Modeling and Querying Sensor Services using Ontologies

Sana Baccar[1], Wassim Derguech[2], Edwards Curry[2], and Mohamed Abid[1]

[1] CES Research Unit, National school of Engineers of Sfax, Tunisia
sana.baccar@ceslab.org, mohamed.abid_ces@yahoo.fr
[2] Insight Centre for data analytics, National University of Ireland, Galway, Ireland
wassim.derguech@insight-centre.org, edward.curry@insight-centre.org

Abstract. We propose in this paper a service description meta-model for describing services from a functional and non-functional perspectives. The model is inspired from the frame based modeling technique and is serialized in RDF (Resource Description Framework) using Linked Data principles. We apply this model for describing sensor services: modeling sensors and their readings enriched with non-functional properties. We also define a complete architecture for managing sensor data: collection, conversion, enrichment and storage. We tested our prototype using live streams of sensors readings. The paper also reports on the required time and storage size during the management and querying of sensor data.

Key words: Service Modeling, RDF, Linked Data, Functional Properties, Non-functional Properties, Sensors.

1 Introduction

Over the last years, there is a considerable interest in the Linked Data [1], which is considered as a best practice for publishing and interconnecting structured data on the Web. Recently, the service computing community tries to benefit from the Linked Data principles to overcome the limitations of the existing Semantic Web Services (SWS) approaches and make services easily interconnected, exchangeable and manageable [2]. Indeed, most of the current SWS approaches add semantics to the service by linking its syntactic operations and messages with some concepts of an external ontology. Although usable in some case, such description is still not able to offer complete information about the implicit knowledge, in addition to the difficulties encountered when it comes to re-use or compose one-to-many parts of the service.

To overcome the previous challenges, a new service description model based on the Linked Data principles is proposed in this paper. Our model describes a service as a structured entity featured via a set of capabilities (functional) and non-functional features. Such description considers a service as an access mechanism to a capability, which is, in its turn, a structured entity that describes what a service can do via an action verb and set of domain-specific attributes.

In order to evaluate the applicability of our model, we propose to use a Wireless Sensor Networks (WSNs) environment as a running example throughout this paper. The main reason of choosing such smart environment is its highly configurable and heterogeneous features. In fact, integrating semantics into WSNs is considered as a step toward the understanding, management and use of sensor-based data sources. This challenging integration includes various issues that are derived from the constrained nature of the sensor nodes (variability, unreliability and heterogeneity), their multiple domains and the need for multiple management of sensors through uncoordinated queries. Another aspect required for an effective analysis and decision making over sensor data is the integration of other sources such as enterprise and open data. For example: linking a sensor with its location (data coming from the enterprise) helps to take timely decisions tight to the location of the sensor (e.g., detecting high water usage in a particular location).

This work comes in the context of Waternomics project [3] aiming to create a Linked Water Dataspace [4, 5] as an emerging information management approach for collecting, standardizing, enriching and linking water usage data coming from sensors. Applying our proposed model and the proposed sensor data management infrastructure, facilitates to integrate various data sources for effective decision making.

In summary, the contributions of this work are:

- Propose an RDF-based Linked Data model for service description that facilitates service ex-changeability, discovery and composition processes. The model describes various aspects of a service (i.e, functional and non-functional).
- Developing a data integration module for generating RDF-based description of sensor service and interlinking them with enterprise data.
- Validating the proposed service model in a real-time Wireless Sensor Networks (WSNs) by defining sensor services and presenting them as reusable services, extensible and accessible for processing and management.

The remainder of the paper is organized as follows. Section 2 presents our conceptual model for describing services by detailing its different components. Section 3 describes our infrastructure for managing sensor data including: capturing, transforming, enriching, publishing and aggregating. In Section 4 we provide qualitative evaluation of our developed system. Before concluding in Section 6, we discusses some related works in Section 5.

2 Linked Data Model for Describing Web Services

At a high level, we share the same vision of OASIS Reference Model [1] and consider a service as an access mechanism to a *capability* under certain requirements (i.e., NFP properties). In the following, we describe these two concepts in detail.

[1] OASIS Reference Model for Service Oriented Architecture 1.0, http://www.oasisopen.org/committees/download.php/19679/soa-rm-cs.pdf

2.1 Capability

We adopt the definition of capability as defined in [6, 7]: "*A capability defines what a program (e.g., a service, a business process, a sensor) does from a functional perspective.*". We also adopt their proposed conceptual model that represents capabilities as an action verb and a set of domain related attributes.

Actually, the researchers proposed in [6, 7] a new conceptual model that describes functional capabilities as an action verb and set of domain specific attributes captured as an RDF-schema. Such description enables representing capabilities at several levels of abstraction, from the most abstract capability, named capability category, to the most concrete one, named capability offer, by explicitly extracting the relation between capabilities (i.e., specify and extend) and linking between the different levels. Such model deals with capabilities at different abstraction levels in a uniform way. Which takes into account attributes dynamicity, sensitivity and inter-dependency, as well as capability re-usability. This capability meta-model is formalized as an RDF vocabulary [2] which makes it a perfect fit to our work. We do not further elaborate on this meta-model as

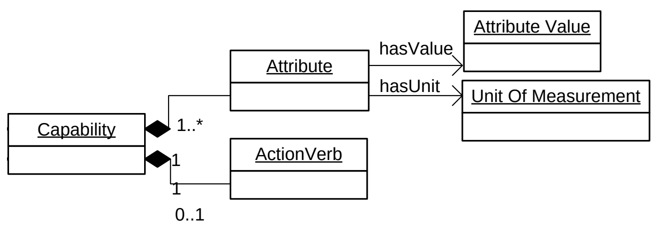


Fig. 1. Capability Meta-Model

it has been proved to be good in practice [6, 7]. However, our main contribution is centered towards the addition of non-functional properties that are described next.

2.2 Non-Functional Properties (NFPs)

As there will be many functionally-equivalent capabilities for a specific user request, we propose to consider the non-functional aspect as part of the service description. Such extension leads mainly to filter and select the best offer among these capability offers to better fulfill customer requirements and ensure his satisfaction. Separating the capability from the non functional constraints leads to reuse capabilities under different constraints. Moreover, in their turn, these constraints can be reused by different capabilities.

As it is shown in Fig. 2, the NFP concept contains four types of NFPs: (i) *user-generated* properties, e.g, reputation, (ii) *policy-related* properties e.g, cost and availability, (iii) *QoS* properties e.g., security and trust, response time, latency and reliability and (iv) *contextual specifications* e.g., location or business requirements. The NFPs are described by a combination of non-functional

[2] Available at: www.deri.ie/cap as accessed on 06/06/2014

(NF) terms (attribute, metrics)and their constraints (attribute values). Such

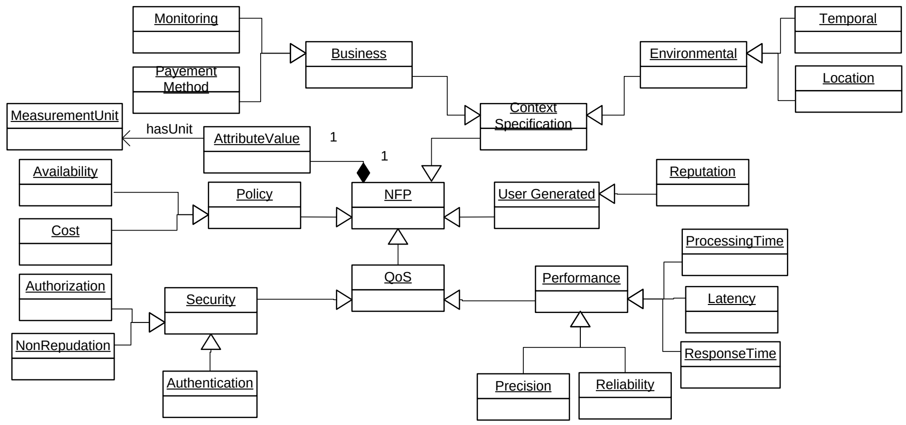


Fig. 2. Non-Functional Properties Meta-Model

NFP properties will ensure that a decision support model that uses data from a particular service is built with respect to these NFPs.

3 RDF-Data Management and Warehousing

A dataspace, as it is understood in the Waternomics project, is a data integration architecture. It allows integrating data from multiple sources into a single space. A dataspace for Waternomics is not only hosting sensor data but also other relevant data for decision analytics. Relevant data includes: sensor and location meta-data, weather data as well as other relevant data sources identified in the project. In this paper, we are interested in sensor and location meta-data that needs to be standardised, interlinked and published in order to facilitate its reuse internally (i.e., enrichment, aggregations, etc.) or externally (i.e., user or corporate applications). The following sections describe the process of collecting and transforming sensor data to RDF (in Section 3.1) then aggregating and storing this RDF data (in Section 3.2).

3.1 From Raw Sensor Data to RDF

For this paper, we assume that location meta-data (i.e., data describing the location of sensors) is available in RDF in our dataspace and is constantly updated using a collaborative approach, that we have previously experienced [8]. This location data is actually captured in our model as NFP properties for sensors as in essence they report only the location which is not part of the function of sensors.

In this section, we focus on sensor data management and propose the architecture depicted in Fig. 3 for this purpose. This figure depicts the entire process of collecting, transforming, enriching, publishing and aggregating sensor data.

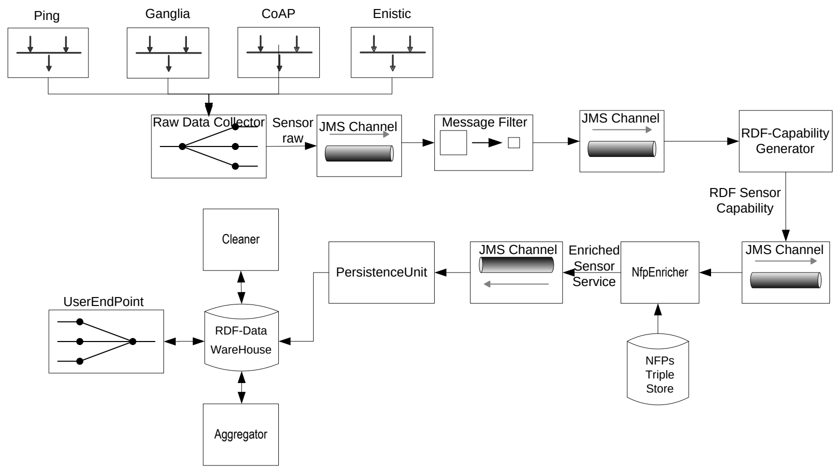


Fig. 3. Data Management and Warehousing Process

The main components of the process depicted in Fig. 3 are:

- *Raw Data Collectors* capture incoming data from heterogeneous data sources and publish them in plain text to a Java Message Service (JSM) channel. They are implemented as Universal Datagram Protocol (UDP) listeners, Constrained Application Protocol (CoAP) clients and RESTfull services, which periodically pull sensors data.
- The *Message Filter* is useful to refine and manage the flow of the incoming messages from the Raw Data Collector. It receives the collected messages and filters the irrelevant and unnecessary information in a way that enables the sensed data to be used on-the-fly for further processing.
- The *RDF-Capability transformation generator* identifies possible triples in the event-data, assigns a URI to each identified subject/predicate and represents event-data in RDF. Lines 1 to 10 of Listing 1 show an example of RDF-Capability in N3. In this RDF graph, *sensors:Sensor_0D6F00945B23* is a sensor that has the capability *sensors:SensingCap_d940a313* reporting on its sensed property and its observation time.
- The *Nfp Enricher* is responsible for enriching the service with the necessary non-functional properties retrieved from the Location meta-data store. After extracting the appropriate NFPs, the enricher merges the service capabilities and its adequate NFPs into a single RDF model that represents the entire service description. Listing 1 shows the result of the entire RDF transformation process where lines 12 to 15 show the NFP enrichment part.

Please note that for Listing 1, we are using multiple prefixes for the following:

- sensors: refers to the base URI for identifying sensors and their capabilities.
- snd: stands for Sensor Network Domain, is our customized domain ontology that captures the capability attributes.

– ssn: Semantic Sensor Network Ontology [3]. We use SSN as it is compliant to the W3C and OGC (Open Geospatial Consortium) standard SensorML.
– om: The Ontology of units of Measure and related concepts (OM). It models concepts and relations important to scientific research. It has a strong focus on units and quantities, measurements, and dimensions.

Listing 1. Event-Data converted to RDF

```
sensors:Sensor_0D6F00945B23 a snd:SensingService ;
cap:hasCapability sensors:SensingCap_d940a313 .

sensors:SensingCap_d940a313 a cap:Capability ;
cap:hasActionVerb snd:Sensing ;
snd:hasObservationValue
[snd:hasUnit om:cubic_metre_per_second-time ;
ssn:hasValue ``1.04''];
snd:hasObservedProperty om:Volumetric_flow_rate ;
ssn:endTime ``1396609412296''^xsd:long .

sensors:Sensor_0D6F00945B23 a snd:WaterVolumicFlowRateSensor;
ssn:featureOfInterest deriRooms:Kitchen ;
ssn:observes om:Volumetric_flow_rate ;
ssn:onPlatform sensors:a209be2b12e686b1b8 .
```

3.2 Data Warehousing

After a successful data transformation into RDF, we propose to store it into a data warehouse for further processing. Components that are manipulating this data store include:

– Data-Persistence Unit: We propose in our implementation the use of OpenLink Virtuoso [4] as data store. A dedicated data-persistence unit is continuously receiving enriched sensor data and saves it to the data store.
– Aggregator: Most of the sensors are frequently sending their readings. For example a water flow sensor is sending a flow rate reading each 10 seconds. It is, however, not necessary to keep every single reading in our data warehouse. In order to avoid handling a large warehouse containing every sensor reading, we propose to aggregate them into a single reading covering a longer period. This operation is done through the aggregator that computes the Sum, Average, Max, Min, etc. of the readings of every sensor. In other words, the aggregator removes all the capabilities of a given sensor that are collected for a predefined period and replaces them by a single aggregated capability.
– Data Warehouse Cleaner: Reading and identifying the required data from large data Warehouse is complex and extensive, both in memory size and time consumption. The cleaner is responsible for limiting the size of the Warehouse and

[3] http://www.w3.org/2005/Incubator/ssn/ssnx/ssn
[4] http://virtuoso.openlinksw.com as accessed on 06/06/2014

ensuring a good response time for data access, identification, and management by removing the old backups and maintaining the recent stored data.

The previously described components have been developed separately, communicate via a JMS server and store the data into the proposed RDF warehouse. We use ActiveMQ [9] as JMS server. It is a message oriented middleware that provides simple and easy to use methods to send, receive and handle JMS messages on the JMS channels. It is used in our system as a message exchange server using publish/subscribe protocols. These JMS messages are then broadcast to components that subscribe to those topics. The developed system is used for evaluation purposes reported in the following section.

4 Evaluation

For evaluating the developed system, we used a dedicated server running 64-bit Windows7 OS, with 4GB of Ram and an Intel Core i5 (2.66 GHz) CPU. We use for this evaluation a real-time sensor event collection and publishing them over ActiveMQ channels. We conducted two quantitative evaluations related to the required storage size and execution time for capturing, converting, enriching, storing and retrieving sensor data.

The first experiment that we conducted consists of evaluating the execution time required for managing sensor data. As it is shown in Fig. 4(a), the required execution time for executing the entire management process presented previously in Fig. 3 increases linearly to the sensor event frequency. This indicator is important to consider in managing environments with large number of sensors and sensor readings, however, due to the fact that we are using a relatively small environment this issue can not be considered as a problem.

The second evaluation carried out consists of showing the impact of the raw event data, sensor capability and sensor service on the size of the data warehouse. As shown in Fig. 4(b), the size of the processed data depends on the size of the incoming events from each sensor, the frequency of the events and the number of the non-functional properties. We notice that there is not a big difference in size after the enrichment step as most of the sensors that we consider share the same non functional properties that are already stored in the data warehouse and are not duplicated for each new sensor reading. Furthermore, we notice a considerable difference when comparing the size of raw sensor readings with the RDFized data. This is obviously expected as we are generating a complete description using additional enterprise data during the enrichment step.

As the size of the warehouse is critical either for resource management or search query processing, we propose the aggregation module that helps to reduce the size of the data warehouse. Using this aggregator, we reach up to 71% of compression rate as shown in Fig. 4(c). Moreover, the difference in size between the raw sensor readings and the aggregated sensor readings is not big because of the compression rate gained during the aggregation. Even though we have lost the entire readings by aggregating them into a single entry, we gained by having a semantically described and reduced set of sensor readings which also help reducing search queries processing time. Indeed, the proposed approach allows describing heterogeneous sensors uniformly which helps in simplifying the querying process of sensor data.

Finally, to test the performance of our system in data mining and discovery, we define a set of different queries (shown in table 1), we execute them on different number of triples stored in the data warehouse and we observe the required query execution time. As we can see in Fig. 4(d), the variations in query execution time is due to variation in the number of triples stored in the data warehouse. As an example, the time taken to process a query is between 509 and 537 milliseconds for around 450 RDF-triples; 528 and 546 milliseconds for 1033 triples; 537 and 548 milliseconds for 2376 triples and 556 and 577 milliseconds for 3350 triples. The system takes between 24 and 50 milliseconds to retrieve required data from 2904 triples. We notice that there was a significant difference in the query execution time when considering all sensor services and after the aggregation process is executed.

Table 1. Evaluation Queries

Q1	Search per 12 minutes the average of the water-usage in the kitchen.
Q2	Search per minute the sum of the water-usage the showers.
Q3	Search the min and the max of the water-usage captured in shawers during the last 10 minutes.

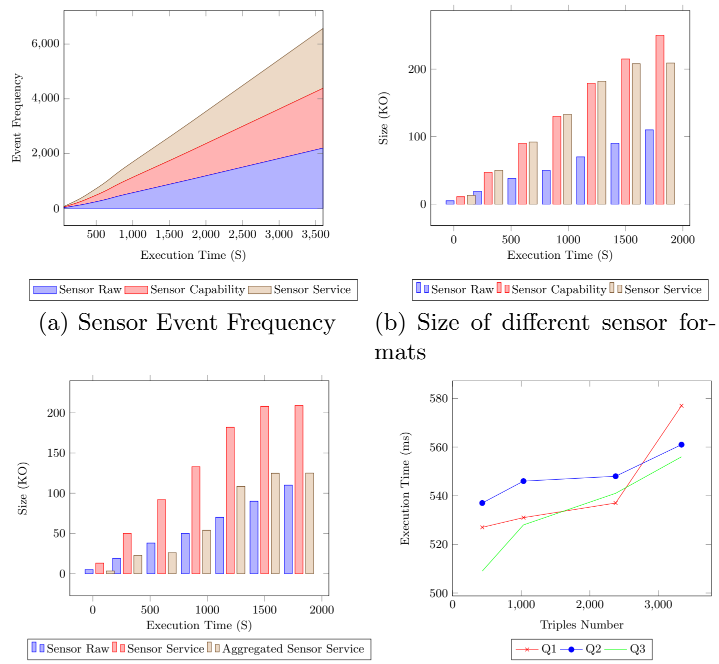
After a random selection and verification of generated capabilities we notice that the system is operating as expected. It is important to notice that the use of a triple store as a data warehouse helps reduce the processing time and complexity of data in contrast to a classical database system. Indeed, OpenLink Virtuoso [5] was hiding the complexity of the data storing process for example when handling duplicate triples.

In conclusion, using our proposed modeling approach in a small smart environment was effective in producing standardized data using RDF. Even though our approach drastically increases the size of raw data, it is still stored in an acceptable size and processed and queried in a reasonable time.

5 Related Work

The work presented in this paper relates to two main research areas: i) Semantic Web Service Description and ii) Sensor data integration and management.

[5] http://virtuoso.openlinksw.com as accessed on 06/06/2014



(a) Sensor Event Frequency

(b) Size of different sensor formats

(c) Data Size before and after aggregation

(d) Query Execution Time

Fig. 4. Performance Charts

5.1 Semantic Web Service Description

Recently, several ontologies and languages have been proposed to describe Web services using semantics. SA-WSDL [6] [10] uses ontologies to enrich WSDL descriptions and XML Schema of Web services with semantic annotations. The effort of this contribution was put towards modelling services as invocation interfaces which relate mainly to the service implementation. However, in this paper, we consider a service capability as an entity featured via a set of attributes. Using such definition, we share the same vision of OASIS Reference Model that considers a service as an access mechanism to a capability. Within this vision, the invocation interface is only one aspect of the whole service description.

In a more refined fashion, languages such as OWL-S [11], WSMO [12], SWSO [7], provide a semantic description of Web services. These approaches do not go beyond the classical Input, Output, Precondition and Effect paradigm to define services descriptions. These contributions focus more on the machine processing of service descriptions while ignoring end-users intuitive way of describing their needs (i.e., attributes and their values). We adopt in our work the attribute-

[6] http://www.w3.org/TR/sawsdl/
[7] http://www.w3.org/Submission/SWSF-SWSO/

featured approach for describing services for serving both machine processing and user-centricity.

[13] propose to give more flexibility to users in searching services while using natural language text and approximate semantic matching. This technique is useful when queries are created without knowing in advance service descriptions. This is not the object in our work, however, our plan is to use the approximate semantic matching of events for the internet of things explored in [14].

5.2 Sensor data integration and management

Managing smart environments requires extensive efforts for collecting, integrating and analysing sensor data. An important step towards an efficient smart environment management is the standardisation of sensor data. Using RDF in such context is a solution that has been also adopted in [15] and [16]. Here, authors specifically address the problem of managing sensor data and propose to transform sensor readings into RDF and generate data cubes on-the-fly from syntactic sensor data. Contrary to our contribution, in these works all the event features are considered as data cube parameters and there is no separation between the sensor capability and its non-functional properties.

Moreover, since event processing requires providing an immediate response over the continuous stream of sensor data, a system that stores generated events and provides fast response to complex queries is needed [17]. To handle such requirements, various database systems have been adopted such as in [18, 19]. The proposed systems, have several critical shortcomings that prevent using them directly to process sensor data. In fact, they are too heavyweight and slow, devoting much complexity to handling queries that are based on indexing and query forwarding techniques. To overcome such gaps, we propose in this paper a system that uses an RDF data-warehouse for managing sensor events. Indeed, our data-warehouse is a large store of RDF triples containing both live and historical aggregated data that overcomes the rigidity of a classical database.

Similar to [20], we use SSN in our sensors descriptions. However, [20] go beyond this and propose a naming convention and a data distribution mechanism for a large-scale sensor-based environment.

6 Conclusion

Data integration and processing from sensor services is a challenging issue that requires data integration, standardisation, linking and enrichment. We proposed in this paper a meta model for describing sensors and their produced data using RDF. Our model explicitly differentiates between sensor service functional and non-functional aspects. We defined and implemented an architecture for manipulating sensor data in order to serve as a base for further decision support and data analysis. Using RDF and Linked Data principle in such context helps building a large linked data cloud that makes our data set together with other open sets as a single database.

The proposed system has been tested on a real world scenario showing its applicability and its potential use in data mining and service discovery process. Our ongoing work focuses on extending our system to handle service composition task and support complex queries. However, from sensor service modeling perspective, using capability description in its original form is insufficient when it comes to the composition and planning scenarios. Indeed, we need two main additional attributes that capture the state/change of the world before/after executing the corresponding action. For that, we propose an extension to the introduced capability meta-model (precondition and effect) so that the semantics of capability is captured at two levels: i) Coarse-grain level: handles the discovery process through a combination of an action verb, domain specific attributes and semantic links between capabilities. ii) Fine-grain level: handles the composition process via a set of preconditions and effects. The evaluation of the actual meta-model is also part of our future plan.

From an architectural point of view, the main problem with our proposed approach is the fact that it is not designed for large scale environments. As part of the Waternomics project, we are interested in processing real-time data in large environments and this leads us to investigate other architectural styles that help overcome this problem. We are particularly interested in investigating the Lambda Architecture for our future plan. We also plan to extend this architecture for generating additional links between our data store to open data for more knowledge discovery.

Acknowledgments

The research leading to these results has received funding under the European Commission's Seventh Framework Programme from ICT grant agreement WATERNOMICS no. 619660. It is supported in part by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289.

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semantic Web Inf. Syst. **5**(3) (2009) 1–22
2. Heath, T., Bizer, C.: Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers (2011)
3. Clifford, E., Coakley, D., Curry, E., Degeler, V., Costa, A., Messervey, T., Andel, S.J.V., de Giesen, N.V., Kouroupetroglou, C., Mink, J., Smit, S.: Interactive Water Services: The Waternomics Approach. In: 16th International Conference Water Distribution Systems Analysis (WSDA 2014), Elsevier (2014)
4. Curry, E.: System of Systems Information Interoperability using a Linked Dataspace. In: IEEE 7th International Conference on System of Systems Engineering (SOSE 2012), Genoa, Italy, IEEE (2012) 101–106
5. Curry, E., Hasan, S., O'Riáin, S.: Enterprise Energy Management using a Linked Dataspace for Energy Intelligence. In: The Second IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT 2012), Pisa, Italy, IEEE (2012)

6. Bhiri, S., Derguech, W., Zaremba, M.: Modelling capabilities as attribute-featured entities. In Cordeiro, J., Krempels, K.H., eds.: WEBIST (Selected Papers). Volume 140 of Lecture Notes in Business Information Processing., Springer (2012) 70–85
7. Derguech, W., Bhiri, S., Hasan, S., Curry, E.: Using Formal Concept Analysis for Organizing and Discovering Sensor Capabilities. The Computer Journal (2014)
8. Hassan, U., Bassora, M., Vahid, A., O'Riain, S., Curry, E.: A collaborative approach for metadata management for internet of things: Linking micro tasks with physical objects. In: Collaborative Computing: Networking, Applications and Worksharing (Collaboratecom). (Oct 2013) 593–598
9. Snyder, B., Bosanac, D., Davies, R.: ActiveMQ in Action. Manning Publications Co., Greenwich, CT, USA (2011)
10. Saquicela, V., Blzquez, L.M.V., Corcho, s.: Adding semantic annotations into (geospatial) restful services. Int. J. Semantic Web Inf. Syst. **8**(2) (2012) 51–71
11. Martin, D., Paolucci, M., Wagner, M.: Bringing semantic annotations to web services: OWL-S from the SAWSDL perspective. In: The Semantic Web. Volume 4825 of LNCS., Springer Berlin Heidelberg (2007) 340–352
12. Roman, D., de Bruijn, J., Mocan, A., Lausen, H., Domingue, J., Bussler, C., Fensel, D.: WWW: WSMO, WSML, and WSMX in a Nutshell. In: ASWC. (2006) 516–522
13. Sangers, J., Frasincar, F., Hogenboom, F., Chepegin, V.I.: Semantic web service discovery using natural language processing techniques. Expert Syst. Appl. **40**(11) (2013) 4660–4671
14. Hasan, S., Curry, E.: Approximate semantic matching of events for the internet of things. ACM Trans. Internet Technol. **14**(1) (August 2014) 2:1–2:23
15. Mehdi, M., Sahay, R., Derguech, W., Curry, E.: On-the-fly generation of multidimensional data cubes for web of things. In: IDEAS. (2013) 28–37
16. Lefort, L., Bobruk, J., Haller, A., Taylor, K., Woolf, A.: A linked sensor data cube for a 100 year homogenised daily temperature dataset. In: SSN. (2012) 1–16
17. Balazinska, M., Deshpande, A., Franklin, M.J., Gibbons, P.B., Gray, J., Hansen, M., Liebhold, M., Nath, S., Szalay, A., Tao, V.: Data management in the worldwide sensor web. IEEE Pervasive Computing **6**(2) (2007) 30–40
18. Diao, Y., Ganesan, D., Mathur, G., Shenoy, P.J.: Rethinking data management for storage-centric sensor networks. In: CIDR. (2007) 22–31
19. Madden, S.R., Franklin, M.J., Hellerstein, J.M., Hong, W.: Tinydb: An acquisitional query processing system for sensor networks. ACM Trans. Database Syst. **30**(1) (March 2005) 122–173
20. Barnaghi, P.M., Wang, W., Dong, L., Wang, C.: A linked-data model for semantic sensor streams. In: Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, IEEE (2013) 468–475