sensors ontology[4] extends SSN to represent the sensor concepts present in *GetThere*. This includes defining users (FOAF[5] *Agent*s) as sensors and several types of observation that they can provide (e.g. vehicle occupancy level, vehicle temperature). Mobile devices running the *GetThere* app are represented as platforms, with several attached sensing devices, each with their own sensing and observation classes allowing the representation of observations produced by the *GetThere* app (e.g. location, ambient noise level, presence of Wi-Fi) on behalf of the user. Along with classes (such as those shown in Fig. 1), the ontology also defines cardinality constraints on the SSN properties to, for example, ensure *OccupancyLevelObservations* are only *observedBy* a FOAF *Agent*.

Observations from users are integrated within the ecosystem via the *Journey* class[6], which represents a trip taken by a user on public transport and is used as an observation's feature of interest. The *Journey* class references the public transport *Route*[7] being travelled, from which further information such as the location of roads the vehicle should travel along and details of stop points (e.g. their location[8]) can be determined. This provides contextual information about the observation that is required to determine RTPI (e.g. finding the locations of vehicles on a specific route) and to support quality assessment (e.g. calculating how far a reported location is from the expected route of travel).

---

[3] http://www.w3.org/2005/Incubator/ssn/ssnx/ssn
[4] http://www.dotrural.ac.uk/irp/uploads/ontologies/sensors.owl
[5] http//xmlns.com/foaf/spec/
[6] Defined by the User ontology - http://www.dotrural.ac.uk/irp/uploads/ontologies/user.owl
[7] Defined by the Transit ontology which describes public transport timetables.
[8] Defined by the NaPTAN dataset - http://data.gov.uk/dataset/naptan
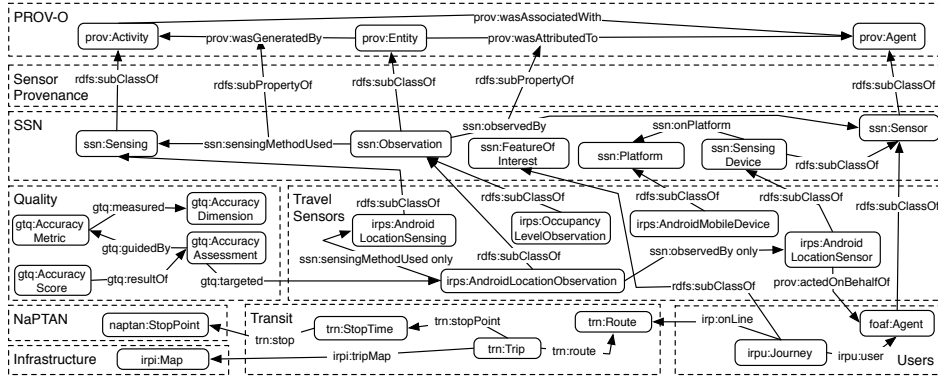
Fig. 1. Ontologies integrating sensor data with other data to support RTPI provision.

Further contextual information is provided by recording and inferring the provenance of observations. Provenance, a record of the agents and activities involved in producing, influencing, or delivering a piece of data can be used to form assessments about its quality, reliability or trustworthiness [6]. To capture observation provenance, we have defined a Sensor Provenance ontology[9], which aligns the SSN ontology with the W3C PROV-O ontology[10] through subclass and subproperty axioms. PROV-O is based around the concepts of *Entity*, a thing that *wasGeneratedBy* some *Activity* (something that occurs and acts upon or with entities), which in turn *wasAssociatedWith* an *Agent* (something that has some responsibility for an activity or entity). PROV-O is also used to capture the relationship between the sensors on a mobile device and the device's user through the *actedOnBehalfOf* property. This enables, for example, a record to be kept of any processing performed on observations within the ecosystem, and retrieval of the user associated with observations produced by a sensing device.

2.2 Services

The ecosystem also features a general-purpose web service architecture[11] suitable for applications that require management of sensor data expressed using the SSN ontology. Five services each provide RESTful APIs for creating, getting, updating, and deleting RDF descriptions of sensors, sensing methods, observations, sensor outputs, and observation values. Upon receiving a request, each service generates a SPARQL v1.1[12] update or query based on the parameter values included with the request[13]. This query is executed on a SPARQL endpoint; the service processes the endpoint's response and then sends a response to the client.

---

[9] http://www.dotrural.ac.uk/irp/uploads/ontologies/sensorprov.owl
[10] http://www.w3.org/ns/prov-o
[11] Available from https://github.com/dcorsar/sensor-service
[12] http://www.w3.org/TR/sparql11-query/
[13] Parameter values are type checked before use and an error thrown if checking fails.

This delegates storage to the technology backing the endpoint, allowing the use of, for example, a database and R2RML [3] if only storage and publication is required, or a triplestore and ontology reasoner if materialisation is required. This also allows the data to be published as Linked Data by, for example, using Pubby[14].

These services can also be extended for different use cases; for example, we have extended them to create a Location Observation service which manages real-time vehicle locations obtained from the *GetThere* app. This includes additional methods for creating and storing observations (and the associated sensor output and observation value) for a given geolocation and user's journey, and for retrieving the latest real-time locations for a particular bus route.

As discussed earlier, employing citizen sensing introduces the potential for low quality observations; therefore we have developed a sensor data quality service, underpinned by our data quality ontology[15] [1]. The service employs a SPIN reasoner guided by a number of SPARQL rules [4] to examine the metadata associated with the location observations provided by users. These rules describe a number of *quality metrics* that define how data should be evaluated against a number of *quality dimensions* [2]. Location observations are currently evaluated against four quality dimensions based on our experience of testing and deploying the system[16]. These are **accuracy** (accurate location observations have an associated error less than 25 metres), **availability** (considers any delay between the observation being observed by the device and received by the ecosystem due to mobile network lag); **timeliness** (timely observations were produced by the sensor less than one minute ago); and **relevance** (relevant observations are no farther than 500 metres from the expected route of travel, to support detecting potentially malicious/erroneous locations).

3   Deployment

The ecosystem and *GetThere* app[17] are currently deployed for nine bus routes in the Scottish Borders, UK (the First Group routes 62/62A, 72, 73, 95/95A/X95, 396, and 397). Fig. 2 (a), outlines the sensing components of the ecosystem for this deployment. This includes datasets for observations, user profiles, timetable, road infrastructure, and bus stop details (NaPTAN). Services reasoning with these datasets (accessed via SPARQL endpoints) provide RTPI functionalities for the *GetThere* app. These include the aforementioned Sensor, Location Observation, and Quality services, along with the User service which handles registering new users and managing user profiles, and the Timetable service which provides route and timetable information. Deployment in other areas simply requires creating the timetable and infrastructure datasets for that area.

---

[14] http://wifo5-03.informatik.uni-mannheim.de/pubby/

[15] http://sensornet.abdn.ac.uk/onts/Qual-O.ttl

[16] The ontology containing these rules is available at http://sensornet.abdn.ac.uk/onts/GetThereQ.ttl

[17] For a video showing the *GetThere* app see http://www.gettherebus.com/ssn2013
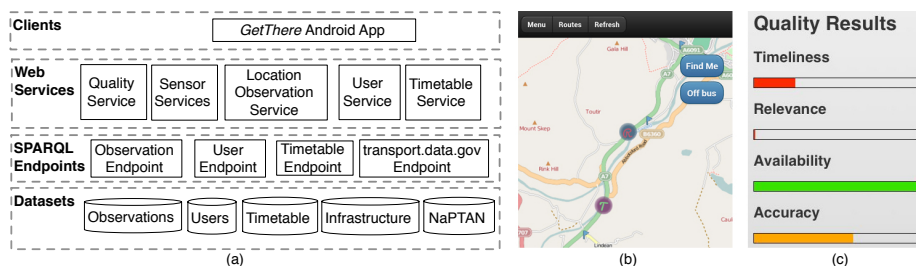
Fig. 2. The ecosytem's sensing components (a), and screenshots of the *GetThere* smartphone app showing vehicle locations (b), and results of quality assessment (c).

The app allows users to register, view the available bus routes, view both the timetabled and real-time bus locations (from other users) on a particular route (Fig. 2 (b)), and upload their location automatically, every minute during journeys. Users can tap the icon representing a real-time location to invoke the quality assessment service for that observation, the results of which are visualised using a colour-coded bar representing the quality score for each dimension. For example, in Fig. 2 (c), the green filled bar under "Availability" indicates a high quality score, while the nearly empty red bar under "Relevance" indicates a low score for that quality dimension. As of June 2013, there were 47 registered users, of which 17 have contributed 1008 location observations during 167 bus journeys.

4   Performance Evaluation

We have conducted a simulation in order to gain an insight into the performance of the sensor architecture when providing real-time vehicle locations with multiple users uploading locations. The simulation was based on buses travelling in two directions on six routes, with five passengers per bus (which we believe is a realistic maximum figure), each providing a location every minute. New buses were introduced every 30 minutes (reflecting the frequency of buses in the deployment area). A further 12 users were introduced to query for real-time vehicle locations every minute (one user per route per direction). The response times for all requests were recorded. The simulation ran for a period of 18 hours, reflecting a full day of bus operations in the Scottish Borders; it was executed on a single machine, with a 3.2 GHz Intel Core i3 processor with 4GB 1333MHz DDR3 memory with a Sesame MySQL triplestore used for data storage.

Lack of space prevents a detailed presentation and discussion of the results in terms of the response times for real-time locations[18]. However, in summary, for the 12960 requests made for real-time vehicle locations, 95% of the responses were generated in under 4.1 seconds; 99% of responses were generated in under 16.2 seconds. Given that the app uploads locations every minute, we believe these are acceptable response times for providing real-time vehicle locations.

---

[18] Full results are available at http://homepages.abdn.ac.uk/dcorsar/pages/ssn2013

5   Conclusions & Future Work

We have presented the sensor architecture developed to support citizen sensing within *GetThere*. This includes describing extensions of the SSN ontology to model citizen sensors. We found extending the SSN ontology simplified the design process for our sensor model, as SSN defines the concepts (sensors, observations, etc.) we required and outlines how to extend them for our own model, The SSN ontology also provides a method for integrating sensor data with other datasets in the ecosystem, and could be aligned with PROV-O to facilitate recording provenance of observations Two design issues faced were: how to handle sensor capabilities, for example, should all location sensors on Android smartphones use the same or different individuals to represent their location sensing capability?; and ensuring observations created by a sensor on a smartphone link to the phone's user, which we opted to capture in the provenance record. Aligning SSN with PROV-O enables provenance to be automatically inferred for each observation, providing data that can be useful for services such as quality assessment. Using ontologies is also beneficial when defining quality metrics, as they allow separate metrics assessing the same dimension for different types of observation. For example, they allow separate metrics for assessing timeliness of location and occupancy level observations, which will require different criteria.

As part of future work, we plan to increase the types of observations that the *GetThere* app acquires from users to capture other aspects of their journeys (e.g. presence of Wi-Fi, vehicle temperature), and extend our sensor architecture to accommodate such observations. We are also developing a model to determine the trustworthiness and reputation of citizen sensors, which incorporates aspects of the ecosystem, such as the quality evaluation of their previous contributions.

References

1. Baillie, C., Edwards, P. Pignotti, E., Corsar, D.: Short paper: Assessing the quality of semantic sensor data. In: Proc. of The 6th International Workshop on Semantic Sensor Networks (October 2013), to appear
2. Bizer, C., Cygniak, R.: Quality-driven information filtering using the wiqa policy framework. Journal of Web Semantics 7, 1–10 (2009)
3. Das, S., Sundara, S., Cyganiak, R.: R2rml: Rdb to rdf mapping language. W3C Recommendation (September 2012)
4. Furber, C., Hepp, M.: Swiqa - a semantic web information quality assessment framework. In: 19th European Conference on Information Systems. pp. 922–933 (2011)
5. Lefort, L., Henson, C., Taylor, K.: Semantic sensor network xg final report. W3C Incubator Group Report (June 2011)
6. Moreau, L., Missier, P.: Prov-dm: The prov data model. W3C Recommendation (April 2012), http://www.w3.org/TR/prov-dm/
7. Sheth, A.: Citizen sensing, social signals, and enriching human experience. Internet Computing, IEEE 13(4), 87–92 (2009)