# Using Sensors to Bridge the Gap between Real Places and their Web-Based Representations

Myriam Leggieri*, Christian von der Weth†, and John G. Breslin*
*Insight Centre for Data Analytics, National University of Ireland Galway
Email: firstname.lastname@insight-centre.org
†SeSaMe Centre, Interactive & Digital Media Institute, National University of Singapore
Email: vonderweth@nus.edu.sg

***Abstract*—With the proliferation of Smart Cities, more and more live data sources such as webcam feeds and physical sensor information are publicly accessible over the Web. However, these sources are typically decoupled from normal websites, and are therefore not within the scope of *traditional* online search using Web search engines. In this paper, we focus on websites that refer to physical locations (e.g., restaurants, hotel, shops) for which live sensor data and information might be available. We propose G-SENSING, our platform for the seamless integration of live data into the normal browsing experience of online users. In a nutshell, we provide a browser add-on that injects sensor information into Google result pages for each result that refers to a physical place. Our backend infrastructure consists of a data repository connecting websites to physical locations, as well as a data source for sensor information based on Linked Data principles. In our evaluation, we first show that websites referring to places are a very common phenomenon, thus motivating the potential benefits of G-SENSING. Furthermore, we show that our system only adds a small overhead to normal bandwidth requirements when browsing the Web.**

## I. INTRODUCTION

Finding relevant and reliable information on the Web is a non-trivial task, yet the tools to help us do this have reached a high level of maturity. For example, Google not only indexes all Web information, but also provides good results from the information that is contained within online fora, social networks, etc. We argue, however, that there are search requests that cannot be satisfied by traditional browsing or by existing technologies. The focus of this paper is on users searching for information about real-world locations such as hotels, restaurants, bars, shops, as well as offices, governmental or private institutions. A user's location-related information needs often refer to data which is only valid for very specific and short time frames. As a result, they are typically not maintained on a Web page, let alone indexed by a search engine.

EXAMPLE 1: Consider a user who is searching for an acupuncture clinic so as to book a visit. Using Google and Google Maps for example, the user can find suitable candidates in a given area as well as links to their official websites. Such websites usually provide information about the services that they offer, contact details, and maybe some pictures. The user can also read reviews about the clinic on recommender sites such as WHATCLINIC.COM. However, the user might also be interested in live data which could help them with their choice of a clinic, e.g., data about the number of free parking spaces close to the clinic or the number of customers currently in the waiting room. □

In recent years, there has been a tremendous uptake in the deployment of sensors within the concept of *Smart Cities*, including live camera feeds, weather sensors, etc. Data streams from such sensors are often publicly accessible, but are usually decoupled from other related Web resources (in our case, websites referring to real-world places near a specified geo-location). From a website provider's perspective, integrating live data is costly. Firstly, the data is distributed across different sources, i.e., live data providers. This includes that the number of sources may change over time. Secondly, the typically static websites have to be modified to accommodate the nature of data streams.

In this paper, we present G-SENSING, our approach towards the seamless integration of live sensor data into a user's normal browsing experience. To accomplish this, as a frontend application, G-SENSING features a browser add-on. The add-on injects live sensor information into Google search result pages – that is, if a search result refers to a real-world location, the add-on requests relevant sensor data from the backend and displays it directly next to the corresponding result. Regarding our backend as a source of live data, we aim for an open and flexible infrastructure that will allow us to easily change between different publicly available data sources. As a set of minimum requirements, a G-SENSING-enabled data source must (a) expose a SPARQL endpoint to query the data, (b) provide semantically annotated sensor data, and (c) register on DATAHUB[1] with its sensor tag metadata. For our current system we have implemented LD4S (Linked Data for Sensors), which fulfils all three requirements. LD4S can be queried using SPARQL or via a RESTful API using the JSON data format. For our evaluation, we randomly generated sensor readings and metadata for 30 sensors. They were divided into groups of 10 sensors, each deployed within 1 km from three of the Galway acupuncture clinics listed in the first Google search result page. We achieved optimal performances at a minimum cost of bandwidth increase (only 30 KB).

The paper outline is: Section II describes G-SENSING, highlighting the core aspects of the browser add-on frontend application and of our LD4S backend infrastructure. Section III presents the results of our evaluation to illustrate the practicality of our system. Section IV provides a discussion and outlines a roadmap for our ongoing and future work. Section V reviews related work to put our approach into context. Section VI concludes this paper.

[1]http://datahub.io/

## II. G-SENSING

The most common way to discover sensor data sources (as well as the live data itself) is through the means of dedicated platforms. For example, users can browse DATAHUB's list of publicly accessible sensors. However, this decouples live data from the more traditional Web resources like websites, and therefore puts them somewhat out of the reach of normal users who are browsing and searching the Web. We argue, however, that users' search requests often refer to information about a physical location. In these cases, users could potentially benefit from the information stemming from live data that is typically not shown on websites relating to a particular location. Regarding Example 1, the websites of acupuncture clinics are unlikely to provide information about the number of currently available parking spots nearby. G-SENSING supports a seamless integration of live data into the user's normal browsing experience. G-SENSING features an add-on that injects relevant live sensor information into GOOGLE search result pages (at present, with more to be added later). The sources of sensor data are registered datasets on DATAHUB. This approach allows the use of different and/or multiple datasets as data sources for the live data to be requested by our add-on. With LD4S, we have implemented our dataset of semantically-annotated sensor data, and registered it on DATAHUB.

### A. *Frontend Application – Browser Add-On*

Once a user has installed our add-on, it performs the following two event-driven tasks: discovery of (new) data sources after starting the browser, and requesting / injecting live data after loading a new Web page.

*(1) Discovery of data sources.* We aim for an open and flexible infrastructure that allows us to easily change between different publicly available data sources. We only assume that such data sources expose a SPARQL endpoint to query the data, provide semantically-annotated sensor data, and are registered on DATAHUB with sensor-tag metadata. DATAHUB is a data management platform from the Open Knowledge Foundation that exposes a JSON API to access metadata from the registered datasets. Anyone can register a dataset for free and specify its copyright, its access endpoints, and the type of data contained therein. As a result, the set of available data sources can change over time. To reflect this, we request information about available data sources registered on DATAHUB each time a user opens the browser. For each dataset retrieved, we verify whether it is publicly open and whether it exposes one or more SPARQL endpoints. If so, we forward a SPARQL query to get the sensor details for the coordinates of interest from each SPARQL endpoint.

*(2) Content request and injection.* The add-on listens to each page load event. If a new page has been loaded into the browser, Algorithm 1 is executed. We first test if the new page is a GOOGLE results page (Line 2). For the time being, we limit ourselves to GOOGLE result pages, but intend to extend our idea to any Web pages that refer to geographic locations (see Section IV for details). We then extract all individual search results from the page (Line 3). For each result, we first extract the URL of the link (Line 5) and request any place information that is associated with that URL by sending a search request to our backend (Line 6). If the URL is linked to a physical location, we again send a request to the backend to fetch all live data from each of the discovered data sources for that location (Line 8). Finally, we update the result dictionary (Line 9) and return the dictionary (Line 13).

**Algorithm 1:** *handlePageLoad(url)*

```
1  liveDataMap ← {};
2  if isGoogleSearchResult(url) = True then
3      searchResults ← extractSearchResults(url);
4      foreach result ∈ searchResult do
5          resultUrl ← result.url;
6          place ← requestPlaceData(resultUrl);
7          if place != ⊥ then
8              liveData ← requestLiveData(place.coords);
9              liveDataMap[resultUrl] ← liveData;
10         end
11     end
12 end
13 return liveDataMap;
```

Having received all of the live data, the add-on displays the data next to the corresponding search result by injecting the content into the HTML page. Figure 1 is an example screenshot showing the additional live data provided for acupuncture clinics in Galway, Ireland.

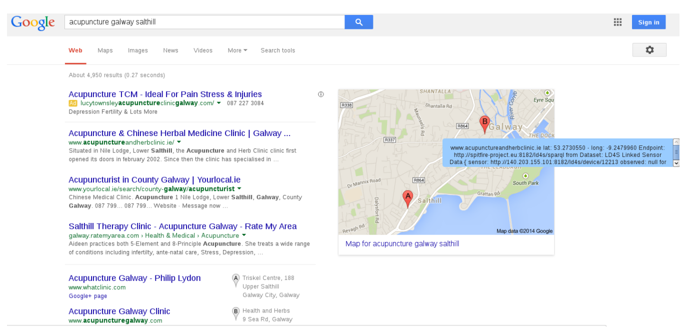



Fig. 1. G-Sensing overlays sensor data and metadata referring to the virtual representations of real places among Google search results in the browser.

### B. *Backend – LD4S and Semantics*

Our backend (acting as the source of live data) feeds the browser extension with sensor readings and sensor metadata queried across distinct distributed datasets. We have developed an open and flexible infrastructure that can alternate between different publicly available data sources and which is based on specific design choices. Such choices include a focus on specification-based sensor representation, storage and retrieval, using the Resource Description Framework (RDF), and following the principles of ontology reuse and Linked Data. RDF is a standard model[2] for data interchange on the Web designed to facilitate data merging, especially in cases where the underlying schemas differ. This allows us to be schema-independent when forwarding our query across distributed datasets.

[2]http://www.w3.org/RDF/

RDF extends the linking structure of the Web to use URIs to name the relationship between things, as well as the two ends of the link. This is usually referred to as a *triple*. For example, a triple in our model would be `<SensorTemporalProperty, hasLocation, Place>`, i.e., `<subject, predicate, object>` as shown in Listing 1. Such a linking structure creates a directed, labelled graph, where resources are nodes and edges are named links between two resources. This simple model allows structured and semi-structured data to be mixed, exposed and shared across different applications. Several syntaxes exist for serialising RDF graphs. In this paper we will use Turtle.[3]

In order to disambiguate the terms *SensorTemporalProperty*, *hasLocation* and *Place* used as subject, predicate and object in the example, we address each term with the URI of their representation. This also allows us to place these terms in relation to other terms (concepts) of the ontologies, thus modelling the concept that each term represents. For example, assuming we use a SPITFIRE ontology that has the namespace `spt`, the DOLCE ontology that has the namespace `dul`, and a Semantic Sensor Network-related ontology that has the namespace `ssn`, our previous triple becomes `<spt:SensorTemporalProperty, dul:hasLocation, ssn:Place>`. Each URI of the form `<namespace>:<conceptID>` will return the description of a concept. However, this also allows us to disambiguate a concept's meaning if a community gradually converges towards an agreement on which ontologies to use, i.e., adhering to the ontology reuse principle. For this reason, we have only used already existing, widely used and design-approved ontologies, including the W3C Semantic Sensor Network (SSN) ontology [3], the SPITFIRE [2] ontology,[4] DOLCE Upper-Level Concepts,[5] and the LD4S namespace to access the data stored in its own triple store. Also, to support future reasoning over concepts, we chose ontologies in languages that fully support logic inference, e.g., OWL.[6]

Linked Data principles [1] state that the above mentioned URIs should also be HTTP URLs which, when dereferenced, should return a proper RDF description of the concept they represent with RDF links to external resources. RDF links are triples where the object is located in an external dataset. We believe that such principles will become even more relevant with the development of the Internet of Things, since mobility and pervasiveness of the available data will need to be reflected by cross-linked public datasets.

The Simple Protocol and RDF Query Language (SPARQL)[7] defines a standard query language and data access protocol for use with the RDF data model. SPARQL allows us to gather information from RDF graphs in the form of subgraphs, URIs, blank nodes or literals. It also allows us to construct new RDF graphs based on the information contained in the queried graphs. We rely on SPARQL because it is specifically designed for running cross-dataset queries which are becoming increasingly relevant with the advent of the Internet of Things.

[3]http://www.w3.org/TeamSubmission/turtle/

[4]http://spitfire-project.eu/ontology/ns/

[5]http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite

[6]http://www.w3.org/TR/owl-ref/

[7]http://www.w3.org/TR/sparql11-query/

We used an RDF-based representation [4] so that sensor readings and metadata are described by `<subject, predicate, object>`-type triples in a graph.

The sensor location and *feature of interest* are part of a separate graph of information that may change over time, e.g., in case of a sensor being attached to a different feature and/or being mobile. Listing 1 shows the RDF serialisation of such a graph, where the start and end time validity for this set of properties is specified (by the predicates `spt:tStart` and `spt:tEnd`).

Listing 1. Example showing a representation of some sensor data.

```
ld4s:sensorID12
        a spt:SensorTemporalProperty;
        ssn:featureOfInterest ld4s:parkingLot;
        spt:temporalOf ld4s:2_99;
        spt:tStart "2013-03-31T01:00:00Z"
                ^^xsd:dateTime;
        spt:tEnd "2015-05-31T01:00:00Z"
                ^^xsd:dateTime;
        dul:hasLocation ld4s:herbClinic.
ld4s:herbClinic
        a ssn:Place.
```

As mentioned, each entity in a triple is represented by a HTTP URI that can be browsed online to retrieve further RDF information on the entity itself. The intrinsic nature of RDF facilitates the export of knowledge across different systems. This, combined with linking whenever possible of each entity to external structured definitions, and with ontology reuse, allows us to avoid ambiguities.

However, annotating sensor data and metadata semantically, while referring to external definitions, and storing them in triple stores is expensive and constitutes a learning barrier. For this reason, we have implemented and used LD4SENSORS (LD4S) [11] as a JSON Web service which exposes a RESTful API to automate the annotation and linking process. LD4S uses the JENA library and the JENA triple DB. Semantically linked sensor annotations as described above can be created, stored and edited through the LD4S API[8] or GUI,[9] where users can submit raw information via common HTML forms and they can be searched via the LD4S SPARQL endpoint.

The semantic annotation can be enriched by searching for relevant external links among *Linked Open Dataset (LOD)* resources – indexed by Sindice[10] – after creating a query according to criteria specified by the user, such as domain and/or context (i.e., time, space, thing). According to the specific criteria matched by the LOD resources, a different type of RDF predicate (e.g., `spt:sameDomain`, `spt:sameThing`, `spt:sameTime`, `spt:sameSpace`) is used in the named RDF link to the external resource of interest. In fact, we believe that for links having external data to be useful, they must change dynamically according to the specific use case.

In particular, we have used the online available instance of LD4S[11] to store – and later search for – sensor readings and

[8]http://spitfire-project.eu/incontextsensing/apiDocumentation.php

[9]http://jbossas7-huangyuan.rhcloud.com/spitfire-gui/

[10]http://sindice.com/

[11]http://spitfire-project.eu:8182/ld4s/ping

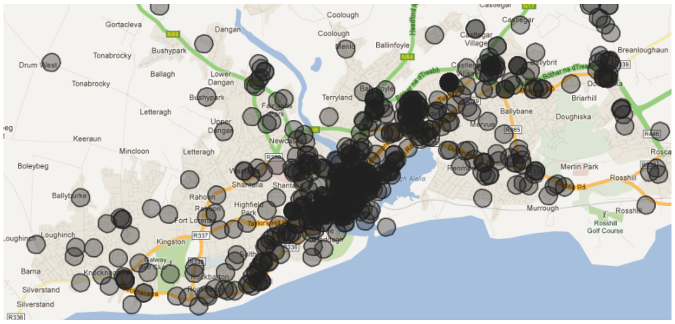


Fig. 2. Qualitative illustration of the coverage and distribution of places associated with a website across the city of Galway.

metadata that we previously generated randomly for our evaluation experiments. The LD4S SPARQL endpoint is published on DataHub[12] to facilitate its discovery by third parties.

## III. Evaluation

We considered three acupuncture clinics based in Galway, Ireland. We randomly chose them from the first page of GOOGLE's search results for the query "*acupuncture galway salthill*",[13] *Acupuncture & Chinese Herbal Medicine Clinic*,[14] *Acupuncture Galway Clinic*[15] and *Evidence-Based Therapy Centre*.[16] We randomly generated data and metadata for sensors deployed at different latitude and longitude coordinates within 1 km (approximately 0.009 degrees) of the coordinates of our three acupuncture clinics.

We simulated the deployment of groups of 10 sensors near each of the three acupuncture clinics of interest. A script implemented in JavaScript generated readings and metadata for a total of 30 sensors. It then forwarded PUT requests to LD4S for semantically linking, annotating and storing of such data (as described in Section II-B).

We crawled GOOGLE PLACES to collect all places within the city of Galway. Our current dataset contains 3,692 locations, 1,455 (39.4%) of which are associated with a website, i.e., they have a URL.

### A. Analysis of Data Repository

We first looked at the coverage, i.e., how much of the area defined by the virtual locations, i.e., places associated with a website, together with their vicinity radiuses overlaps with the city of Galway. Figure 2 illustrates the coverage with a vicinity radius of $r = 150$m. To get more quantitative results, Figure 3(a) shows the percentage coverage as we vary the vicinity radius. Naturally, the coverage increases for larger $r$, resulting in up to 72% coverage for $r = 250$m. Regarding the distribution of virtual locations (i.e., real places that have a corresponding official website as their virtual representation), we divided the areas of Galway into squares with different side lengths $l$ and counted the number of virtual locations within each square. Figure 3(b) shows the percentage of non-empty squares, which naturally increases for larger-sized squares. Empty squares typically cover city parks or purely residential areas. Figure 3(c) shows the distribution of non-empty squares for $l = 100$m. Not unexpectedly, the number of virtual locations per square and their respective frequency shows a power-law relationship: while most squares only contain a small set of locations, a few squares will contain a very large number of locations (e.g., city centres, business parks). Given these results, we argue that there are many websites that refer to physical locations, emphasizing the added value of our approach for integrating live data into such websites.

### B. Performance

For the performance experiments, we installed the browser add-on on commodity hardware equipped with an Intel Core 2 Duo processor and 305 GB of disk space. To be of practical use, we required that G-SENSING would not significantly add to a user's bandwidth consumption. We used the LD4S service instance running on an external server in order to support and test a modular and distributed architecture.

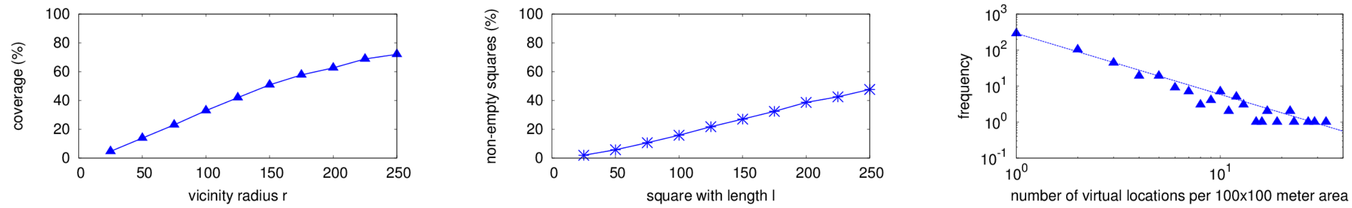
On average, GOOGLE result pages were around 145 KB in size. When G-SENSING is enabled, the bandwidth consumption is around 175 KB, an increase of just 30 KB ($\sim$20%). Part of the bandwidth consumption can also be attributed to the use of HTTPS rather than HTTP. 20% is a modest but reasonable additional overhead, particularly, since the overhead is comprised of information that is useful to the user. In our ongoing work, we aim to filter the requested live data by tailoring it to the information needs of an individual user (cf. Section IV).

Apart from the overhead in terms of required bandwidth, we also measured the average time to request and receive the live data. For this, we first forwarded a query to the DATAHUB API to retrieve all of the available sensor datasets. This represents the data sources discovery task during a browser start-up. The execution time was 3 milliseconds and returned 20 datasets, out of which three had an open license and also exposed a SPARQL endpoint. Among these, LD4S was actually the only accessible source, so the average response time for SPARQL queries we run is actually referring only to queries running on LD4S. Listing 2 shows an extract of one of these queries (the namespace prefixes have been omitted for clarity) which is used to retrieve sensors within a specific time range and near certain location coordinates. LD4S provided a response to the query in Listing 2 within 246 milliseconds.

Listing 2. SPARQL query targeting sensor data in a time and location range.

```
SELECT ?sens ?starttime ?endtime ?obs
?foi ?value ?location
{ ?sens spt:obs ?obs.
        ?ov spt:outOf ?sens;
                spt:value ?value;
                spt:tStart ?starttime;
                spt:tEnd ?endtime.
        ?tsp spt:temporalOf ?sens;
                ssn:featureOfInterest ?foi;
                wgs:lat ?latitude;
                wgs:long ?longitude.
FILTER (
xsd:dateTime(?starttime) >=
                '2014-11-30T02:00:00Z'
                ^^xsd:dateTime
                [..]
```

[12] http://datahub.io/dataset/ld4s-linked-sensor-data
[13] https://www.google.ie/#q=acupuncture+galway+salthill
[14] http://www.acupunctureandherbclinic.ie
[15] http://www.acupuncturegalway.com
[16] http://www.ebtc.ie/acupuncture




(a) coverage for different vicinity radiuses $r$ (b) ratio of non-empty squares (c) distribution of non-empty squares

Fig. 3. Coverage and distribution analysis regarding virtual locations across the city of Galway, Ireland.

```
        && xsd:double(?latitude) <=
                 '53.2692120'^^xsd:double)
                 [..].
}
```

## IV. Discussion & Roadmap

Our experimental results demonstrate that: (a) websites about or referring to real-world locations are a common phenomenon in urban areas; (b) the performance of G-Sensing does not impede on a user's browsing experience in terms of the average response time and additional bandwidth overhead. At present, we also note that there is limited availability of sensor datasets that are both open and public, while some of the SPARQL endpoints for such open and public datasets were inaccessible. However, we expect a larger number of available as well as accessible datasets in the future, since providing public data via a standardized access mechanism is still quite a recent trend.

*Beyond search result pages.* For the time being, we are showing live data alongside Google search results. Our add-on-based approach can, in practice, allow us to inject sensor information into any website. For example, we could display information about the parking situation around a restaurant on the restaurant's official website. With this, users can benefit from live data during normal browsing sessions, i.e., while navigating from website to website without relying on explicit search requests.

*Extended linkage.* So far, we have linked Web content with geolocations using public data crawled from Google Places, where many locations are associated with a URL (typically the websites of hotels, restaurants, shops, etc.). In the next step, we aim to extend these connections by injecting sensor information into other Web pages that also refer to such venues. For example, we are currently extending our data repository by crawling user reviews from TripAdvisor. Linking the review URLs to geolocations using our existing Google Places data will enable us to show relevant live data to users who are reading reviews on TripAdvisor. In the long run, we will explore which types of connections are meaningful in a given application context and how such connections can be established. For example, we envision displaying the latest webcam feeds showing a location that is mentioned in a news article. Creating such links in an automatic and reliable manner is a challenging task.

*User-centric live data representation.* The G-Sensing output relevancy for an end user depends on the user's current interests and on the type (and low level location) of sensor data displayed, e.g., sensor data about occupancy in the clinic waiting room or in the surrounding parking area, rather than sensor data about the temperature of the fridge in the clinic's kitchen. Future releases of our system will include a recommender system that decides whether to display or not the retrieved sensor data according to a prediction of their current relevancy for the user.

## V. Related Work

*Sensors on the Web.* Several efforts have tried to make sensors accessible from the Web, enabling the so-called *Sensor Web*. This includes standards defined by the Open Geospatial Consortium's (OGC) Sensor Web Enablement (SWE) project [16]. In addition to SensorML [7], they defined the Sensor Observation Service framework (SOS) [15] which enables publishing and accessing of sensor data using XML-based protocols and APIs.[17] The usage of XML means that ad-hoc mappings of different schemas are needed to integrate data. Thus, SOS, like all the other OGC standards, neither enables semantic interoperability nor supports reasoning.

*Semantic Sensor Web* [19] investigated a semantic version of the OGC vision. A semantic extension to OGC's SOS was developed, SemSOS [6] (Semantic Sensor Observation Service). In parallel, the W3C Semantic Sensor Network (SSN) Incubator Group, realized an ontology [3], based on SensorML, adopted by several research projects and which also LD4S relies on. This ontology aims to provide cross-domain concepts for sensors, and was inspired by several domain-specific ontologies that existed previously. It supports annotations of sensor-related features, e.g., deployment, observations and measurement capabilities. Thus, it enables the automation of further tasks like fine-grained discovery (e.g., a search for sensors which are observing wind direction with a specific accuracy level and maintenance scheduling).

A further step in using simple semantic annotations is to follow *Linked Data* principles as in our LD4S. To the best of our knowledge, this has not yet been done for dynamic sensor datasets, mainly due to the lack of dereferencable URIs which point to the sensor data source, i.e., the sensor that generated the data. The usefulness of the external links found for sensor data has been investigated by [12], where [10] stressed the importance of contextual information for sensor data. Finally, a relevant attempt to implement an RDF triple store and brokerage system on the sensor nodes themselves has been successfully accomplished by [5].

[17] http://www.opengeospatial.org/standards/sos

*Map interface for sensors.* In line with efforts to make sensors accessible from the Web, several projects have focused on overcoming sensor network heterogeneity. They usually create an abstraction layer and visualize sensors on world maps. Microsoft's SensorMap [14] mashes up sensor data from a worldwide heterogeneous sensor network (SenseWeb) on a map interface and provides interactive tools to selectively query sensors and visualize data, along with authenticated access to manage sensors. This was followed by similar efforts since then, with GraphOfThings [17] being the most recent one. While we also rely on latitude and longitude coordinates to locate a sensor device, we provide an alternative perspective. Rather than representing the device location on a map we represent it in relation to its already existing virtual representation in the form of websites. Also, by injecting our system into Google search results, we actually make sensors more accessible to most Web users.

*Virtual versus Real Spaces.* The difference between real and virtual places has been analysed in several research areas including philosophy [9], e-learning [20], augmented reality [8], collaborative software development [13], social networking and communities [18], etc. More recently, von der Weth et al. [21] proposed a scientific foundation for the problem of mapping the physical presence of people in real places to the online presences of users in virtual places (on websites). Similar to this work, they propose enriching virtual places with real information. However, they rely on users to act as sensors, providing information about those real places in a chat-type browser extension. This research is the closest so far to our approach. We build on top of it, investigating the use of sensor devices as opposed to human beings for providing virtual content enriched with live data.

## VI. Conclusions

In this paper, we have proposed an approach to bridge the gap between the semi-static content offered by websites and the short-lived information offered by sensors. While the majority of sensor-related research goes in the direction of dedicated platforms through which users can explore available live data, we aim towards the seamless integration of sensor information into users' everyday browsing behaviors. As our main contribution, G-Sensing offers live data tailored to the information needs of online users of search engines. We facilitate this by providing a browser add-on which injects enriched sensor information into Google search results pages. At the same time, we keep our data source as open and flexible as possible, and approach the sensor data integration problem in innovative ways. Our backend infrastructure exposes a GUI, RESTful API and SPARQL endpoint to enable the annotation, storage and retrieval of semantic sensor data. Our evaluation showed both the potential benefits of G-Sensing and its applicabilty for large-scale settings.

In ongoing and future work, we focus on two main directions. Firstly, we aim to adapt the live data displayed according to a user's current context – that is, we would expose, for example, different types of sensor data to a user commuting on a bus and a user sitting at home. This requires appropriate context-modeling techniques as well as extending both the add-on and backend infrastructure to support context-dependent content delivery and presentation. Secondly, we want to extend our browser extension to inject live data into any relevant Web page referencing a physical location (e.g., the official websites of hotels, restaurants, businesses) and not just into Google result pages.


## Acknowledgment

This work is funded by the European Union (Grant ICT-2011.1.3) and by Science Foundation Ireland (SFI/12/RC/2289), partly carried out at the SeSaMe Centre. It is supported by the Singapore NRF under its IRC@SG Funding Initiative and administered by the IDMPO.


## References


[1] T. Berners-Lee. Linked Data. Design issues for the world wide web, World Wide Web Consortium, 2006. http://www.w3.org/DesignIssues/LinkedData.html.

[2] D. Bimschas, H. Hasemann, M. Hauswirth, M. Karnstedt, O. Kleine, A. Kroeller, M. Leggieri, R. Mietz, M. Pagel, A. Passant, D. Pfisterer, K. Roemer, and C. Truong. SPITFIRE: Toward a Semantic Web of Things. *IEEE CommMag*, 2011.

[3] M. Compton, P. Barnaghi, L. Bermudez, R. G. Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. L. Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor. The SSN Ontology of the Semantic Sensor Networks Incubator Group. *JWS*, 2012.

[4] R. W. Group. Resource description framework (rdf), 2001.

[5] H. Hasemann, A. Kroeller, and M. Pagel. RDF provisioning for the Internet of Things. In *Internet of Things International Conference for Industry and Academia*. IEEE.

[6] C. A. Henson, J. K. Pschorr, A. P. Sheth, and K. Thirunarayan. SemSOS: Semantic sensor Observation Service. *CTS*, 2009.

[7] C. Jennings, Z. Shelby, and J. Arkko. Media Types for Sensor Markup Language (SENML), 2012.

[8] D. Kalkofen, E. Veas, S. Zollmann, M. Steinberger, and D. Schmalstieg. Adaptive Ghosted Views for Augmented Reality. In *IEEE International Symposium on Mixed and Augmented Reality* , 2013.

[9] D. Kolb. *Sprawling Place.* University of Georgia Press, 2008.

[10] M. Leggieri, A. Passant, and M. Hauswirth. A contextualised cognitive perspective for linked sensor data. In *SSN Workshop co-located with ISWC*, 2010.

[11] M. Leggieri, A. Passant, and M. Hauswirth. inContext-Sensing: LOD augmented sensor data. In *ISWC Posters and Demos*, 2011.

[12] M. Leggieri, A. Passant, and M. Hauswirth. inContext-Sensing: LOD augmented sensor data. In *ISWC Posters and Demos*, 2011.

[13] A. D. Lucia, R. Francese, I. Passero, and G. Tortora. SLMeeting: supporting collaborative work in Second Life. In *Working Conference on Advanced Visual Interfaces*, 2008.

[14] S. Nath, J. Liu, and F. Zhao. Sensormap for wide-area sensor webs. *IEEE Computer*, 2007.

[15] Open Geospatial Consortium. Sensor Observation Service, 2010.

[16] Open Geospatial Consortium. Sensor Web Enablement, 2010.

[17] D. L. Phuoc, H. N. M. Quoc, Q. H. Ngo, T. T. Nhat, and M. Hauswirth. Enabling live exploration on the graph of things. In *Semantic Web Challenge at the Internaltional Semantic Web conference*, 2014.

[18] D. M. Powazek. *Design for Community: The Art of Connecting Real People in Virtual Places.* New Riders, a Division of Pearson Technology Group, 2001.

[19] A. Sheth, C. Henson, and S. Sahoo. Semantic Sensor Web. *IEEE Internet Computing*, 2008.

[20] A. Sutcliffe and A. Alrayes. Investigating user experience in second life for collaborative learning. *International Journal of Human-Computer Studies*, 2012.

[21] C. von der Weth and M. Hauswirth. Finding information through integrated ad-hoc socializing in the virtual and physical world. In *IEEE/WIC/ACM International Conference on Web Intelligence*, 2013.