

Ontologies for the Internet of Things

Sara Hachem, Thiago Teixeira, Valérie Issarny

► **To cite this version:**

Sara Hachem, Thiago Teixeira, Valérie Issarny. Ontologies for the Internet of Things. ACM/IFIP/USENIX 12th International Middleware Conference, Dec 2011, Lisbon, Portugal. hal-00642193

HAL Id: hal-00642193

<https://hal.inria.fr/hal-00642193>

Submitted on 17 Nov 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ontologies for the Internet of Things *

Sara Hachem, Thiago Teixeira, and
Valérie Issarny
INRIA Paris-Rocquencourt
firstname.lastname@inria.fr

ABSTRACT

Challenges the Internet of Things (IoT) is facing are directly inherited from today's Internet. However, they are amplified by the anticipated large scale deployments of devices and services, information flow and involved users in the IoT. Challenges are many and we focus on addressing those related to scalability, heterogeneity of IoT components, and the highly dynamic and unknown nature of the network topology. In this paper, we give an overview of a service-oriented middleware solution that addresses those challenges using semantic technologies to provide interoperability and flexibility. We especially focus on modeling a set of ontologies that describe devices and their functionalities and thoroughly model the domain of physics. The physics domain is indeed at the core of the IoT, as it allows the approximation and estimation of functionalities usually provided by things. Those functionalities will be deployed as services on appropriate devices through our middleware.

1. INTRODUCTION

The earliest recorded mention of the term “Internet of Things” (IoT) goes back to a presentation by MIT's Kevin Ashton in 1999. In it, he famously stated that adding RFIDS to everyday objects would create an Internet of Things ¹. And although his predicted IoT is undoubtedly becoming a reality, it now goes far beyond the original concept, encompassing not only RFIDS, but also sensors, actuators, mobile devices, and so on. In the new vision, all of these are considered to be *things* that can act upon, measure, or provide services based on real-world entities.

*This work is supported by the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement number 257178 (project CHOReOS - Large Scale Choreographies for the Future Internet - <http://www.choreos.eu>).

¹Kevin Ashton, RFID Journal, 22 June 2009: “I could be wrong, but I'm fairly sure the phrase ‘Internet of Things’ started life as the title of a presentation I made at Procter & Gamble (P&G) in 1999”

However, a number of challenges still stand on the way of the IoT. And perhaps their most visible effect is that the emerging networks of *things* are currently hard to deploy and operate, requiring — except in the most trivial cases — the intervention of highly-specialized field experts to interpret the sensor data and come up with actuation commands. This approach is clearly too costly and time-consuming, and simply does not scale as the IoT pushes into the general population.

To address the above, we have identified the following five core challenges [1] underlying the IoT:

- *Scale*: A sensing/actuating task that requires the cooperation and coordination of thousands of devices (within an Internet of billions), is often infeasible due to time, memory, processing, and energy constraints. Even a single application, such as calculating daily temperature variations around the globe, can require the use of millions of devices that will provide an amount of the information that will easily grow unmanageable.
- *Deep heterogeneity*: Sensor/actuator networks in the Internet of Things will be deployed by distinct entities, and the deployed hardware will display different operating characteristics, such as sampling rates and error distributions. Moreover, the degree of heterogeneity increases strongly as we move up from sensors/actuators components (chips) to the complex and diverse devices (sensor/actuator nodes) that integrate many of those components. It increases further as we move to local networks comprising numerous such devices and then global networks composed of those local networks and so on.
- *Unknown topology*: The Internet of Things is characterized by a network topology that is unknown and highly dynamic. This characteristic has two consequences. The first is that applications will require services that could once have been available and no longer are. The second is that services might themselves rely on devices that had once joined the network and left it abruptly (either permanently or temporarily), or may require devices that never existed at the desired geographical location in the first place.
- *Incomplete or inaccurate metadata*: A common solution to all challenges above is the use of semantic technologies to increment knowledge with metadata

[2, 3, 4, 5]. However, this requires input from human operators who are highly prompt to provide incomplete/inaccurate metadata. In addition, some of this information includes characteristics that change over time (e.g., calibration parameters).

- *Conflict Resolution*: Conflict resolution is an issue that arises mainly with actuators, but not so much with sensors. Conflicts arise, for instance, when multiple applications attempt to actuate the same device in opposing ways, or when they would like to exert mutually-incompatible changes on the environment.

In light of the challenges above, we propose a new middleware for the Internet of Things [1]. Our solution, as outlined in Section 2, is based on the service-oriented paradigm that abstracts *things* as *services*, therefore allowing us to keep *services* loosely coupled in order to increase reusability. A similar approach is not uncommon in the literature, being pursued, for instance, by [6, 7, 2]. Where our approach differs from others is that we propose an architecture that makes widespread use of approximations and estimations in order to address the IoT’s challenges without requiring the intervention of domain experts. For this, **in the backbone of our middleware, lies a knowledge base composed of three ontologies: a *Device Ontology*, a *Physics Domain Ontology*, and an *Estimation Ontology***. In this paper, we focus on defining these ontologies in detail (Section 3) and, later, contrasting them with those from the literature (Section 4). We, then conclude the paper in Section 5 with a brief discussion pointing to future work.

2. SOLUTION OVERVIEW

The overall architecture of our proposed IoT middleware is shown in Figure 1. As can be seen in the figure, our design consists of three main modules. The module on the right, the Knowledge Base (KB), is the main focus of this paper.

The information flow within our middleware is as follows. An application (or even a service) makes a sensing/actuating request, which gets handled by the Composition & Estimation module. In order to resolve the request, this module accesses the Discovery module and (especially) the Knowledge Base, creating a composition of services. Since we take a service-oriented approach, in this discussion we use the word “service” to refer to the various incarnations of *things*, such as sensors, actuators, etc.. A result is, then, obtained by executing this composition within the existing network.

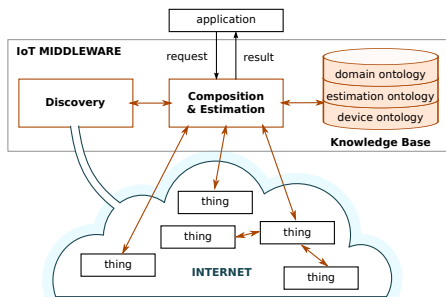


Figure 1: IoT middleware architecture

As introduced in [1], the novelty in our approach lies in the following characteristics of the system:

Approximate Composition: Composition is the process through which a dataflow graph is constructed, connecting available IoT services, in order to produce some desired result. Normally, service composition approaches aim to find the dataflow graph that is *optimal* in some metric. In this process, an ontology of services must be traversed, and all compositions that satisfy the input request must be found. However, in a network of billions of *things*, the problem-space of all possible dataflows (which is a combinatorial function on the number of *things*) becomes unmanageable. Therefore, to address both scalability and network dynamicity issues, we propose to seek an *approximately* optimal dataflow graph, rather than an *exactly* optimal one. For this, we require our knowledge base to carry an ontology of different approximation models for all sorts of physical processes. With this ontology, constructed by field experts, the composition module can use approximation functions to produce only a set of candidate dataflows with the highest likelihood of satisfying some predefined constraints. Then, the approximate solution to this composition is found by picking the optimal dataflow from this much smaller set.

Probabilistic discovery: Discovery is the process through which IoT services that match a set of required attributes are found in the network. Desired attributes can be the sensing modality, geographic location, or error characteristics, etc. However, to address the challenge of IoT scale, we go a step further by introducing the concept of *probabilistic discovery*, affecting both the device registration and device look-up processes that are natural parts of discovery. During registration, *things* will use a number of non-deterministic functions to randomly select the registries with which they will register; the registration times; and which metadata attributes to register with each registry. As for the look-up process, we require registries to be able to find a set of IoT services that best approximates the desired set, within some given time constraints. For this, a well-defined set of metadata attributes is a definite requirement, as well as an ontology describing how the attributes relate to one another, in order to allow us to substitute one service for the other when necessary. However, to best approximate services, issues such as the quality of service (QoS) of devices become extremely relevant. For this purpose, the middleware must be able to easily traverse the knowledge base to look up each device’s accuracy ratings, their related error functions, as well as the formulas for how different physical phenomena vary in space and time.

Automated Estimation: Estimation is the process through which the most likely value of a missing data point is approximated, by applying physical/statistical models on a dataset provided by a set of *things*. In our middleware, this estimation will be fully-automated and require no human intervention, thus allowing non-technical users to take full advantage of IoT services. The idea is that when the *Composition & Estimation* module discovers that the requested datapoint has not been captured by the available *things*, it should be able to query the KB for an appropriate estimation model, and then execute it within the network. To support this, the knowledge base must, therefore, contain an ontology of es-

timination methods, error models, probability distributions, and the spatio-temporal distribution of different physical phenomena.

What should be clear from the description of the three concepts above is that they all rely heavily on the information contained in the IoT Knowledge Base. As such, a fundamental contribution of our work is the development of the three aforementioned ontologies that comprise the KB. We build the KB from the ground up to support probabilistic discovery, approximate composition, and automated estimation, to end up with a multi-faceted Global Ontology that describes device metadata, domain-specific information, and mathematical models. To our knowledge this is the first ontology to unify these three aspects, and the only one to include and classify mathematical models in this manner. We describe the three faces of our Knowledge Base in more detail in the next section.

3. IOT ONTOLOGIES

An ontology is defined as “a formal, explicit specification of a shared conceptualization” [8] and is used to represent knowledge within a domain as a set of concepts related to each other. There are four main components that compose an ontology: Classes, relations, attributes and individuals. Classes are the main concepts to describe. Each class can have one or several children, known as subclasses, used to define more specific concepts. Classes and subclasses have attributes that represent their properties and characteristics. Individuals are instances of classes or their properties. Finally, relations are the edges that connect all the presented components.

3.1 Global Ontology for The IoT

We envision the representation of the IoT-based real world to be divided into 3 layers: a *physical layer*, i.e., *things*; an *information layer*, i.e., data and metadata about knowledge provided by *things*; and a *functional layer* comprising services provided by *things*. To match our vision of the real world and its representation by the Internet of Things, we aim at building an ontology that actually models all three layers. In fact, the physical layer is represented by a **Device Ontology**. The information and service layers are represented by a (**Physics and Mathematics Domain Ontology** and **Estimation Models Ontology**). To describe the ontologies more precisely:

Device Ontology: The **Device Ontology** models actual hardware *devices* that may exist in the network. For our middleware, it can be regarded as the device description repository that can be accessed for discovery.

Domain Ontology: The (**Physics and Mathematics**) **Domain Ontology** models information about real world physical concepts and their relations among each other. For our middleware, it can be regarded as the main repository to access for service composition.

Estimation Ontology: The **Estimation Ontology** contains information about different estimation models (“linear interpolation”, “Kalman filter”, “naive Bayesian learning”, etc.), the equations that drive them, the services that implement them, and so on. For our middleware, it can be

mainly regarded as the repository describing the device’s quality of service, and provides information needed for service composition. We aim at providing this ontology to be used as a reference by any middleware or application requiring IoT services, i.e., services provided by real world *things*. Those services, in most cases, generate approximate but never 100% accurate outcomes.

Most existing ontology work focused on modeling either *devices* as done, e.g., in MMI ontology² and [9, 10], or physics [11, 12] separately. The novelty of our approach is that it combines and takes advantages of the three ontologies by linking, all together, the domain of knowledge for sensing, actuating, and processing tasks and the real world representation through IoT services, that are aware of their environment. An important contribution is the level of abstraction at which we represent *things*, as we allow users to describe *devices* in an expressive manner while still avoiding complex details. In fact, as we target scalability, we consider simplicity in modeling knowledge to be an essential criteria. We argue that too much details might hinder the readability and quick traversability of the ontologies, thus effecting their scalability and usability. Of course the full ontologies are too large to be described in this paper. So, in the following, we outline only the most important concepts.

3.2 Device Ontology

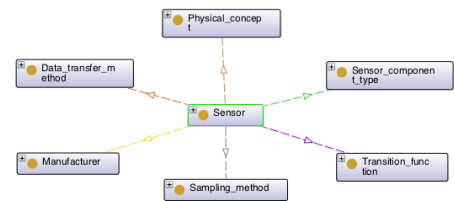


Figure 2: Sensor and related first class entities

As discussed in Section 2, the **Device Ontology** is accessed to identify what *things* should be looked up to satisfy an application’s requirements. We consider that applications built on top of an IoT middleware should be network and device agnostic. Therefore, it becomes the task of the middleware to identify what *devices* to seek in order to provide needed services. For this purpose, the ontology should clearly describe and yet not over-specify device metadata.

Our main contribution is the **high-level abstraction for device metadata**, especially regarding the internal components of *devices*. Internal components are the electronic chips and hardware parts, built inside the device, that together define its technical functionalities. Hence, looking at each of them separately as independent entities is not informative, as their functionalities are tightly related to one another’s. That being said, understanding the characteristics of a singular chip requires an understanding of the whole device’s internal schema, which grows to be too complex to include. We can further argue that they can just be considered as a black box, especially that those components are

²<http://marinemetadata.org/community/teams/ontdevices>

not directly accessible by applications. However, we chose to allow users to describe the internal components of *devices* (also done in [9, 10]) by their name and type only.

Another main contribution is that our ontology holds knowledge that is **independent of device deployments**, e.g., information related to the device's actual location. Instead, deployment information is presented in the metadata, reported by *devices*, during the discovery registration process. The ontology becomes thus easily pluggable with any middleware or application.

To elaborate on the ontology, we consider that IoT *devices* can be divided in four main classes:

- *Sensor*: A device that has the capability to measure a physical property of the real world.
- *Actuator*: A device that has the capability to perform an operation on or control a system/physical entity in the real world.
- *Processor*: A device that has the capability to perform computation operations on data.
- *Composite*: A device that consists of at least 2 of the *devices* above.

In the following, we focus on modeling sensors, as they are representative of *things* and models of other *devices* adhere to the same conceptualization approach. Based on the current literature [13, 14], we have identified several ontology concepts that are commonly used to model sensors (sensorML³). As shown in Figure 2, those concepts are:

- *Manufacturer*: The manufacturer of the sensor.
- *Sensor component type*: The sensor internal hardware components.
- *Physical concept*: The real world property measured by the sensor (e.g., temperature, wind speed, etc.). This concept is the main link between the **Device Ontology** and the **Physics Ontology**.
- *Sampling method*: The way the sensor is triggered to sample its environment (e.g., periodic).
- *Data transfer method*: The way the sensor is triggered to transfer its readings (e.g., push).
- *Transition function*: The process used to convert the input phenomenon to a digital value.

We chose the entities above as we consider that a sensor can be properly identified given any of their respective values. With the exception of the last entity, which we introduce as it clarifies what and how phenomena or values are output by a sensor after a measurement is performed. This is needed so that a sensor's outputs can be meaningful to and usable by other applications.

³<http://www.opengeospatial.org/standards/sensorml>

3.3 Physics Domain Ontology

The **Physics Domain Ontology** is created with two main goals. The first is to model real world entities as physical concepts so that any IoT middleware can extract knowledge about the real world, as this is a common task to be performed within the Internet of Things. The second is to model mathematical formulas and functions as they are the first alternative to be utilized when no device can provide needed services. The main classes of the **Domain Ontology** are:

- *Physical concept*: A real world object or property that can be measured.
- *Physical unit*: The output unit of the real world property measurement.
- *Mathematical datatype*: The set of numbers that can represent a real world property measurement.
- *Formula*: Mathematical expression that computes a numerical value representing a real world property.
- *Function*: Formulas are implemented by functions that define the required input and output machine datatypes.

Our main contribution in this ontology is that we model and establish a **direct relation between physical concepts, mathematical formulas and functions**. We argue that this relation is essential as it allows services to be provided as mathematical computations over physical concepts. This process can be used by any middleware to substitute services of unavailable *devices* with alternative services that can be deployed on any other appropriate device, which is a very familiar scenario within the highly dynamic IoT. This relation further allows our ontology to be useful in any context requiring mathematical and physical knowledge by clearly modeling formulas that can compute mathematical values as measurements over a physical property.

However, a same physical concept can have several formulas that vary based on the units of measurement of input/output parameters. Hence, our second contribution is to introduce two, not previously described, first class entities: **unit constraints** and **conversion formulas**. The former allows users to specify if a formula can have only one output and one input unit per concept, or can have a defined set of such units, or it stands correct for any input/output units, linked to a physical concept. For instance, the formula $speed = distance/time$ stands correct for any distance unit over any time unit. On the other hand, a windchill formula for temperature in Celsius and wind speed in km/h is different than that of a temperature in Fahrenheit and wind speed in mph. As for the conversion formulas class, it allows users to model conversion formulas between one measurement unit to another. By adding the constraints class and conversion functions to our ontology, we introduce a higher degree of flexibility as it allows any middleware or application using our model to dynamically adapt to unit constraints.

A common reference model for representing and categorizing physical concepts is the DOLCE⁴ representation. It is

⁴<http://www.loa-cnr.it/DOLCE.html>

adopted by several works such as [15, 11], as it has a well organized vocabulary. However, it does not categorize entities by their physical properties but rather by the human perception of those entities. This organization is not in line with our representation of concepts that should be both intuitive and physics oriented. In our ontology, each physical concept is linked to:

- *Sensor*: All sensors that can measure its value.
- *Units of measurement*: All units by which it can be measured.
- *Mathematical datatype*: Set the mathematical values the concept can be represented
- *Formula*: All mathematical formulas that can compute its value.

Regarding formulas, authors in [12] provide an approach to modeling physics in an ontology. However, their model is only applicable for the biological domain as they focus on mapping laws of physics to biological processes. SPACE is another ontology that models physics but it only applies in the space physics domains [16]. In our ontology, each formula is linked to:

- *Mathematical expression*: Mathematical equation.
- *Input parameters*: Measurements of physical concepts that will be used to evaluate another physical concept and their measurement units.
- *Output parameter*: Computed output and its measurement unit.
- *Physical concept*: The physical concept being evaluated.
- *Unit of measurement*: It is in fact the output's unit.

We argue that those concepts are well representative of physics and mathematical models and they specify all the parameters needed to define a mathematical equation. The Sensei project [2] models a decomposition of physical concepts into a set of other physical concepts. This decomposition can be similar to a direct link between our formula output and input concepts. However, the relation between their concepts is not clearly specified and therefore, their decomposition cannot substitute our formula model.

3.4 Estimation Ontology

The Estimation Ontology is, perhaps, the most unusual among the three described here. This ontology is in charge of storing the different *mathematical models* that make up the mental toolbox carried by expert system designers in fields such as Robotics, Estimation, Sensor Networking, etc. However, in addition to simply storing these models, the Estimation Ontology must also organize them in a way that makes them machine-accessible. After all, our middleware must be able to discern (1) which models are appropriate for a given situation, and (2) which model is, in some sense, optimal.

For this reason, the Estimation Ontology must provide a well-designed set of attributes for each model, as well as an intricate web of relationships between models, devices, and physical concepts. These rather indispensable elements, and how to best express them in this ontology, are something that we are currently investigating. For now, we limit ourselves to providing below a first-level set of classes that group the different types of mathematical models:

- *Estimation & Prediction Models*: These models are used during the automated estimation process, as well as before the look-up phase in probabilistic discovery. An example of such a model is a Kalman filter where each component in its matrices and vectors are functions of physical concepts from the Domain Ontology (for instance, an input vector for use in target-localization could be defined as a triplet of 3D-acceleration, 3D-velocity, and 3D-position).
- *Association & Correlation Models*: These describe the numerous conditional probability relations that are used in Estimation Theory, relating one physical phenomenon as a function of another. For instance, the probability of the value of a temperature sensor given the value of a daylight sensor. In addition, these models can also be used to solve the Association Problem that occurs, for instance, with multiple-target tracking;
- *Error Models*: These models describe the different ways that uncertainties can be introduced into measurements and actuations. These are usually represented in the form of a stochastic model (for instance, a simple additive Gaussian noise model).

4. RELATED WORK

Semantic technologies within IoT context are perceived as having three main benefits: high-level **abstractions** of complex information and incremented knowledge that provides a support for service **composition** and better **interoperability**.

Abstractions in IoT solutions, that integrate ontologies in their approach, can be divided in two categories: On the one hand, some IoT projects take advantage of ontologies as they abstract devices as services (such as in HYDRA[3, 4], SENSEI [2]). On the other hand, others use them as they abstract data/information as services (among which are SOFIA [17], SATware[18], Global Sensor Networks GSN[19], and Sensor-Masher[20]). A common approach towards this purpose is the use of *virtual/semantic sensors* [3, 4, 19] to abstract one or several physical *devices*. Similarly, [21, 20, 18] adopt the concept of semantic sensors and semantic streams. However, their implementation is different as they focus on abstracting data streams into higher level semantically rich knowledge. *Semantic devices* provide **composition** in some manner where the composed functions are specified at design time, and the mapping onto the network *devices* happens dynamically at run time.

To provide better **interoperability**, three aspects of the real world are modeled thoroughly in ontologies created within IoT solutions: *things*[2, 4], information and reasoning over data generated by *things* [13, 4, 20], and services [3, 4]. Some

projects go a step further by using ontologies to model context information [2], or dynamic reconfiguration, and adaptive resource management [22]. The target in [20], however, is the integration of sensor data streams into the World Wide Web rather than into an Internet of Things.

It should be noted that none of these solutions try to model and combine knowledge domains representing the real world into one global ontology as we do, to address the challenges presented in Section 1. They are however, mostly focused on modeling their ontologies for specific purposes only. Further, it is not clear how any of those ontologies are modeled to address scalability.

5. CONCLUSION

We presented in this paper a Global Ontology we are building for the Internet of Things. The Ontology models three aspects of the real world present in the Internet of Things. The first aspect is the “things” aspect described in a Device Ontology. The second aspect consists of real world concepts and functionalities of things, modeled in a Domain Ontology as mathematical formulas, and third is a real world approximation aspect that describes models to be used to approximate unavailable services and estimate missing information. The proposed ontology is at the core of a Service Oriented middleware for the Internet of Things, we are developing, that is scalable, flexible and provides the needed interoperability between deeply heterogeneous IoT components as detailed in[1].

Our future work will consist of further investigating the sensor modeling approach on different levels of details, as we plan on performing deeper comparison with existing solutions. We pay special attention to SensorML as it provides an appropriate modeling approach, although too detailed for our purposes. We later plan on investigating actuator and processor modeling approaches, but we consider they will strongly adhere to sensor models. Furthermore, we plan on modeling the estimation ontology comprising spatiotemporal and statistical correlation models of data. As for the middleware solution, we plan on implementing our vision and integrating the ontology to evaluate its feasibility.

6. REFERENCES

- [1] T. Teixeira, S. Hachem, V. Issarny, and N. Georgantas, “Service oriented middleware for the internet of things: A perspective,” in *ServiceWave*, to appear end of Octobre 2011.
- [2] M. Presser, P. Barnaghi, M. Eurich, and C. Villalonga, “The SENSEI project: Integrating the physical world with the digital world of the network of the future,” *IEEE Communications Magazine*, pp. 1–4, 2009.
- [3] W. Zhang and K. Hansen, “An evaluation of the NSGA-II and MOCell genetic algorithms for self-management planning in a pervasive service middleware,” in *14th IEEE International Conference on Engineering of Complex Computer Systems*, 2009, pp. 192–201.
- [4] M. Eisenhauer, P. Rosengren, and P. Antolin, “Hydra: A development platform for integrating wireless devices and sensors into ambient intelligence systems,” *The Internet of Things*, pp. 367–373, 2010.
- [5] W. Zhang and K. Hansen, “Semantic web based self-management for a pervasive service middleware,” in *2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, 2008, pp. 245–254.
- [6] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [7] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, “Interacting with the SOA-Based internet of things: Discovery, query, selection, and on-demand provisioning of Web Services,” *IEEE transactions on Services Computing*, vol. 3, no. 3, pp. 223–235, 2010.
- [8] N. Guarino, M. Carrara, and P. Giarretta, “An ontology of meta-level categories,” in *4th International Conference Principles of Knowledge Representation and Reasoning*. Citeseer, 1994, pp. 270–280.
- [9] H. Neuhaus and M. Compton, “The semantic sensor network ontology,” in *AGILE Workshop on Challenges in Geospatial Data Harmonisation, Hannover, Germany*, 2009, pp. 1–33.
- [10] M. Gomez, A. Preece, M. Johnson, G. De Mel, W. Vasconcelos, C. Gibson, A. Bar-Noy, K. Borowiecki, T. La Porta, D. Pizzocaro, et al., “An ontology-centric approach to sensor-mission assignment,” *Knowledge Engineering: Practice and Patterns*, pp. 347–363, 2008.
- [11] W. Kuhn, “A functional ontology of observation and measurement,” *GeoSpatial Semantics*, pp. 26–43, 2009.
- [12] D. Cook, J. Mejino Jr, M. Neal, and J. Gennari, “Bridging biological ontologies and biosimulation: the ontology of physics for biology,” in *AMIA Annual Symposium Proceedings*, vol. 2008. American Medical Informatics Association, 2008, pp. 136–140.
- [13] M. Eid, R. Liscano, and A. El Saddik, “A universal ontology for sensor networks data,” in *IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 2007, pp. 59–62.
- [14] J. Liu and F. Zhao, “Towards semantic services for sensor-rich information systems,” in *2nd International Conference on Broadband Networks*, 2005, pp. 967–974.
- [15] A. Devaraju, H. Neuhaus, K. Janowicz, and M. Compton, “Combining process and sensor ontologies to support geo-sensor data retrieval,” in *6th International Conference on Geographic Information Science*. Citeseer, 2010, pp. 1–5.
- [16] T. Narock, A. Szabo, and J. Merka, “Using semantics to extend the space physics data environment,” *Computers and Geosciences*, pp. 791 – 797, 2009.
- [17] J. Honkola, H. Laine, R. Brown, and O. Tyrkko, “Smart-M3 information sharing platform,” in *IEEE Symposium on Computers and Communications (ISCC)*, 2010, pp. 1041–1046.
- [18] D. Massaguer, M. Diallo, S. Mehrotra, and N. Venkatasubramanian, “Middleware for pervasive spaces: Balancing privacy and utility,” in *10th International Middleware Conference Middleware: ACM/IFIP/USENIX*, ser. LNCS, 2009, vol. 5896, pp. 247–267.
- [19] K. Aberer, M. Hauswirth, and A. Salehi, “Infrastructure for data processing in large-scale interconnected sensor networks,” in *International Conference on Mobile Data Management*, 2007, pp. 198–205.
- [20] D. Le-Phuoc and M. Hauswirth, “Linked open data in sensor data mashups,” *Semantic Sensor Networks*, pp. 1–16, 2009.
- [21] K. Whitehouse, F. Zhao, and J. Liu, “Semantic streams: A framework for composable semantic interpretation of sensor data,” *Wireless Sensor Networks*, pp. 5–20, 2006.
- [22] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, and V. Terziyan, “Smart semantic middleware for the internet of things,” in *5th International Conference on Informatics in Control, Automation and Robotics*, 2008, pp. 11–15.