Electronic Communications of the EASST
Volume X (2011)

Workshop on Semantic Services for the Internet of Things (SSIT 2011)

Semantic-Service Provisioning for the Internet of Things

Daniel Bimschas, Henning Hasemann, Manfred Hauswirth, Marcel Karnstedt, Oliver Kleine, Alexander Kröller, Myriam Leggieri, Richard Mietz, Alexandre Passant, Dennis Pfisterer, Kay Römer, and Cuong Truong

12 pages

Guest Editors:
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Semantic-Service Provisioning for the Internet of Things

Daniel Bimschas[1], Henning Hasemann[2], Manfred Hauswirth[3], Marcel Karnstedt[3], Oliver Kleine[1], Alexander Kröller[2], Myriam Leggieri[3], Richard Mietz[1], Alexandre Passant[3], Dennis Pfisterer[1], Kay Römer[1], and Cuong Truong[1]

[1] University of Lübeck, Germany
{bimschas,kleine,pfisterer}@itm.uni-luebeck.de
{mietz,roemer,truong}@iti.uni-luebeck.de
[2] University of Braunschweig, Germany
{hasemann,kroeller}@ibr.cs.tu-bs.de
[3] Digital Enterprise Research Institute, National University of Ireland, Galway
{manfred.hauswirth,marcel.karnstedt,myriam.leggieri,alexandre.passant}@deri.org

Abstract: The increasing convergence of the Internet as the global information backbone and embedded systems and sensors as key information providers on the state of the physical world to this backbone requires efficient design and implementation methodologies which do not exist to date. Currently, the development of applications spanning both worlds is cumbersome and inefficient due to the lack of suitable frameworks and the limited understanding of the two development communities for the problems and underlying assumptions which need to be taken into account of "on the other side of the fence." The goal of any methodology should be to provide the elegance and widespread adaption of Web-based standards in combination with the efficiency of the underlying embedded system and sensor layers, abstracting away "unnecessary" details while providing powerful paradigms which enable the necessary level of control in terms of APIs. This paper investigates unified concepts, methods, and software infrastructures that support the efficient development of applications across the Internet and the embedded world, minimizing development efforts through flexible data and meta-data driven integration based on Semantic Web technologies.

Keywords: Service-Oriented Architectures, Semantic Web, Internet of Things, Sensor Networks

1 Introduction

We are currently witnessing the integration of two long-lost relatives: On the one hand, general-purpose computing and its networking backbone, the Internet, and on the other hand, embedded computing. Standards enabling embedded IP networking such as 6LoWPAN are currently being implemented in low-footprint embedded IP stacks [Sen, DAW+08] and used in first commercial products. Mobile phones include an increasing number of sensors and connect to the Internet via a variety of wireless networking standards such as WiFi, Bluetooth, GPRS, and UMTS. With this trend to continue, the Internet will extend into the real world, providing online, real-time access to and control over the state of real world objects and places. Moreover, this will allow the integration of resources and services available in the traditional Internet with novel real-world services offered by embedded networked sensors, giving rise to a new class of applications.

However, the development of applications exploiting this combined infrastructure is currently cumbersome and difficult: Despite the IP-based integration of the embedded world, application-level protocols, software and development environments, and design and evaluation methodologies in the Internet and in the embedded world are vastly different and lack integration. The developer currently has to bridge this gap "manually" and has to be an expert in both worlds.

This paper takes on these challenges and investigates unified concepts, methods, and software infrastructures that support the efficient development of applications across the Internet and the embedded world. The key metric is the *effort* required for developing robust, interoperable, and scalable applications in the Internet of Things (IoT). Our starting point is the observation that the Web has opened the Internet to the greater public by providing open standards resulting in a usable and interoperable software infrastructure, and well-understood methodologies for design, implementation, and evaluation. Our approach is to extend the Web into the embedded world to form a Web of Things (WoT), where Web representations of real-world entities offer services to access their physical state and to mash up these real-world services with traditional services and data available in the Web to create novel applications.

From an abstract point of view, the main task of IoT application developers is obtaining the data for a specific task. In distributed systems, this requires (1) to identify entities holding the data and (2) to retrieve them. In other words, developers must find the relevant services to query for the data as required by their application. This identification of services is achieved by selecting services based on meta-data about their corresponding entities. Today, the identification of these services is typically a task performed manually. For instance, developers know the addresses (e.g., device-ID or IPv6 address) of devices and invoke these to obtain the required data. In the Internet of Things, where millions of devices interact, such an approach does not scale, offers only limited flexibility at run-time, and hampers the development of novel applications.

In this paper, we propose a methodology to simplify IoT application development. Our approach combines technologies from the Internet of Things and the Semantic Web to provide this service efficiently. The central idea is to let entities provide self-descriptions of their type, capabilities, services, etc. in a machine-readable manner. This is indexed by semantic search engines, which applications can query. The queries return a set of services, which are then used to obtain current data from entities. Such result sets can contain large amounts of services to be queried, leading to long processing times and high energy consumption. It is therefore crucial to sort the results such that services with a high probability of having a certain state are queried first. Another issue is that individual services may be too fine-grained for some applications. We propose a means to semantically aggregate individual services to higher-level entities. The key benefit of this approach is that it enables a completely different way of programming IoT applications. Instead of custom-tailored applications that are tightly integrated at a coding level, we provide a solution which enables the dynamic use of the best resources driven by a concept of data-centric integration while providing the same level of efficiency.

This paper is organized as follows: Section 2 introduces a use case which will serve as a running example throughout the paper. Section 3 discusses the state-of-the-art and related work. Section 4 introduces the design and components of our approach to simplify IoT application development. Section 5 concludes this paper with a summary and an outlook for future work.

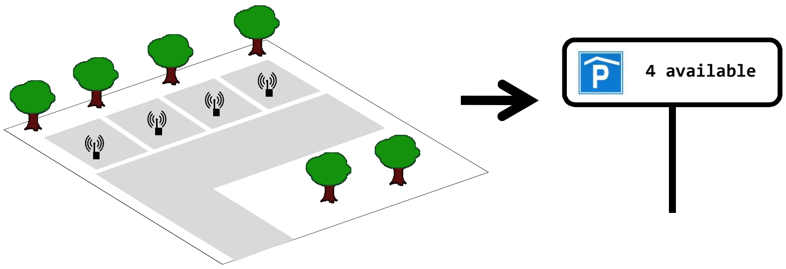

Figure 1: Parking lot use case

2 Use Case

In this section, we propose a use case for our approach, i.e., a concrete scenario that allows for several applications. Each of these is technically a simple aggregation of services, while providing people with valuable information.

Consider car parks, operated by different companies (cf. Figure 1). Assume that all parking spaces are equipped with sensor nodes to detect whether they are occupied. For robustness reasons, most spaces are monitored by several sensors. The sensors are connected wirelessly to gateway computers that have access to the Internet.

A first application provides the car park owner with live information about occupancy. This information can be simply used to trigger ”free/occupied” signs at the car park entrance. The available status information allows establishing an automatic guidance using electronic signs to help customers entering the area in finding a free spot.

The second, more valuable, application is available for drivers looking for parking spots. A driver could use some Internet-enabled device such as his Smartphone or navigation system to run a query that returns car parks with free spots in his vicinity. Ideally it would find closer spots first, considering far-away car parks only when necessary. This application can be a Smartphone app, using the technology presented in this paper to find parking spots matching certain criteria (such as considering female-only spots only for women). The application could also be a service, hosted on Internet servers, with some standard interface such as a website or a phone call.

We let the operator of each car park decide for himself how much information he wishes to publish. Some may allow direct access to the sensors, some might just push regular summaries, and some others may offer a service to query only whether there is free space or not. As we will describe in the following sections, our approach allows building the aforementioned applications even under these constraints, in a simple and straightforward manner.

3 State of the Art

Services. The basic abstraction that we believe to be the most promising for the IoT is service-orientation. This conjecture is backed by recent publications, e.g. [RGB+09], in which the authors attach great importance to service-based abstractions. In the literature, the most promising foundation for the integration of IoT devices with the current Internet is the concept of Service-

Oriented Architectures (SOA). The downside of SOAs is that the technologies widely used to realize them – such as XML, SOAP, Web Services and the Business Process Execution Language (BPEL) – are not applicable in the IoT domain, due to the different and resource-constrained nature of the class of devices. But, some approaches exist to apply these technologies on resource-constrained devices. One class of such approaches tries to enable Web Services on embedded devices (e.g. the ones emerging out of the EU FP7 SOCRADES [FSA08] project or Microsoft Research [PKGZ08]). Another approach is the use of efficient compression of the XML-based Web Service messages (i.e., SOAP messages), e.g. [TH02, Wor]. A third class of approaches tackles the interconnection of IoT devices with the Internet. Here, one trend is to enable IP networking on resource-constrained devices (e.g. 6LowPAN [MKHC07]). Some prefer gateway solutions, e.g. [SRMA09], while still others propose using a network-agnostic transport protocol for Web Service messages usable in the Internet and the IoT domainm, e.g. [WBJ05].

Efforts to establish a standards based solution have gained great momentum recently by the development of the Constrained Application Protocol (CoAP) [SFS10] in the IETF Constrained RESTful Environments (CoRE) Working Group. CoAP provides a subset of HTTPs methods on top of UDP and is especially designed for small scale embedded devices, enabling both consumption and provision of RESTful Web Services [Fie00] in the IoT.

Semantics. Various representation models exist for sensor data and sensor networks, including standards defined within the Open Geospatial Consortium's Sensor Web Enablement initiative [OCG10]. However, due to the lack of a machine-understandable representation, they do not provide semantic interoperability, nor a basis for reasoning that can simplify the development of advanced applications. Ontologies available on the Web [RKT05, M. 07] and related semantic technologies are being designed to overcome these issues. However, due to their specialization, these ontologies do not cover all relevant features of sensor networks. Thus, the W3C Semantic Sensor Network Incubator Group (SSN-XG) [W3C10] proposed a general ontology to be further integrated with external ontologies. They extended the Sensor Markup Language [OCG] to support semantic annotations. This provides a coherent framework compliant with efforts in the Semantic Web and the Linked Data movement [BHB09]. The important challenge now is to develop and choose the right ontologies to integrate with the SSN-XG in order to improve the representation of sensor data and sensor networks. Additionally, the integration of social networks and sensor networks is getting into the focus of active research [BDH$^+$09].

Automatic inference of sensor semantics in open systems has not been studied before. Localization systems like GPS and [LR03] typically require additional hardware and infrastructure. Mapping geometric locations, i.e., points in a coordinate system, to semantic locations, e.g., a certain room in a building, requires world models, which are inherently incomplete. Alternatively, *unknown* semantic information should be automatically inferred from known semantic information by exploiting correlations. Here, metrics to measure correlations between sensors, such as used for clustering sensors with *known* semantics [GSH$^+$08], play a key role.

With respect to content-based sensor search, the most closely related work is [ERO$^+$09]. Previous approaches to dynamic Web search, such as [IF99], are confined to a small subset of the Web and would not scale to a future Internet of Things. Sensor middleware [AHS07] and sensor search engines [WTL08] do not support content-based sensor search.

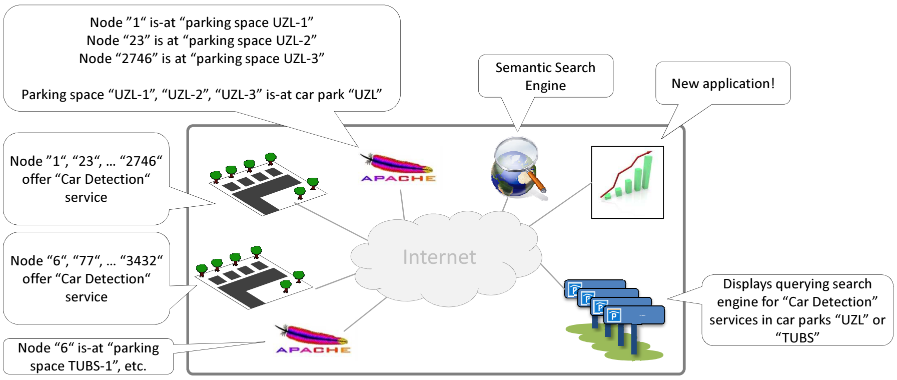
Node "1" is-at "parking space UZL-1"
Node "23" is at "parking space UZL-2"
Node "2746" is at "parking space UZL-3"
Parking space "UZL-1", "UZL-2", "UZL-3" is-at car park "UZL"
Semantic Search Engine
New application!
Node "1", "23", … "2746" offer "Car Detection" service
Node "6", "77", … "3432" offer "Car Detection" service
Node "6" is-at "parking space TUBS-1", etc.
Internet
APACHE
APACHE
Displays querying search engine for "Car Detection" services in car parks "UZL" or "TUBS"

Figure 2: Implementing the use case using semantic services

4 Semantic-Service Provisioning for the Internet of Things

Let us consider again our use case, in which we aim at providing drivers with efficient tools to find nearby, and available, parking spaces. See Figure 2 for a technical overview. First, there is a need to provide Web services to access sensor data without having to dig into low-level network access. We enable RESTful services providing sensor data description to the end user, so that standard Web APIs can be used. That way, anyone can access information about sensors using HTTP, which does not require complex programming nor engineering skills regarding sensor networks (cf. Section 4.1). Then, we need to enable data integration on top of these applications. Indeed, different applications (for instance by different manufacturers) may, internally, decide to store their data using different models or API. However, to integrate this heterogeneous data into applications and let people easily build their own applications without learning one API per parking application, data must be available, through the previous services, using common data models. This leads us to the use of Semantic Web technologies in the project, where the use of lightweight ontologies provide common data modeling for sensor data, whatever the low-level representation is (cf. Section 4.2). Finally, as sensor data exposed by these services may be raw data, we provide means to extract "semantic entities" from it. A semantic entity can be, for instance, "a parking place booked my company and available in the next 30 minutes in any parking of Dublin airport", and is thus the combination of different features and constraints associated to a group of sensors. To do so, these entities are mined by querying Web services using SPARQL, the RDF query language (cf. Section 4.3).

Combined together, we enable a complete framework for semantic integration and description of sensor data on the Internet of Things, relying only on Web standards, hence bridging the gap between the on-line and the off-line worlds. We shall now describe these different components in detail.

4.1 Service Infrastructure

Applications in the Internet of Things are typically distributed over sensors nodes (e.g., data collection, preprocessing and aggregation) and more powerful Internet nodes (e.g., complex data analysis and decision making). The car park use-case from section 2 describes such a scenario where the sensors measure and provide quite simple information (occupancy of a single parking spot) while one of the abovementioned internet nodes could act as a gateway providing more complex services like the number of free spots or other summarized information. Hence, application-level protocols are also needed that span sensor networks and the Internet to integrate the distributed application logic.

To achieve this, we employ both IPv6, based on the 6LoWPAN adaptation layer, and the aforementioned Constrained Application Protocol (CoAP) inside sensor networks, allowing them to provide light-weight RESTful Web Services. Unfortunately, the CoAP protocol is quite new and cannot be expected to become as widespread as HTTP in today's Internet. Because of that, a protocol conversion mechanism is used on the gateway to convert between HTTP and CoAP. This allows sensor nodes to consume HTTP-based RESTful Web Services in the Internet, as well as to provide HTTP-based RESTful Web Services to the Internet. The gateway is usually an embedded, yet more powerful device than a sensor node and by that, it can provide additional services (e.g., an aggregated view of the sensor networks data) or performance-enhancing features like caching of HTTP responses. The latter helps to both decrease delay and increase energy efficiency of the sensor network. Given an IPv6-enabled sensor network and HTTP based Web Service connectivity applications can easily span over IoT networks and the Internet.

4.2 Semantics

In order to enable the proposed approach, there is a need to provide means to semantically describe sensor information. This requires data models that can represent various characteristics of sensors and sensor data and making sensor description and sensor data available using these models, i.e., (1) defining ontologies for representing sensor data; and (2) providing sensor data available as Linked Data. .

Ontology for Sensors and Sensor Data. We developed an ontology to support a proper exposition of sensor self-awareness information. This ontology [LPH10] combines (1) a domain-agnostic ontology to describe sensor-related concepts; (2) an ontology to describe events and their relations; (3) an upper level ontology.

We decided to use the SSN-XG ontology [W3C10] as a domain-agnostic sensor ontology for two main reasons: (1) All basic aspects of sensors (and sensor data) are taken into consideration, and the ontology allows users to further describe them by integrating external ontologies; (2) It is aligned with DOLCE+DnS Ultralite [Gan10]. Ontology alignment with foundational ontologies ensures robustness of the ontology hierarchy structure and supports future interoperability with other ontologies. As an event description model, we choose the Event Model F [SFSS09] as: (1) it allows us to describe relations among events, i.e., Correlation, Causality, Mereologic and Interpretation, in the most detailed way, as discussed in [SFSS09]; (2) it relies exclusively on ontology design pattern; (3) it is also aligned with DOLCE+DnS Ultralite. Then, as an upper level ontology we use DOLCE+DnS Ultralite, as it embeds the Description and Situation (DnS) ontology: we consider the description of sensor context to be critical for sensor discovery. To address this issue, the DnS ontology is very useful: It allows us to describe situations, taking into account which entities are involved, their role, and the algorithm that they must satisfy with

```
@prefix : <http://example.org/carpark#> .
@prefix ex: <http://example.org/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix vso: <http://www.heppnetz.de/ontologies/vso/ns.owl#> .
#prefix dul: <http://www.loa-cnr.it/ontologies/DUL.owl#>

ex:ACME a foaf:Organization ;
 :owns ex:ACMECarOne .
 foaf:member ex:JohnDoe .
ex:JohnDoe a foaf:Person .

ex:ACMECarOne a vso:Vehicle  ;
 :occupies ex:ACMEParkingPlace2 .
ex:ACMEParkingPlace2 a :ReservablePlace , :CoveredPlace ;
 :reservedBy ex:ACME ;
 :isFree "false" .

ex:time a dul:TimeInterval ;
 dul:hasIntervalDate "2010-05-30T09:00:00";
 dul:hasIntervalDate "2010-05-30T12:00:00".
ex:ACMEParkingPlace1 a :BookablePlace ;
 :bookedBy ex:JohnDoe ;
 :bookingTime ex:time .
```

Figure 3: Model describing a parking place reservation

respect to the involving situation. In addition, the model is already aligned with the SSN-XG and the Event Model F ontologies.

Ontologies representing domain data can be associated with these sensor ontologies, to represent higher-level objects. For instance, considering our use case, we are currently developing a parking ontology. We aligned it with DOLCE+DnS Ultralite, FOAF (Friend Of A Friend [1]), and the Vehicle Sales Ontology [2]. This model allows us to describe parking places inside parking areas. In particular it allows us to distinguish between covered and uncovered, reservable and bookable parking places. A parking place can be booked by an instance of `foaf:Agent` for a specific limited range of time; while if it has been reserved, the reservation is not linked to a range of time. The model also allows specifying a target location and associating a distance from the parking place to it.

An example of the model is shown in Figure 3. It represents an organization that owns a vehicle, occupying a parking place reserved by this same organization. In addition, it also represents another parking place that has been booked by a user for a particular time interval.

Sensor Information as Linked Data. To fully exploit sensor information, representing it as Linked Data implies is essential. We follow the practices defined in [BL06], i.e.: (1) Use URIs as names for things; (2) Use HTTP URIs so that people can look up those names; (3) When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL);

[1] http://foaf-project.org

[2] http://www.heppnetz.de/ontologies/vso/ns

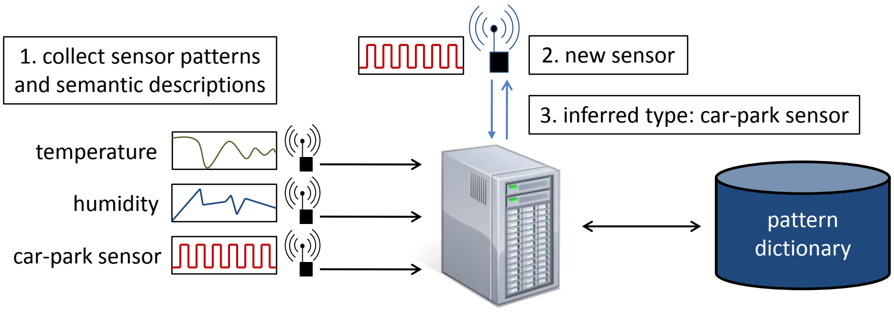
1. collect sensor patterns and semantic descriptions
2. new sensor
3. inferred type: car-park sensor
temperature
humidity
car-park sensor
pattern dictionary

Figure 4: Sensor semantic description mining

(4) Include links to other URIs so that they can discover more things. To do so, and especially to provide URIs where one can directly access the sensor, we are building RESTful Web Services described in 4.1. On access of the URI of a node or a parking place, the related information is returned in a format processable with SPARQL. In addition, the architecture supports content-negotiation, so that software can get results in different formats, e.g., plain-text or RDF, while humans can get a proper HTML page that could also be semantically-enhanced using RDFa.

Description Mining. Sensor description mining deals with automatic inference of semantic descriptions of a sensor from the semantic descriptions of other sensors to enable scalable automatic generation of descriptions and support of non-expert users.

To achieve automatic sensor annotation, we exploit correlations and statistical features of sensor output data, and domain knowledge. Given the degree of correlation of output data of two sensors, where one is already annotated and the other is not, we can conclude if the latter one could be annotated with the same description. Additionally, domain knowledge helps to increase the accuracy of the process. We know, e.g., that a certain disturbance in magnetic field suggests the presence of a car. If a sensor produces such disturbance pattern, then it is very likely that the sensor is of type AMR, and has a ”state:occupied”. We propose an architecture for our approach as depicted in Fig.4.

Since pair-wise correlation checking among sensors requires quadratic communication overhead as a function of the number of nodes in the network, we collect statistical features of sensors and their corresponding descriptions at a gateway, and store them in a database called “pattern dictionary.” Thus, the spent resources scale linearly with the size of the network.

4.3 Semantic Entities

To be able to provide useful semantic information, it is crucial to aggregate sensors into meaningful higher-level entities. These Semantic Entities can then deliver high-level information that abstract from individual sensor nodes. For example, instead of caring whether an AMR sensor measures values in a specified range, we are interested in whether a certain parking place is occupied or not. The system provides a single Semantic Entity (SE) representing the parking space and its current state. Nodes can be part of multiple of such entities at the same time. The

```
@prefix ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
SELECT ?node
WHERE {
 ?node ssn:attachedSystem ex:ACMEParkingPlace1 .
}
```

(a) Definition of the Semantic Entity `:ACMEParkingPlace1`

```
SELECT ?parkingPlace
WHERE {
 ?parkingPlace a :BookablePlace .
 ?parkingPlace :isFree "true" .
 ?parkingPlace dul:hasLocation <http://dbpedia.org/resource/Dublin> .
}
```

(b) Query for free parking spaces in Dublin

Figure 5: Exemplary definition of Semantic Entities by SPARQL and queries on them.

whole parking lot for instance could be represented by another Semantic Entity, which includes all nodes guarding individual parking spaces. We use SPARQL [W3C10] to express the semantic connections between sensor nodes that define entities. A semantic entity is defined as the set of nodes returned as the result of a SPARQL query. Since Semantic Entities have a semantic representation, their state can be queried using SPARQL. See Figure 5 for an illustration.

The realization of Semantic Entities happens either in-network (INSE), or on the service level described in Section 3 (SLSE). The former constructs entities within the sensor network, so entity queries are answered by its nodes. This form enables the nodes to optimize their behavior accordingly, e.g., nodes that are not related to the INSE can enter sleep mode to conserve energy.

For complex entities or those that are spread over a larger physical area, such an approach implies a considerable communication overhead. The system can decide to construct a SLSE instead. SLSEs are maintained by the services that create them. In this model sensor nodes stream their collected data to the service, utilizing the architecture described in Section 3. The service presents the Semantic Entity to the outside. Note that the nodes of the SLSE are not necessarily limited to act as plain data providers in this case. That can still run sleep scheduling and other activities, so a combination of INSE and SLSE properties is possible.

The set of nodes defining a SE is responsible for ensuring responsiveness at all times, deciding which of the nodes will answer a certain query and keeping the entity intact. When a new node appears, the entity decides whether or not to integrate it or if to inform other entities about a potential new member. Similarly, if a node of an entity fails or quits intentionally, the SE will either substitute it appropriately or decide to cease to exist because it cannot correspond to its definition anymore.

4.4 Content-Based Search

Content-based search tries finding a set of sensors (or SEs) with a certain state. With a growing amount of sensors it becomes infeasible to contact all sensors to read their current state due to energy and latency issues. Hence, we propose two approaches to optimize the search process.

```
@prefix sr: <http://example.org/carpark/sensor-ranking#>

SELECT ?parkingPlace
WHERE {
  ?parkingPlace dul:hasLocation <http://dbpedia.org/resource/Dublin > .
}
ORDER BY (sr:probabilityOrder(?parkingPlace, "free"))
```

Figure 6: Using custom functions to integrate Sensor Ranking into SPARQL.

Firstly, we exploit semantic annotations to minimize the search space. E.g., if sensors are annotated with location information and we want to find sensors in proximity to us, we can limit the search space to sensors in that region. Secondly, we apply the idea of *Sensor Ranking* and exploit characteristics of sensors and correlations between different sensors to construct prediction models for these sensors. [ERO+09] introduces different prediction models based on time-series forecasting. Although these models perform well in predicting states of sensors which monitor time-dependent features like the temperature over day, they might perform badly for other types of sensors like movement detection which does not follow a regular temporal pattern. However, if several motion detection sensors are placed close to each other it is very likely that their output is correlated. We will exploit those correlations to predict the values of sensors given the values of other correlated sensors.

We will integrate both types of models and choose the best-performing approach based on the scenario and type of sensor. With these models, we can forecast the current state and sort sensors by decreasing probability of matching our query. By contacting the sensors in that order to read their actual current state, we spend resources on the most promising sensors first. We will integrate the idea of Sensor Ranking in our architecture by injecting a custom function in the ORDER BY-element of SPARQL (see Figure 6). The returned list of sensors is ordered by decreasing probability of matching the query top and can be used directly to contact the sensors in this order.

5 Conclusion and Future Work

This paper has presented our ongoing work on an development methodology and platform to support the efficient and cost-effective development of applications integrating sensors and information systems using Semantic Web technologies. The goal is to take the burden of in-depth knowledge and repeated tasks off software developers by providing powerful building blocks. These can be combined in a component-oriented fashion using semantic technologies. While this is ongoing work, we have tried to present initial results in this paper and provide the development path and rationale for the final system.

Acknowledgements: This work has been partially supported by the European Union under contract number ICT-2009-258885 (SPITFIRE) and by Science Foundation Ireland under Grant No. SFI/09/CE/I1380 (Líon2).

Bibliography

[AHS07] K. Aberer, M. Hauswirth, A. Salehi. Infrastructure for data processing in large-scale interconnected sensor network. In *8th International Conference on Mobile Data Management (MDM-2007)*. 2007.

[BDH+09] J. G. Breslin, S. Decker, M. Hauswirth, G. Hynes, D. Le Phuoc, A. Passant, A. Polleres, C. Rabsch, V. Reynolds. Integrating Social Networks and Sensor Networks. In *W3C Workshop on the Future of Social Networking*. January 2009.

[BL06] T. Berners-Lee. Linked Data. Design issues for the world wide web, World Wide Web Consortium, 2006. http://www.w3.org/DesignIssues/LinkedData.html.

[BHB09] C. Bizer, T. Heath, T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web & Information Systems* 5(3):1–22, 2009.

[DAW+08] M. Durvy, J. Abeillé, P. Wetterwald, C. O'Flynn, B. Leverett, E. Gnoske, M. Vidales, G. Mulligan, N. Tsiftes, N. Finne, A. Dunkels. Making sensor networks IPv6 ready. In *SenSys*. Pp. 421–422. 2008.

[ERO+09] B. M. Elahi, K. Römer, B. Ostermaier, M. Fahrmair, W. Kellerer. Sensor ranking: A primitive for efficient content-based sensor search. In *Proceedings of the 2009 Intl. Conference on Information Processing in Sensor Networks*. Pp. 217–228. 2009.

[Fie00] R. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, 2000.

[FSA08] T. Frenken, P. Spiess, J. Anke. A Flexible and Extensible Architecture for Device-Level Service Deployment. In *ServiceWave '08: Proceedings of the 1st European Conference on Towards a Service-Based Internet*. Pp. 230–241. Springer, 2008.

[Gan10] A. Gangemi. DOLCE+DnS Ultralite. 2010. http://ontologydesignpatterns.org/wiki/Ontology:DOLCE%2BDnS_Ultralite.

[GSH+08] M. Gauger, O. Saukh, M. Handte, P. J. Marrón, A. Heydlauff, K. Rothermel. Sensor-Based Clustering for Indoor Applications. In *SECON*. Pp. 478–486. 2008.

[IF99] A. C. Ikeji, F. Fotouhi. An adaptive real-time Web search engine. In *WIDM*. Pp. 12–16. 1999.

[LPH10] M. Leggieri, A. Passant, M. Hauswirth. A Contextualised Cognitive Perspective for Linked Sensor Data. In *3rd Intl. Workshop on Semantic Sensor Network*. 2010.

[LR03] K. Langendoen, N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks* 43(4):499–518, 2003.

[M. 07] M. Eid and 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*. Pp. 59–62. 2007.

[MKHC07] G. Montenegro, N. Kushalnagar, J. Hui, D. Culler. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944 (Proposed Standard), Sept. 2007.

[OCG] OCG – Open Geospatial Consortium. The OpenGIS Sensor Model Language Encoding Standard (SensorML). http://www.opengeospatial.org/standards/sensorml.

[OCG10] OCG - Open Geospatial Consortium. Sensor Web Enablement (SWE). 2010.

[PKGZ08] N. B. Priyantha, A. Kansal, M. Goraczko, F. Zhao. Tiny Web Services: Design and Implementation on Interoperable and Evolvable Sensor Networks. In *SenSys'08: Proceedings of the 6th ACM Conf. on Embedded Network Sensor Systems*. 2008.

[RGB$^+$09] B. Rochwerger, A. Galis, D. Breitgand, E. Levy, J. Cceres, I. Llorente, Y. Wolfsthal, M. Wusthoff, S. Clayman, C. Chapman, W. Emmerich, E. Elmroth, R. Montero. Design for Future Internet Service Infrastructures. In Tselentis et al. (eds.), *Towards the Future Internet: A European Research Perspective*. Pp. 227–237. IOS Press, Amsterdam, The Netherlands, The Netherlands, 2009.

[RKT05] D. Russomanno, C. Kothari, O. Thomas. Building a Sensor Ontology: A Practical Approach Leveraging ISO and OGC Models. In *The 2005 International Conference on Artificial Intelligence*. Pp. 637–643. 2005.

[Sen] Sensinode Nanostack. http://sourceforge.net/projects/nanostack/.

[SFS10] Z. Shelby, B. Frank, D. Sturek. Constrained Application Protocol (CoAP). Online version at http://www.ietf.org/id/draft-ietf-core-coap-03.txt (10/26/2010), 2010.

[SFSS09] A. Scherp, T. Franz, C. Saathoff, S. Staab. F–a model of events based on the foundational ontology dolce+DnS ultralight. In *K-CAP '09: Proceedings of the 5th Intl. Conference on Knowledge Capture*. Pp. 137–144. ACM, New York, 2009.

[SRMA09] A. Salehi, M. Riahi, S. Michel, K. Aberer. GSN, Middleware for Streaming World. In *Proceedings of the 10th International Conference on Mobile Data Management (MDM 2009)*. May 2009.

[TH02] P. M. Tolani, J. R. Haritsa. XGRIND: A Query-Friendly XML Compressor. In *Proceedings of the 18th International Conference on Data Engineering, 26 February – 1 March 2002, San Jose, CA*. Pp. 225–234. IEEE Computer Society, 2002.

[W3C10] W3C Semantic Sensor Network members. W3C Semantic Sensor Network Incubator Group. 2010. http://www.w3.org/2005/Incubator/ssn/.

[WBJ05] C. Werner, C. Buschmann, T. Jäcker. Enhanced Transport Bindings for Efficient SOAP Messaging. *IEEE Intl. Conference on Web Services*, pp. 193–200, 2005.

[Wor] World Wide Web Consortium (W3C). Efficient XML Interchange (EXI) Format 1.0. W3C Working Draft.

[WTL08] H. Wang, C. C. Tan, Q. Li. Snoogle: A Search Engine for the Physical World. In *Infocom*. 2008.