



## AN INTERNET OF THINGS PLATFORM FOR REAL-WORLD AND DIGITAL OBJECTS\*

SUPARNA DE, TAREK ELSALEH, PAYAM BARNAGHI, AND STEFAN MEISSNER †

**Abstract.** The vision of the Internet of Things (IoT) relies on the provisioning of real-world services, which are provided by smart objects that are directly related to the physical world. A structured, machine-processible approach to provision such real-world services is needed to make heterogeneous physical objects accessible on a large scale and to integrate them with the digital world. The incorporation of observation and measurement data obtained from the physical objects with the Web data, using information processing and knowledge engineering methods, enables the construction of "intelligent and interconnected things". The current research mostly focuses on the communication and networking aspects between the devices that are used for sensing and measurement of the real world objects. There is, however, relatively less effort concentrated on creating dynamic infrastructures to support integration of the data into the Web and provide unified access to such data on service and application levels. This paper presents a semantic modelling and linked data approach to create an information framework for IoT. The paper describes a platform to publish instances of the IoT related resources and entities and to link them to existing resources on the Web. The developed platform supports publication of extensible and interoperable descriptions in the form of linked data.

**Key words:** Internet of Things, Modelling, Linked-data, Semantics, Information model

**1. Introduction.** The Internet of Things (IoT) vision aims to enable the machine perception of the real world and seamless interactions with it. This vision is lent credence with the growing availability of smart objects that are directly related to the physical world and have the communication and computation capabilities to connect and interact with their surrounding environment. The data and/or services offered by such objects can provide information about the physical world and allow interaction with it. The services can either be information services exposing functionalities that can provide data on the surrounding physical world entities, or actuation services that can bring about a change in the state of the physical world objects. Initially, the IoT vision considered physical objects tagged with RFID transponders. However, this has grown to encompass sensor networks and distributed smart objects collaborating via local networks or through the Internet [9]. Thus, the resulting real-world data/services need to be defined and made available in a homogeneous way to allow integration of the data from the wide variety of heterogeneous sources and to support autonomous reasoning and decision making mechanisms. This points to the applicability of Semantic Web technologies that can provide a formal, structured and machine-processible platform to heterogeneous data sources, as well as providing context to the data and to the objects themselves. Initial efforts in this area have resulted in ontologies for sensor descriptions [6] as well as standardisation efforts towards semantic descriptions of sensor networks [17]. However, the semantic sensor descriptions need to be linked to the measurements and domain knowledge and then to the observed IoT entity in the domain.

Another key requirement for the IoT is a platform that facilitates the virtualisation of real world objects. Existing research works have focused on sensor (and actuator) middleware frameworks that offer sensor descriptions [1], sensor site data [10] and measurement data services [16] on the Web and/or at the application level. More generic approaches include those that provision flash interfaces for instantiating semantic profiles of connected objects [7]. To extend this to heterogeneous real world objects, the data from the physical world needs to be interlinked to domain knowledge and existing data sources on the Web. This can facilitate automated annotation and reasoning on the physical world data and lead to provisioning of intelligent applications.

Towards these twin aims, this paper describes the Sense2Web platform that allows publishing data descriptions for the components of the IoT domain in the form of Linked Data and makes this data available to other Web applications via SPARQL endpoints. The platform offers both a Human-to-Machine (H2M) and Machine-to-Machine (M2M) interface for publishing data. It incorporates a semantic description framework for the IoT components and provides a formal representation to the interactions. The platform allows generating of networked resources which collect data from the physical world as well as data and services on the Web. Interlinking data from the physical world and the Web supports the provision of networked knowledge [11].

---

\*This work is supported by the EU IoT-A project, IoT-A: Internet of Things Architecture (<http://www.iota.eu/public>) contract number: 257521. The second and third authors are also supported by the EU PPP FI-Ware project (<http://www.fi-ware.eu/>).

†Centre for Communication Systems Research, University of Surrey, Guildford, GU2 7XH, Surrey, UK ({S.De, P.Barnaghi, T.Elsaleh, S.Meissner}@surrey.ac.uk).

This also enables other related data and relevant information to be discovered and facilitates interconnection and integration of data from different communities and sources. The applicability of the platform is illustrated through a reference application scenario that implements a mash-up application using the generated linked data.

The rest of the paper is organised as follows: section 2 presents challenges and background information on linked data. The proposed information models are detailed in section 3. Section 4 presents the developed Sense2Web platform architecture and explains linking IoT concept descriptions to existing data sources on the Web. An example application that builds upon the platform functionalities is presented in section 5. Section 6 concludes the paper and discusses future work.

**2. Challenges and background information.** In this section we first discuss the issues related to annotation and publication of the IoT related data and making the data machine-interpretable to support automated scenarios. We then provide the principle of creation and publication of the linked data.

**2.1. Challenges.** In order to use the Linked data approach for publishing data from heterogeneous IoT concepts, we need to address the following challenges:

1. How to annotate "plain" data to make it semantically linked data: this refers to deciding what ontologies need to be leveraged to semantically describe the IoT domain. Moreover, the heterogeneity of possible IoT concepts requires using several ontologies together and this in turn, gives rise to the challenge of aligning and relating them to each other. This paper proposes a suite of ontologies to define an IoT information model. The ontologies build upon existing vocabularies and where appropriate, properties are included to allow linking the proposed ontologies to external ontologies where the given concept may be more completely described.
2. How to actually "link" the data together: the Linked Data principle is to "make data refer to each other so that it eventually forms a data network" [22]. However, currently, most of the data linkages are made manually or are very sparse [22]. The Human-to-Machine interface of the proposed platform offers automated hints for linking entered data to existing internal and external data repositories.
3. How to serve the published data in an application-programmable compliant way: currently, sensor network data applications apply Device Profile for Web Services (DPWS) [15]-based implementations [1], [14] to sensor gateways to offer sensor measurement data services. DPWS defines a limited set of WS-\* standards for resource limited devices. The majority of Linked data is currently served through SPARQL endpoints.

The Sense2Web platform offers the published IoT component descriptions as Linked data through SPARQL endpoints.

**2.2. Background Information - Linked data.** Publishing data on the Semantic Web with machine interpretable representations facilitates more structured and efficient access to the resources; however semantic descriptions without being linked to other existing data on the Web would be mostly processed locally and according to the domain descriptions (i.e. domain ontologies). Linking data to other resources enables obtaining more information related to a particular data item by exploring the links across different concepts and domains. The linked data concept was initially introduced by Tim Berners-Lee in 2006 [4]. Berners-Lee suggested four main principles to publish linked data: - using URIs as names for data, - providing HTTP access to those URIs, - providing useful information for URIs using the standards such as RDF and SPARQL, - Including links to other URIs. Publishing annotated and interlinked data is the underlying principal of creating linked Web resources that is referred to as the Web of Data [4]. In the Web of Data resources are connected via links that can be queried and interpreted using discovery and search agents [5]. Linked data enables navigation between different data sources by following the data connection links. This allows the linked data consumers to start with one data source and then browse through a vast number of resources interconnected by machine interpretable links (e.g. RDF links).

**3. IoT Information Models.** This section defines the main abstractions and concepts that underlie the IoT domain and describes the relationships between them. The main tenet of the IoT is extension of the Internet into the physical world, to involve interaction with a physical entity in the ambient environment. The entity

constitutes 'things' in the Internet of Things and could be a human, animal, car, store or logistic chain item, electronic appliance or a closed or open environment. The 'entity' is the main focus of interactions by humans and/or software agents. This interaction is made possible by a hardware component, a 'device', which either attaches to an entity or is part of the environment of an entity so it can monitor it. The device allows the entity to be part of the digital world by mediating the interactions. The actual software component that provides information on the entity or enables controlling of the device, is a 'resource'. As implementations of resources can be highly dependent on the underlying hardware of the device, a 'service' provides a well-defined and standardised interface, offering all necessary functionalities for interacting with entities and related processes. The services expose the functionality of a device by accessing its hosted resources. Other services may invoke such low-level services for providing higher-level functionalities, for instance executing an activity of a specified business process. The relations between services and entities are modeled as associations. These associations could be static, e.g. in case the device is embedded into the entity; they could also be dynamic, e.g., if a device from the environment is monitoring a mobile entity. These identified concepts of the IoT domain and the relations between them are depicted in Figure 3.1.

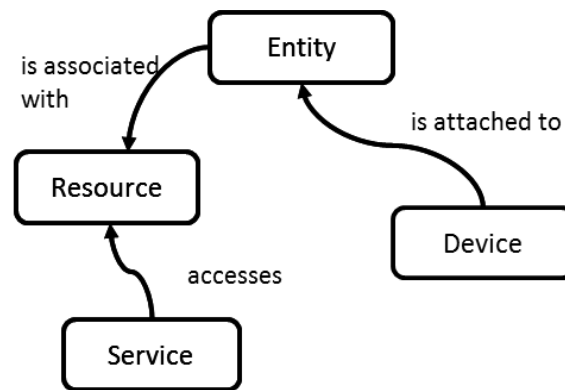


Fig. 3.1: IoT model: key concepts and interactions

Based on the identification above, of the main concepts in the IoT domain, this paper proposes a suite of ontologies that models entity, resources and IoT services. The ontologies are modelled in the Web Ontology Language - Description Logic (OWL-DL). Where appropriate, properties are included to allow linking the proposed ontologies to external ontologies; for example, the global location URI of an entity could link to the relevant location instance in the GeoNames ontology<sup>1</sup>, where the given location is more fully described. This enables reusability of ontologies and fosters modularity.

**3.1. Entity Model.** In addition to the required properties of an identifier and some attributes, an entity can have certain other aspects that need to be taken into account. For example, we may need to know about the location of an entity and the features that can be observed by a sensing mechanism to provide data about the observed feature. A diagram of the main attributes of the entity model is shown in Figure 3.2.

An entity has certain features, which include domain attributes, temporal features and location (Entity:hasA U(DomainAttribute, TemporalFeatures, Location)). The OWL union operation (U) on these features denotes that a particular entity instance can have either or all of these features. Moreover, an entity instance can have multiple values for the domain, temporal or location feature.

Domain attributes tie the entity instance to a particular domain and a semantic realisation of the model can link the entity instance to a domain ontology. The domain attribute is specified in terms of the attribute name (*hasAttributeName*), attribute type (*hasAttributeType*) and value. These attribute properties together

<sup>1</sup><http://www.geonames.org/ontology/documentation.html>

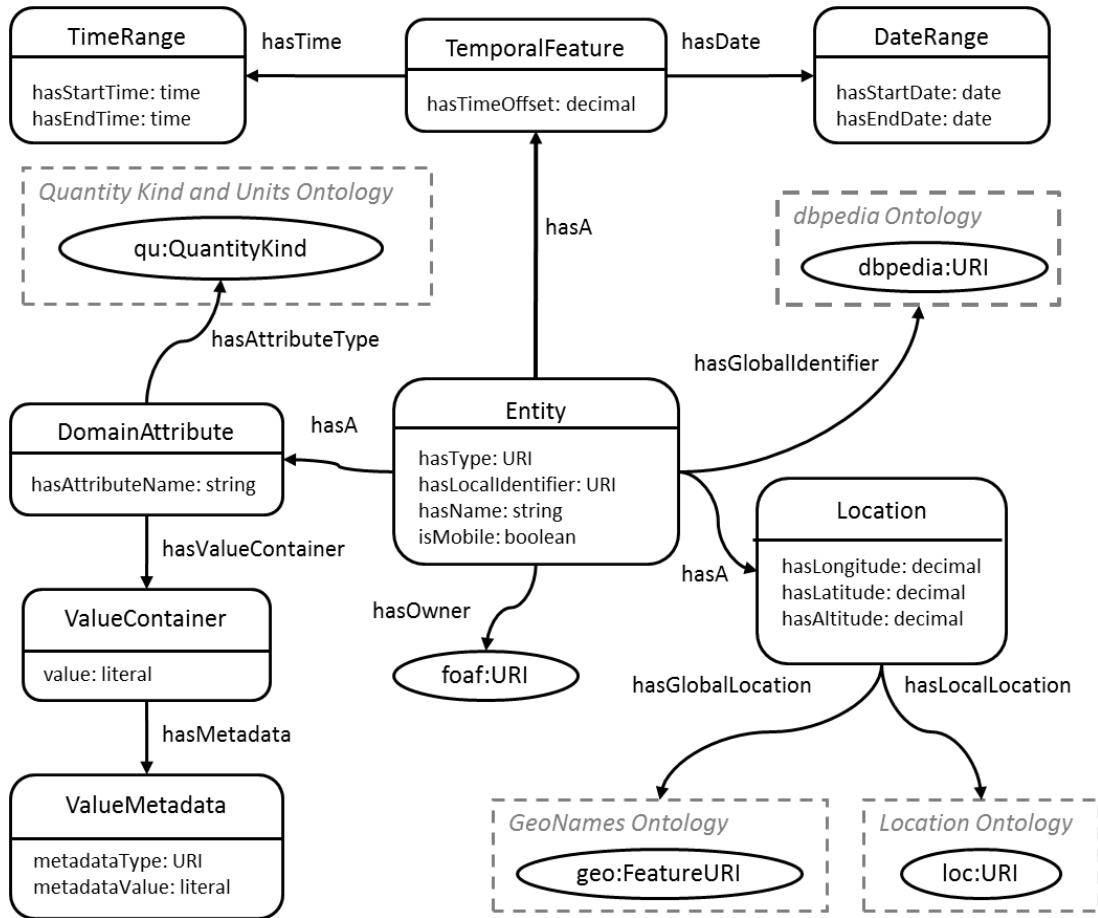


Fig. 3.2: The Entity model

describe an observable feature of the entity. Having the attribute name and type as distinct properties allows for two levels of data specification. The *DomainAttribute* instance's name property refers to the domain specific attribute of the virtual entity, e.g. Ambient Temperature. What a resource (e.g. sensor) will be able to measure will be the attribute type, i.e. Temperature, in this case. Thus, for two distinct domain attributes of the same virtual entity, e.g. Ambient Temperature and Body Temperature, what a resource would be concerned with, would be the attribute type, i.e. Temperature, which is the same for both domain attributes. Only the domain attribute property, which is intrinsic to the entity, puts what the resource senses, into context. The type of the "*DomainAttribute*" is further defined as "*QuantityKind*", taken from the "Library for Quantity Kind and Units" [13]. That library contains a list of physical phenomena, such as temperature or acceleration, which can be measured by sensors or influenced by actuators. The value itself has a literal 'value' and associated metadata information (*ValueMetadata*). The metadata could include information on, for instance, the units of measurement. It is specified in terms of the metadata value and metadata type.

The entity's lifetime is described by "*TemporalFeature*"'s further refined by "*hasTimeOffset*", "*TimeRange*" and "*DateRange*". The latter specifies intervals in a scale of days, months and years; "*TimeRange*" describes ranges in hours, minutes, seconds and fractions of seconds. A time offset to Coordinated Universal Time (UTC) in hours indicates the time zone the entity is currently located in. These capture the temporal properties of entities that may have temporal attributes, e.g. Meeting Rooms. The values of these properties can be compared with other dates by using date and time comparison built-ins (such as those available in Semantic Web Rule Language (SWRL) [23] to deduce facts about temporal aspects of the relevant entity.

Physical entities have a location at the time they exist in the real world. In this work, we focus on locations on the earth that can be described by geographic coordinates as well as symbolic locations, such as relative locations within a building. Barnaghi et al. [3] identify two location attributes for describing sensor data: the first attribute to refer to an instance of a local location ontology, which is a model of the current location offering high granularity and detailed information on the location in terms of the relative positioning of rooms, floors and buildings. The second location attribute was identified to be from a high-level concept available on the Web of data, such as DBpedia [2]. We adopt a similar approach in this paper and extend it to include specification of geographical coordinates for the entity location as well. Thus each 'Entity' can be given a "'Location'" that is modelled as a triple of float values describing longitude, latitude, and altitude as geographic position. The location concept also has properties that could link to local location (*hasLocation*) ontologies.

Additionally, an entity has datatype properties that specify the URI of an owner (*hasOwner*) where the URI could point to a foaf<sup>2</sup> profile, a literal name (*hasName*) and a Boolean property to denote if the entity could be mobile (*isMobile*). An important attribute of an entity is the entity type (*hasType*), which could be specified through the *rdf:type* property and hence, allow a Semantic Web engine to infer the type of the entity from its asserted properties, especially in cases where the entity could have multiple types. The local identifier (*hasLocalIdentifier*) property is the ID of the virtual entity. It could as well point to a local naming schema. The global identifier (*hasGlobalIdentifier*) property is a placeholder to associate the entity to the open Linked Data<sup>3</sup> platform; for instance, to a Dbpedia entry.

**3.2. Resource Model.** A resource is the core software component that represents an entity in the digital world. It allows the entity to be part of the digital world by mediating the interactions. Figure 3.3 details the resource description model.

The resource concept has *datatype* properties that specify its name (*hasName*), an ID (*hasResourceID*) and time offset (*hasTimeOffset*). The resource provider can specify certain keywords (or free text tags) describing the resource through the *hasTag* property. This is an optional property to allow the resource provider to provide a free text search for the resource instance. A resource also has a location property (*hasResourceLocation*) that links to the Location concept. This location could be the location of the device the resource runs on. The definition of the location concept is similar to that in the entity model. The resource type is denoted in terms of the type property (*hasType*) to the *ResourceType* concept. Resources can be instances of either of the following types: sensor, actuator, RFID tag, storage or processing resource. The different resource types are not disjoint, hence, resources can be an aggregation of several of these types. When the type is a sensor, the *hasType* property serves as a link to an instance of a sensor that conforms to an available sensor ontology (e.g. SSN sensor ontology). This allows linking the resource concept to external ontologies which already define in detail related concepts, without the need of repeating them in the resource model. Actuator resources modify the physical state of a physical entity. The RFID tag type is a specialised kind of sensing resource. A storage resource stores information obtained from other resources (such as sensors) and a processing resource includes methods to process the information aggregated from other resources (e.g. an aggregate of a temperature value coming from a number of sensors). As the access to a resource is provided by an IoT service, this link to the service is denoted by the "'isExposedThroughService'" object property that links the resource model to an IoT Service instance of the service model. The resource model also captures the link to the hardware 'device' on which it hosted (*isHostedOn*), which may be further described in a Device ontology.

**3.3. Service Model.** Resources are accessed by services which provide functionality to gather information about entities they are associated with or manipulate physical properties of their associated entities. The Service Model contains information needed for discovering and looking up the service as well as information on how to invoke the service. The service model is shown in Figure 3.4.

The actual technology used to invoke the service is modelled through the *hasServiceType* parameter, which could take a value such as 'REST' for a RESTful web service. The link to the resource to which the service

<sup>2</sup><http://www.foaf-project.org/>

<sup>3</sup><http://linkeddata.org>

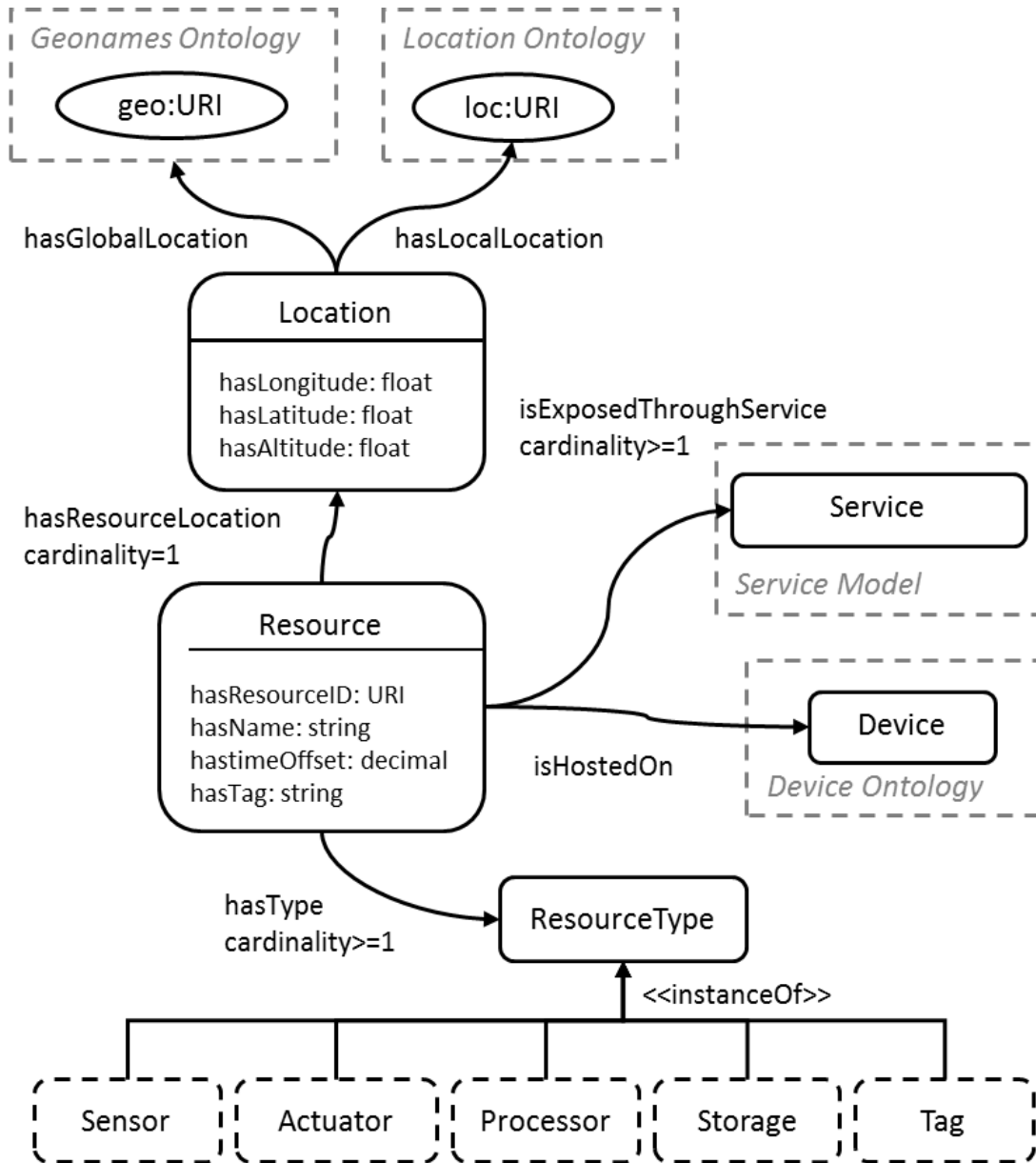


Fig. 3.3: The Resource model

provides access is specified through the exposes property that links back to an instance of the resource model.

One of the important aspects of a service is to allow for associations with virtual entities in the IoT domain. For this, the IoT-A proposed service model utilises the OWL-S [20] model as its upper ontology. The "*ServiceModel*" part of the OWL-S ontology is used to specify the input, output, preconditions and effects (IOPE) related parameters of the service model. Since the service model exposes the underlying resource's functionalities, the resource attribute that is exposed through an IoT service either as output data type (*hasOutput*) or as an input parameter (*hasInput*) is captured in the service specification. The feature can then be matched with the attribute type of the virtual entity with which it can be associated. For instance, a virtual entity can have an attribute that represents its "*indoorTemperature*". The generic type of this particular attribute is "*temperature*". Then, if there is a service exposed by a resource that measures temperature, specified as

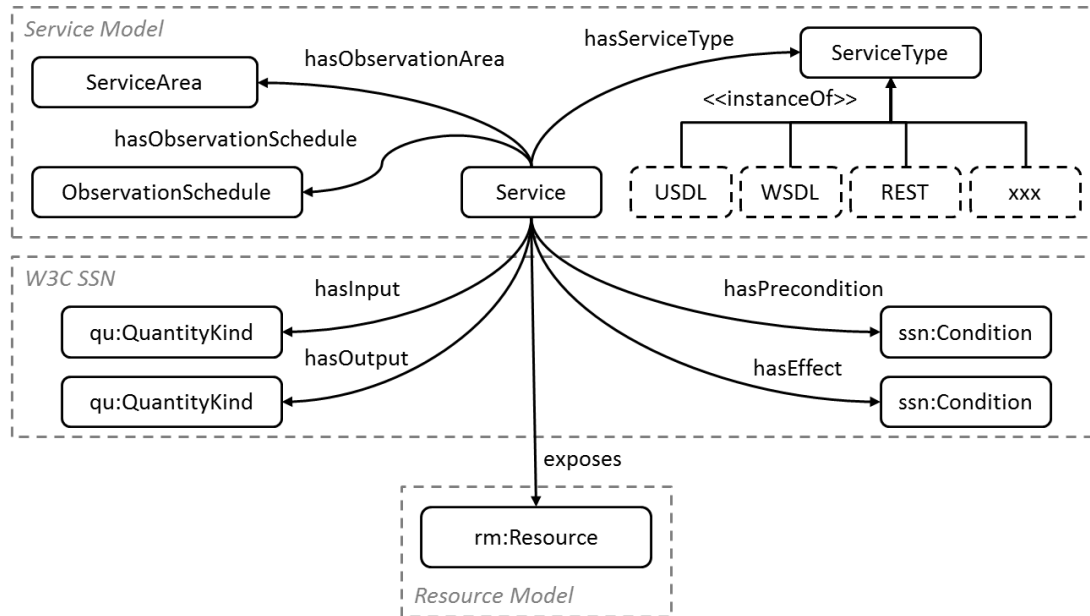


Fig. 3.4: The Service model

the service's *hasOutput* parameter, the corresponding service can be a candidate for possible association to the relevant virtual entity. The input and output parameters can be specified in terms of the generic instance quantities from the QU ontologies [18], such as "temperature" or "luminosity".

For actuating services, the state of the entity attribute being controlled is also important. This post-condition state is modelled through the *hasEffect* parameter in the service model. Similarly, any pre-conditions that need to be met before the service execution can be specified through the *hasPrecondition* parameter. The state object properties link to instances of the 'Condition' class of the SSN ontology [17], so that conditions that affect the resource's measurement or actuating capabilities can be specified. This is also an example where the SSN ontology concepts can be extended to include actuating conditions.

With location being an important criterion for service search and resolution, the area affected by the service is specified through the *hasServiceArea* property. For sensing services, this would be the observed area, while actuating services would specify the area of operation. The observation area of sensors can be different to their actual location. An example for that are camera resources observing areas at some distance to their position. The possibility of specifying time constraints on service availability is captured through the *hasServiceSchedule* property. This can allow IoT users to be informed about downtimes of resources, for instance, for energy efficiency reasons.

**4. Sense2Web Linked Data Platform.** The Sense2Web platform<sup>4</sup>, depicted in Figure 4.1, is a six-tiered framework for publishing linked IoT concept instances. The platform was developed in Java and deployed on the Apache Tomcat<sup>5</sup> web server.

The platform offers both a H2M as well as M2M interface for publishing IoT data and associating it to existing vocabularies on the Web. The core functions that are supported by the platform are essentially the CRUD (Create, Read, Update, and Delete) methods used for interacting with IoT entities and resources; in this case this translates to publishing, reading, updating and deleting the IoT concept descriptions. For all these

<sup>4</sup>An online version is available at: <http://ccsriottb3.ee.surrey.ac.uk:8080/IOTA/>

<sup>5</sup><http://www.tomcat.apache.org>

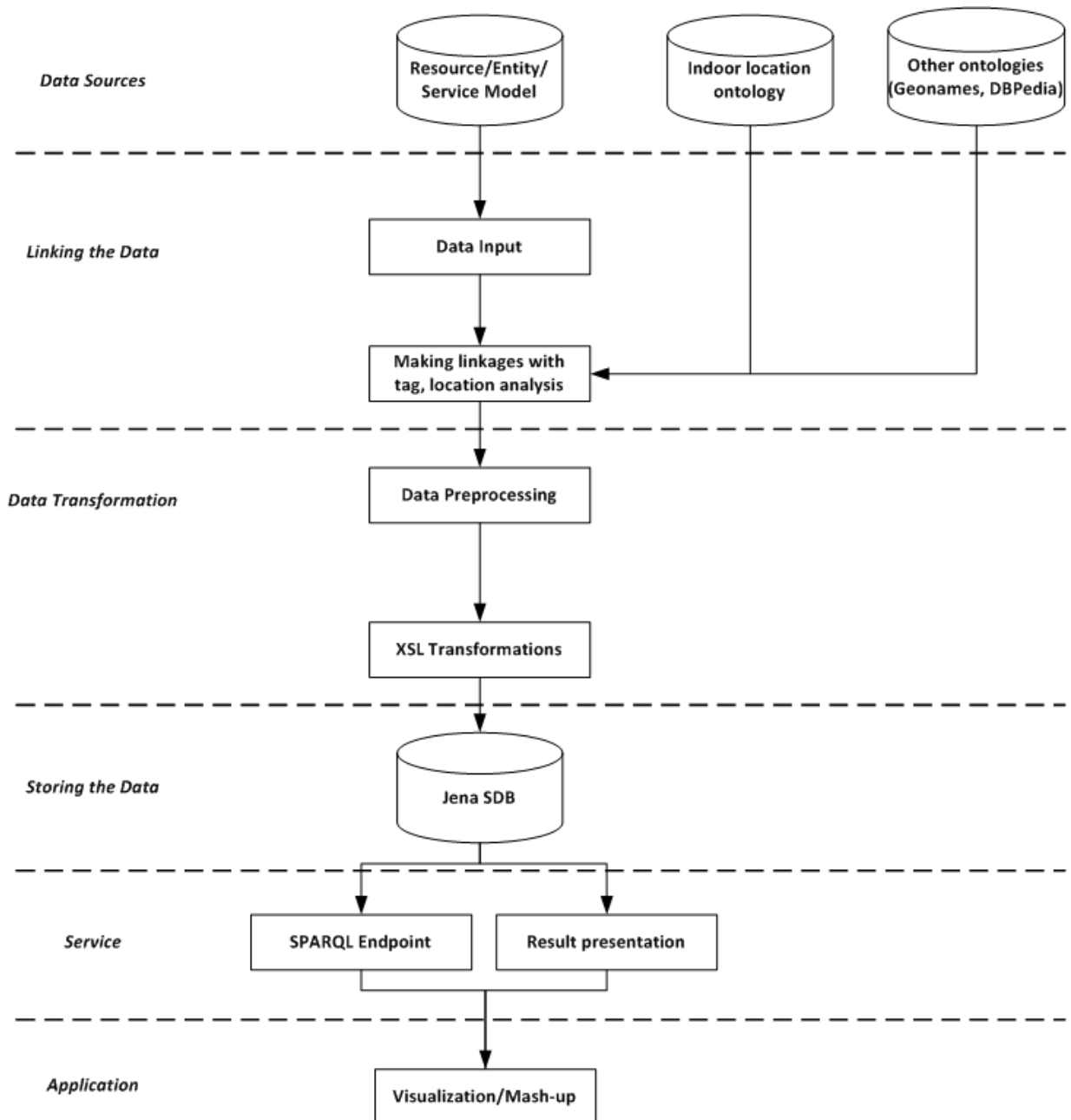


Fig. 4.1: Sense2Web architecture

methods, a web user interface is provided for H2M interaction, and RESTful interfaces are exposed for M2M interactions. The Sense2Web Web user interface is shown in Figure 4.2.

The different layers of the framework are explained below:

1. Data sources: the IoT information models detailing entities, resources and services that have been proposed in this paper (section 3) form the primary source of data structures for the H2M and M2M interfaces. In addition to these, the platform also accesses DBPedia and an indoor location ontology<sup>6</sup>

<sup>6</sup><http://ccsriottb3.ee.surrey.ac.uk:8080/IotaDataFiles/models/LocationModel.owl>



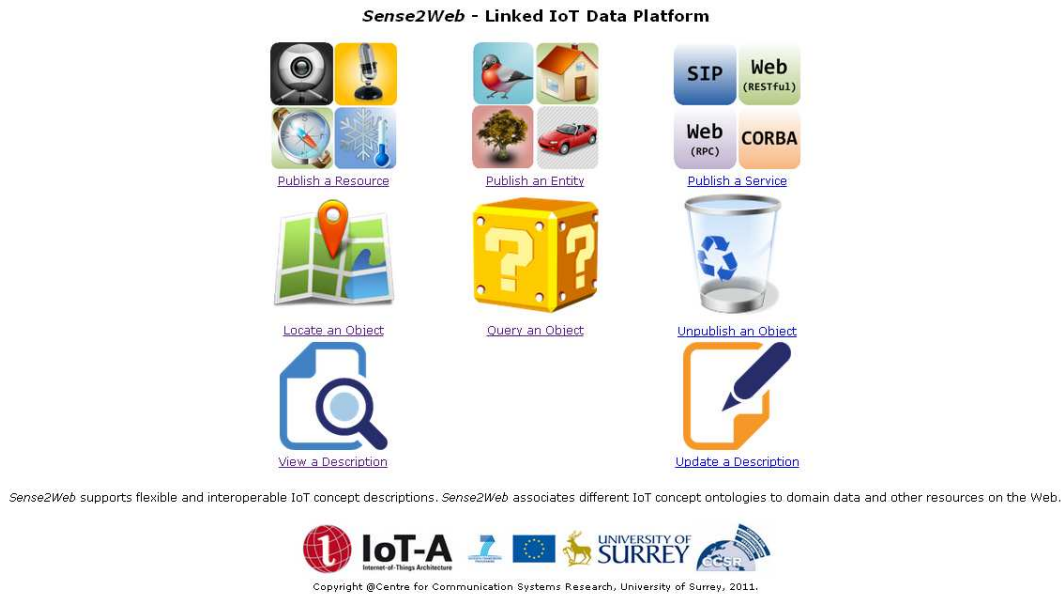


Fig. 4.2: Sense2Web user interface

to obtain values for location, type and descriptive properties.



Fig. 4.3: Entity Publication H2M Interface

2. Linking the data: the H2M interface consists of a Web interface with a form to populate the elements of the object (entity/resource) description. This constitutes the data input stage; Figure 4.3 shows the H2M interface for publishing an entity description.

To establish linkages with existing data repositories, we use Jena API to query the DBpedia and other resources and serialise the results using AJAX technology directly to the page; so the user can type a keyword and obtain relevant suggestions. For instance, in the Linked-data tag field, suggestions for relevant RDF links (URIs) to an input entered by the user in this field are retrieved from the DBpedia

knowledge base. This facilitates the interconnection and integration of data from different communities and sources. RDF link suggestions are also provided in the global and local location (from the indoor location ontology) fields with respect to the user's input. The form also provides location fields with respect to latitude and longitude. These fields make use of a Google mini-map which contains a marker which can be displaced to the position required, which will then populate the field with the respective co-ordinates.

The M2M interface is realised through a RESTful interface, developed using the Restlet API<sup>7</sup> that offers a RDF file upload option. The M2M interface is utilised for publishing instances of the service model.

3. Data transformation: When the form is submitted, the servlet handling the form processes the input data by collecting the fields and their respective values into an XML serialisation. This pre-processing step is succeeded by an Extensible Stylesheet Language Transformation (XSLT) [19] step which converts the input into an RDF instance that adheres to the corresponding IoT model (i.e. entity model in this case). This makes generation of the RDF data flexible and less dependent on the current model.
4. Storing the data: The RDF instance is then handed over to the SDB [12] interface, which then stores the RDF instance in SQL as nodes, triples, prefixes and quads.
5. Services: the platform supports retrieving, updating, or deleting a description, which can be done by providing the ID value of the published IoT concept. In addition to these methods, a SPARQL interface is provided for users to query for objects (resources or entities) or services of interest. Different results format are supported as well. The SPARQL [21] query page is shown in Figure 4.4.
6. Applications: an application can consume the services provided by the platform to make use of the published linked IoT concept instances. In the following section, we showcase a mash-up application that demonstrates the linked data usage and integration of data from different sources.

**5. Google Maps Mash-up Application.** The developed application is a map application that has been implemented using Google Maps API<sup>8</sup> to illustrate the location of the IoT instances and provide a summary of its description.

For this application, we use the location attributes and retrieve geographical coordinates of the resources and entities by processing the Linked data descriptions. The application retrieves related properties of the published IoT concept from the repository and lists available resources and entities through a Google Maps overlay. Figure 5.1 shows a screen-shot of the application and shows a published temperature sensing resource.

The map page refreshes periodically to show any changes in location that can be observed when object descriptions are updated. This is best noticed for example when remote sensor device gateways update Resource description when Resources migrate from gateway to gateway. The work in [8] has been integrated with the platform to demonstrate this scenario. In this scenario, a mobile sensor device attaches to a gateway. The gateway then creates a web service instance to expose the sensor resource to the web. The gateway also retrieves essential metadata from the sensor device and populates it in a RDF instance description which is then published to the Sense2Web platform via the M2M RESTful interface. As the sensor migrates to another gateway and re-attaches, the gateway will then update the description already stored at the platform, with the new location properties.

**6. Discussion.** To achieve scalability in real IoT deployments, the major issues involve providing semantic annotations, publishing the metadata, supporting large-scale distributed repositories and indexing and query support over the data. Manual resource annotation and tagging the data can hinder publishing large number of resources. Automating mechanisms are required to publish the resource and entity descriptions directly into the repositories. In a different work [24], we have studied and implemented a gateway component for large-scale sensor networks that publishes semantically annotated resource descriptions when the resources are discovered and associated to the gateway. This can help to automate the semantic annotation of resources. We have also implemented RESTful (M2M) interfaces that support direct publication and edit/update of the resources. The

<sup>7</sup><http://www.restlet.org>

<sup>8</sup><http://code.google.com/apis/maps/>

## OPTION 1: Query Form



## OPTION 2: Advanced Query (SPARQL)

Choose repository:  Resource  Entity  Service

Enter your SPARQL Query:

```
Select *
WHERE {
?uri om:hasName ?name.
?uri om:hasResourceID ?id.
?uri om:hasTag ?tag.
?uri om:hasType ?type.
?uri om:hasGlobalLocation ?globalLocation.
?uri om:hasLocalLocation ?localLocation.
?uri om:hasInterfaceType ?interfaceType.
?uri om:hasAccessInterface ?accessInterface.
?uri om:hasLatitude ?latitude.
?uri om:hasLongitude ?longitude.
?uri om:hasURITag ?uriTag.
}
```

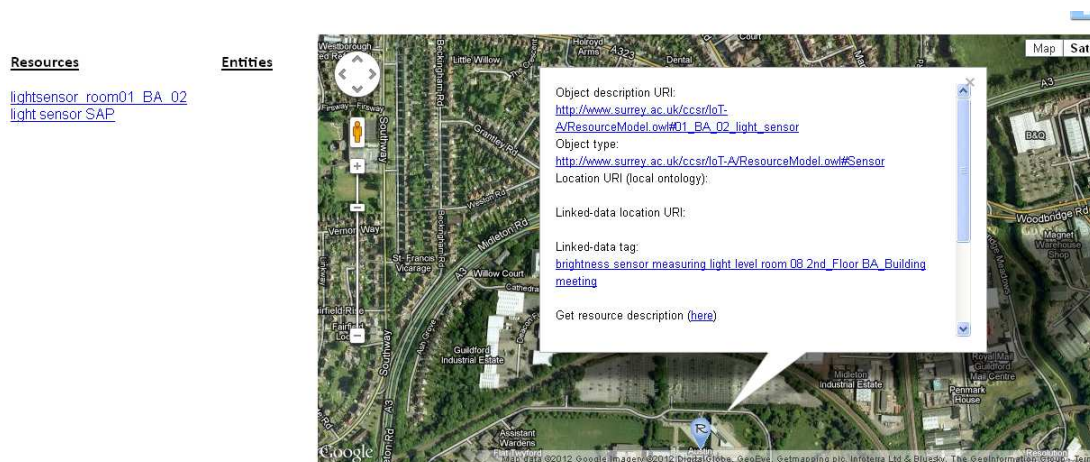
Result format:  XML  Text

Fig. 4.4: SPARQL Query page

interfaces can be accessed directly by third-party applications and software agents and can support automated semantic annotation and query of the resources. Federation of repositories and coordinating search and query over a number of semantic data stores in multiple domains can be also supported by publishing data in different domain repositories. The domain repositories can be defined based on network domain, geographical distribution or other aspects that can help to distribute the data more efficiently and then queries can be distributed based on the selected features. Peer-to-Peer communication and data update in the repositories is also another issue to enable up-to-date and efficient distribution and publishing of semantic annotation. These aspects will be investigated in future extensions of the presented platform.

**7. Conclusions and Future Work.** This paper presents a set of interlinked semantic models for the IoT domain and describes a platform that provisions both a H2M and M2M interface for publishing data descriptions conforming to the developed semantic models in the form of Linked Data. The platform allows making this data available to other Web applications via standard SPARQL endpoints. The models proposed in this paper are designed based on our previous work and experiences in the SENSEI project<sup>9</sup> and SSN ontology. The proposed models provide associations between different components in the IoT domain. The models support a

<sup>9</sup><http://www.sensei-project.eu/>



Copyright ©Centre for Communication Systems Research, University of Surrey, 2011.

Fig. 5.1: Google Maps mash-up application

semantic annotation framework so the legacy data can be also enhanced using these descriptions. The semantic annotation allows that the model data is represented as linked data and can be associated with the existing data on the Web and in particular Linked Open Data. Future work will involve development of a resolution framework that allows searching the large scale data of the instances of the models in the IoT domain and also automated inference of dynamic associations that can be identified by exploring and reasoning the interlinked descriptions.

## REFERENCES

- [1] ABANGAR, H., BARNAGHI, P., MOESSNER, K., TAFAZOLLI, R., NNAEMEGO, AND A., BALASKANDAN, K., *A Service Oriented Middleware Architecture for Wireless Sensor Networks*, In Proceedings of Future Network & Mobile Summit 2010, Florence, Italy.
- [2] AUER, S., BIZER, C., KOBILAROV, G., LEHMANN, J., CYGANIAK, C., AND IVES, Z., *DBpedia: A Nucleus for a Web of Open Data*, In Proceedings of the 6th International Semantic Web Conference (ISWC2007), 2007.
- [3] BARNAGHI, P., PRESSER, M., AND MOESSNER, K., *Publishing Linked Sensor Data*, In Proc. 3rd International Workshop on Semantic Sensor Networks (SSN), in conjunction with the 9th International Semantic Web Conference (ISWC 2010), 2010.
- [4] BERNERS-LEE, T., *Linked data*, Retrieved from <http://www.w3.org/DesignIssues/LinkedData.html>
- [5] BIZER, C., HEATH, T., IDEHEN, K., AND BERNERS-LEE, T., *Linked data on the web (ldow2008)*, In WWW '08: Proceeding of the 17th international conference on World Wide Web, New York, NY, USA. (pp. 1265-1266), 2008.
- [6] BOWERS, S., MADIN, J. S., AND SCHILDHAUER, M. P., *A Conceptual Modeling Framework for Expressing Observational Data Semantics*, In Q. Li et al. (Eds.), ER 2008, LNCS 5231, pp. 41-54, 2008.
- [7] CHRISTOPHE, B., VERDOT, V., AND TOUBIANA, V., *Searching the "Web of Things"*, In Proc. Fifth IEEE International Conference on Semantic Computing, Palo Alto, CA. (pp. 308 - 315), 2011.
- [8] ELSALEH, T., GLUHAK, A., AND MOESSNER, K., *Service Continuity for Subscribers of the Mobile Real World Internet*, In Proc. IEEE International Conference on Communications Workshops, 2011.
- [9] GLUHAK, A., KRCO, S., NATI, M., PFISTERER, D., MITTON, N., AND RAZAFINDRALAMBO, T., *A Survey on Facilities for Experimental Internet of Things Research*, IEEE Communications Magazine, 49(11), 58 - 67, 2011.
- [10] GUINARD, D., TRIFA, V., KARNOUSKOS, S., SPIESS, P., AND SAVIO, D., *Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services*, IEEE Transactions on Services Computing, 3(3), 223-235.
- [11] HAUSWIRTH, M., AND DECKER, S., *Semantic reality - connecting the real and the virtual world*, In Microsoft SemGrail Workshop. (pp. 21-22), 2007.
- [12] Jena, SDB - persistent triple stores using relational databases. Retrieved from <http://incubator.apache.org/jena/documentation/sdb/index.html>
- [13] KONING, H. P. D., ROUQUETTE, N., BURKHART, R., ESPINOZA, H., AND LEFORT, L., *Library for Quan-*

- tity Kinds and Units: schema, based on QUDV model OMG SysML(TM)*, 2009, Version 1.2. doi: <http://www.w3.org/2005/Incubator/ssn/ssnx/qu/qu.html>
- [14] LEGUAY, J., LOPEZ-RAMOS, M., JEAN-MARIE, K., AND CONAN, V., *Service oriented architecture for heterogeneous and dynamic sensor networks*, In Proceedings of the Second international Conference on Distributed Event-Based Systems, 2008.
  - [15] OASIS. (2009). Devices Profile for Web Services (DPWS). Retrieved from <http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01>
  - [16] SPIESS, P., KARNOUSKOS, S., GUINARD, D., SAVIO, D., BAECKER, O., SOUZA, L., AND TRIFA, V., *SOA-based integration of the Internet of things in enterprise services*, In Proceedings of IEEE ICWS, Los Angeles, CA, USA, 2009.
  - [17] W3C SSN Incubator Group Report. Retrieved from [http://www.w3.org/2005/Incubator/ssn/wiki/Incubator\\_Report](http://www.w3.org/2005/Incubator/ssn/wiki/Incubator_Report)
  - [18] SysML, O. Library for Quantity Kinds and Units: schema, based on QUDV model. 1.2. Retrieved from doi:<http://www.w3.org/2005/Incubator/ssn/ssnx/qu/qu>
  - [19] XSL Transformations (XSLT). W3C Recommendation v 1.0. from <http://www.w3.org/TR/xslt>
  - [20] OWL-S: Semantic Markup for Web Services. Retrieved from <http://www.w3.org/Submission/OWL-S/>
  - [21] SPARQL Query Language for RDF. W3C Recommendation, from <http://www.w3.org/TR/rdf-sparql-query/>
  - [22] YU, L., AND LIU, Y., *Using the Linked Data Approach in a Heterogeneous Sensor Web: Challenges, Experiments and Lessons Learned*, In Proc. Sensor Web Enablement (SWE) Workshop, Banff, Alberta, Canada, 2011.
  - [23] HORROCKS, I., PATEL-SCHNEIDER, P. F., BOLEY, H., TABET, S., GROSOFF, B., AND DEAN, M., *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, W3C Member Submission, 2004. Retrieved from <http://www.w3.org/Submission/SWRL/>
  - [24] GANZ, F., BARNAGHI, P., CARREZ, F., AND MOESSNER, K., *Context-aware Management for Sensor Networks*, In Proceedings of the 5th International Conference on COMMunication System softWARE and middlewaRE (COMSWARE11), 2011.

*Edited by:* Dana Petcu and Daniela Zaharie

*Received:* March 1, 2012

*Accepted:* April 15, 2012