Semantic-enhanced resource discovery for CoAP-based sensor networks

Filippo Gramegna, Saverio Ieva, Giuseppe Loseto, Agnese Pinto
DEI - Politecnico di Bari
via Re David 200, I-70125, Bari, Italy
{gramegna, ieva, loseto}@deemail.poliba.it, agnese.pinto@poliba.it

***Abstract*—The integration of knowledge representation and reasoning techniques (originally devised for the Semantic Web) in most common Wireless Sensor Networks (WSNs) protocols can allow to reach higher levels of autonomicity w.r.t. classic network architectures that basically provide only simplistic discovery capabilities. This paper presents a complete Semantic Sensor Network (SSN) framework, supporting a resource discovery based on non-standard inferences. A backward-compatible extension of Constrained Application Protocol (CoAP) has been proposed to support semantic matchmaking for retrieving and ranking resources annotated w.r.t. a reference ontology. Data mining procedures were also exploited to detect high-level events from gathered raw data. A case study on environmental monitoring has been proposed to test the effectiveness of our approach.**

Keywords*—*Semantic Sensor Networks, CoAP, Resource discovery, Matchmaking, Data mining

I. Introduction

In latest years, *Wireless Sensor Networks (WSNs)* are assuming a more and more relevant role in the field of sensor communication. Such infrastructures integrate distributed heterogeneous resources as sensors, actuators, network devices exploiting common interfaces and general purpose communication protocols to monitor physical or environmental conditions. Unfortunately, WSNs are usually application-dependent and engineering-oriented [1] presenting several limits in terms of interoperability particularly in case of large-scale complex architectures. Semantic technologies were acknowledged as a mean to overcome these issues. *Semantic Sensor Networks (SSNs)* refer to WSNs where component resources, sensor data and available services are annotated following a semantic model. In particular, Knowledge Representation formalisms allow to use metadata to describe all the relevant processes and actors of such architectures in a machine understandable format. This facilitates automatic discovery and integration of resources as well as advanced tasks involving different nodes such as device management [2], query processing [3], application-level services [4]. Moreover, many research studies have shifted toward the integration of SSNs with the World Wide Web, following the *Semantic Web of Things (SWoT)* vision [5]. Its goal is to associate semantically rich and easily accessible information to real-world objects, locations and events, by means of inexpensive, disposable and unobtrusive micro-devices. Semantics is devoted to increase both autonomicity and integrability, in order to truly fulfill the potential of the SSN paradigm through enhanced device and event annotation enabling logic-based applications.

Different alternatives aiming to define standard protocols in the field of object networks have been proposed, in particular 6LoWPAN [6] and the Constrained Application Protocol (CoAP) [7]. Nevertheless, current solutions only allow simplistic data-oriented representation and basic discovery procedures based on trivial syntactic comparisons, only providing binary yes/no outcomes. Full request/resource matches are very uncommon in real-world scenarios, particularly in case of sensor networks. A more efficient resource discovery should also take into account partial correspondences, possibly providing a measure of the semantic similarity degree between a request and the available devices. In this paper we draw approach and formalisms from the Semantic Web and Internet of Things initiatives and adapt them to SSNs. The proposed SSN architecture enables to: (i) describe data streams, sensors and actuator devices by means of semantic-based annotations expressed w.r.t. a shared domain conceptualization (*i.e.*, ontology); (ii) exploit non-standard inference services for semantic-based matchmaking to rank [8], compose [9] and retrieve resources best matching a given request, supporting not only full matches but also approximate ones; (iii) detect and annotate high-level events from raw data collected from sensing devices in the field using a simple data mining component. The proposal is based on slight backward-compatible extensions to CoAP[1] and CoRE Link Format[2] resource discovery protocol for sensor and actor networks whereas the SSN-XG ontology of W3C (World Wide Web Consortium) [10] is adopted as reference vocabulary for resource annotations. A plugin for the open source Java OpenStreeMap (JOSM) application was also developed to support sensor discovery through a user-friendly interface.

The remainder of the paper is organized as follows. Section II surveys most relevant related research. Section III introduces basic notions on the CoAP protocol and outlines the proposed semantic-based enhancements. In Section IV the discovery framework is presented, also providing details about reasoning algorithms and event mining. In Section V an illustrative case study clarifies the main features of the proposed approach. Final remarks are in Section VI.

II. Related work

CoAP [7] is an alternative protocol to HTTP for interconnected objects, exploiting a binary data representation and a

[1] Constrained Application Protocol, IETF CoRE Working Group Internet-Draft, version: 13, 6 December 2012, http://tools.ietf.org/id/draft-ietf-core-coap-13.txt

[2] CoRE Link Format, IETF CoRE Working Group Internet-Draft, latest version: 14, 1 June 2012, http://www.ietf.org/id/draft-ietf-core-link-format-14.txt

subset of HTTP methods. It follows the REST (REpresentational State Transfer) paradigm for making data and resources accessible. 6LoWPAN [6] is another protocol for WSNs defined to enable IPv6 packets to be carried on top of low power wireless networks. Particularly CoAP is assuming relevance for its lightweight impact on storage and computation, resulting useful for a variety of application domains [11], [12].
In latest years, interesting SSN approaches were developed to integrate WSNs and smart objects with the Semantic Web [13]. Existing architectures largely vary in scope but usually aim to: (i) exploit reference ontologies –*e.g.*, OntoSensor [14] and SSN-XG [10]– to annotate data, devices and services; (ii) share sensor data along the Linked Open Data (LOD) [15] guidelines and by means of RESTful interfaces [16]. Particularly, *SPITFIRE* [17] proposes a service infrastructure to develop semantic applications exploiting Internet-connected sensors and lightweight protocols, *e.g.*, CoAP. In such framework, sensors are described as RDF[3] triples and service discovery is based on meta-data such as device features or location. Furthermore, in [18] the *Linked Stream Middleware* (LSM) platform was proposed to integrate sensor data with other LOD sources. A processing engine was used to perform SPARQL[4] queries across sensor dataset, mashup the data and process results. The sensor ontologies were also used to provide sensor composition capabilities to SSN frameworks. Tran *et al.* [19] describe an application of semantics to automatically create sensor compositions. In particular user goals, functional and non-functional properties of sensors are described w.r.t. an OWL[5] ontology so that the composition system is able to combine sensors and processes to satisfy a user request. In [20] ontology-based sensor descriptions allow the users to express requests in terms of device characteristics. Quantitative reasoning and semantic querying techniques were employed to improve the resource discovery and select appropriate sensors. Unfortunately, the above solutions only allow elementary queries in SPARQL fragments on RDF annotations and basic resource discovery features. Ontology-based *complex event processing* [21] and *semantic matchmaking* [8] can be used to improve data management and sensor discovery in mobile and pervasive contexts [22], [23] exploiting logic-based reasoning to support approximated matches, resource ranking and explanation of outcomes.

III. Semantic enhancements to CoAP protocol

Basically, a CoAP message is composed of: (i) a 32-bit *header*, containing the request method code (or response status); (ii) an optional *token* value, used to match responses to requests, (iii) a sequence of *option* fields, which carries information such as resource URI and payload media type, (iv) *payload* data. CoRE Link Format specification is adopted for resource discovery, allowing any host to expose its resources. A client will access the reserved URI path `/.well-known/core` on the server with `GET` to discover available ones. `GET` requests can include `URI-query` fields to retrieve only resources with specific attributes. Standardized query attributes include resource type (`rt`), interface usage (`if`), content-type (`ct`), and MIME (Multipurpose Internet Mail Extensions) type/subtype (`type`) for a resource. Further non-reserved attributes can be freely used. CoAP also provides push notifications without the need of polling[6]; these features is particularly useful in scenarios where data have to be monitored over a time span. In CoAP-based scenarios, each sensor is seen as a server, exposing both sensor readings and internal information as resources toward clients, which act on behalf of end-user applications. Furthermore, since CoAP supports proxies, cluster-head or sink nodes can reply on behalf of a set of (possibly more constrained) sensor nodes deployed in an area, exploiting caching and decreasing the load at the edge of the network. This feature allows also the adoption of data fusion and mining techniques at various levels along the path from sensors in the field to nodes managing high-level application logic.

In order to support semantic-based resource discovery, the CoAP protocol has been improved with an innovative usage of standard `URI-query` options and on the addition of new ones. The CoAP-based SSN framework proposed here extends the enhancements described in [24] integrating novel non-standard logic-based inferences to support approximate match calculation and automated sensor discovery and composition. However the resulting framework is still fully backward compatible: servers which do not support semantics will simply reply to requests returning no resource records.
In particular, **annotated request** is embedded into standard CoAP request by introducing three novel attributes: (i) reference ontology (`ro`), containing the URI of the reference ontology; (ii) semantic description (`sd`), maintaining a request annotation, compressed to cope with the verbosity of XML-based languages; (iii) annotation-type (`at`), specifying the compression format. Reference **geographical location** is achieved by specifying `lg` (longitude) and `lt` (latitude) attributes. Furthermore, `md` (maximum distance) is used to indicate the maximum acceptable distance (in meters) from the reference location. The adoption of a (center,distance) constraint allows the server to pre-filter resources, so avoiding the relatively expensive inference procedures for resources outside the requested area. Finally, each provided **reasoning task** enhancing the resource discovery is identified by a numeric code, specified by `st` (semantic task) attribute. In addition to existing tasks, a new code for sensor discovery based on Concept Covering [9] has been added. Furthermore, semantic threshold (`sr`) attribute is used in discovery requests to specify a minimum score threshold. Resources having an overall score w.r.t. the request lower than this threshold will not be returned to the client. This allows to modulate the granularity of discovery and to limit data transfers when many resources are available. In replies to discovery, this field contains the overall score of a resource w.r.t. the request.

IV. Framework Architecture

In what follows the approach we present will be thoroughly described, particularly detailing the semantic matchmaking

[3]Resource Description Framework, W3C Recommendation, 10 February 2004, http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

[4]SPARQL Query Language for RDF, W3C Recommendation 15 January 2008, http://www.w3.org/TR/rdf-sparql-query/

[5]Web Ontology Language, W3C Recommendation, 11 December 2012, http://www.w3.org/TR/owl-overview/

[6]Documented in: Observing Resources in CoAP, IETF CoRE Working Group Internet-Draft, latest version: 8, 25 February 2013, http://tools.ietf.org/id/draft-ietf-core-observe-08.txt

framework adopted for resource discovery and the event mining process.

A. *Resource discovery via concept covering*

CoAP resource discovery protocol only allows a syntactic string-matching of attributes, lacking every explicit and formal characterization of the resources semantics. In [8], framework and algorithms were proposed for a logic-based matchmaking between a request and one or more resource descriptions, both expressed using languages grounded on a well known logic. Also a ranking of resource annotations w.r.t. the original request was made possible according to the meaning of descriptions with reference to a shared conceptualization, *i.e.*, an ontology. Description Logics (DL) [25] was the reference formalism and particularly the $\mathcal{ALN}$ (Attributive Language with Unqualified Number Restrictions) DL subset was used, which has polynomial computational complexity for standard and non-standard inferences. Suck kind of services can be used to identify also partial correspondences –very frequent in practical scenarios involving heterogeneous resources– between a request and device descriptions. In [24], *Concept Abduction Problem* (CAP) [26] inference was exploited to determine, given a request D and a resource S, what should be hypothesized in S in order to completely satisfy D also enabling a logic-based relevance ranking of a resource w.r.t. a given request. In addition to non-standard inference services proposed in [24], *Concept Covering* [9] service has been introduced to select the minimum set of resources best covering a request. Given a request D and a set of resources $\mathtt{S} = \{\mathtt{S}_1, \mathtt{S}_2, ..., \mathtt{S}_k\}$, where D and $\mathtt{S}_1$, $\mathtt{S}_2$, ..., $\mathtt{S}_k$ are satisfiable in $\mathcal{T}$, the Concept Covering Problem (CCoP) aims to find a pair $\langle \mathtt{S}_c, H \rangle$ where $\mathtt{S}_c$ includes concepts in $\mathtt{S}$ (partially) covering D w.r.t. $\mathcal{T}$ and H is the (possible) part of D not covered by concepts in $\mathtt{S}_c$. Such non-standard inference service is particularly useful in sensor network scenarios, where we need to gather data from different type of sensor with specific features to infer proper events. Moreover, it could be necessary to automatically and dynamically substitute no longer available sensor nodes. In this case, the status of registered nodes will be periodically monitored and, if any is down, CCoP can be used to replace disabled ones with most suitable available sensors.

An example will clarify the structure and content of request and reply messages in the semantic-enhanced variant of CoAP protocol. Semantic annotations will be voluntarily omitted here for the sake of clarity.

Discovery request. An application queries a SSN sink having 193.204.59.75 IP address to find a set of sensors most suitable for a semantic description expressed in OWL/RDF language w.r.t. SSN-XG ontology and compressed with gzip. The application is interested only in sensors located within 800m from the location at $(41.079769, 16.763571)$ coordinates. The application will therefore send a `GET` CoAP request to:

```
coap://193.204.59.75:5683/.well-known/core?
&ro=SSN-XG-IRI&sd=yyyyyy=&at=30004&lg=16.763571
&lt=41.079769&md=800&st=2&sr=70
```

Discovery reply. Upon receiving the request, the CoAP server will start a matchmaking process comparing it w.r.t. all stored annotations referred to the same ontology. The semantic matchmaking is carried out by solving CCoP, in order to find the set of resources best covering the request and what elements of the request are lacking in the retrieved sensor list. Let us suppose that the returned set is composed of two sensors matching the above request. The CoAP server response payload will be:

```
</Hts2030HumidSens>;ct=0;ct=41;at=30004;lg=16.768277;
lt=41.077286;md=480;ro=SSN-XG-IRI;sd=aaaaaaa;
title="Humidity-Sensor-2030",
</BitLineAnemomSens>;ct=0;ct=41;at=30004;lg=16.758347;
lt=41.081983;md=500;ro=SSN-XG-IRI;sd=bbbbbbb;
title="Anemometer-Sensor-111"
```

B. *Event annotation*

The proposed framework includes a resource-effective statistical data mining process. It aims at a semantic-based event annotation starting from low-level data audit. Recall that a SSN produces large amounts of data which have to be collected and interpreted to extract application-oriented information –this is particularly relevant in case of event detection where (semi) automatic procedures are strongly required. In particular, each sink device in the SSN collects data from sensors in the field and analyzes them by exploiting a threshold classifier for associating a semantic description. Whenever an event is detected, a dedicated and general-purpose CoAP resource record is updated by adding the obtained annotation. The record will also contain extra-logical context parameters such as a timestamp and geographic information about the monitored area useful to refine the further matchmaking. Additional parameters can also be added according to the specific post-processing task. In fact, after detection, the sink will serve as a CoAP gateway, waiting for resource discovery requests coming from client applications searching for specific events in the area.

The procedure for identification of sensory events via mining carries along several stages:

– Data are read from sensors in the field through standard CoAP GET requests, possibly using `Observe` option to be notified of updates. Then a list of elements is built, consisting of a minimum set of three fields: *ID*, storing the identifier of the sensor and the related produced data type; *value*, where the detected data is stored; *timestamp*. This list will group measurements in time slots of application-defined period T, which are used to compute statistical indexes.

– For each data set, average, variance and standard deviation values are computed for the current time slot, to assess the variability of collected information within the monitored area and to reduce the contribution of possible noisy data collected by the sensors.

– Statistical indexes of elapsed periods are then exploited to compute an incremental ratio able to evidence trends and significant event changes inside the monitored area.

– For every data collection the application defines a binary or multiple classifier, to reveal a situation when given conditions occur. Each event annotation is associated with the parameter range that contributes to the occurrence of the event itself. The data collected from sensors are examined and whenever their value or combination fall into pre-defined range, it is possible to identify the event.

– The output of each classifier is a logic-based expression, mapped according to knowledge modeled in the reference ontology.

It is important to note that semantic-enhanced CoAP discovery *per se* does not impose restrictions on where data

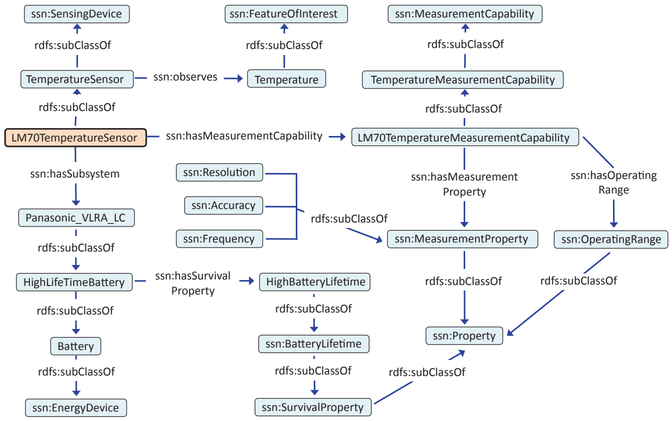

Fig. 1. Temperature sensor modeling

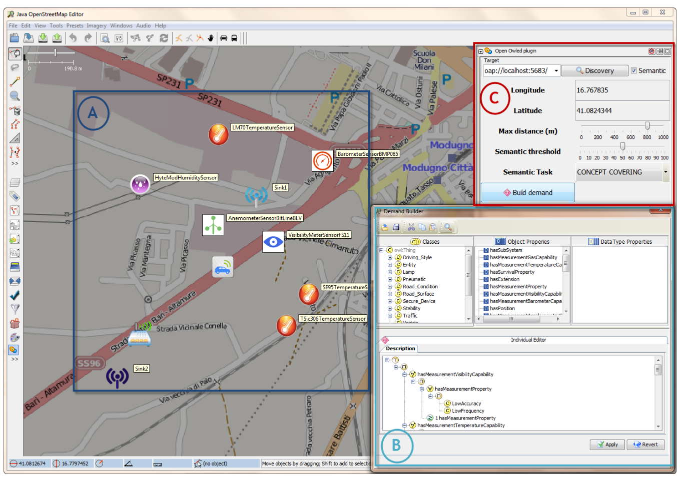

Fig. 2. JOSM plugin for CoAP-based SSNs

mining happens, whether in clients running application logic, in sink nodes or in sensors having processing capabilities enough. In the approach proposed here, mining and event detection are executed at sink level. In fact, sensors are able to send sensed data only via standard CoAP frames. Hence, the sink will take care to: (i) collect data from sensors; (ii) apply mining algorithms; (iii) annotate event descriptions w.r.t. the reference ontology and make them available for external client applications.

V. Environmental monitoring with CoAP: a case study

In what follows a simple case study for environmental monitoring is reported with the aim to show the benefits of the proposed framework. The scenario is configured as a Vehicular Ad-Hoc Network (VANET), in which vehicles receive useful information from Road-Side Units (RSUs). CoAP protocol enhancements and concept covering are used to enable an advanced sensor discovery in order to annotate environmental events detected from sensor data. Starting from event annotations and vehicle descriptions, RSUs can exploit a matchmaking process to send safety warnings to vehicles. The VANET is grounded on a SSN composed of several CoAP sink nodes which aggregate different types of sensors compliant with CoAP. Each RSU acts as a *first-level* sink and represents the interface of the network toward external applications.

Periodically, each RSU queries the CoAP sinks (or directly sensors) in its range. Before gathering raw data, the sinks select sensors by means of a logic-based resource discovery and return to the RSU the most appropriate device set. It is important to notice that this task is computationally lightweight –thanks to the optimized algorithms provided by the mobile matchmaker [27] embedded on each sink– and independent from the amount of available raw data. The RSU can now query the sensors to obtain collected data and exploits data mining, as described in Section IV-B, to detect a meteorological event. Each sensor is able to measure a proper parameter (*e.g.*, temperature, humidity, atmospheric pressure, wind speed), with specific measurement capabilities (*e.g.*, accuracy, precision, resolution, frequency). Sensor descriptions are represented by conjunctive concept expressions referring to the same ontology, which extends SSN-XG [10]. Figure 1 shows an example of temperature sensor modeling. We exploited the pattern defined in [10] to describe the measuring features of a sensor, with some differences. A sensor can "observe" properties modeled as subclasses of `ssn:FeatureOfInterest` and has proper measurement capabilities expressed as subclasses of the `ssn:MeasurementCapability` class. In turn, each specific subclass of `ssn:MeasurementCapability` has a set of measurement and (optional) operating properties, represented as subclasses of the `ssn:Property` class. Furthermore, a sensor is related to a subclass of `ssn:EnergyDevice` through the `ssn:hasSubSystem` property to model its energy source.

Particularly, we developed a prototypical software tool for sensor discovery through a fully visual user interface. A plugin for the open source JOSM[7] application was devised supporting the CoAP-based framework described above. Communication with sensors has been implemented using a modified version of *Californium CoAP library* [28], extended with the semantic-based enhancement proposed in Section III. A screenshot of the prototype GUI is shown in Figure 2. Thanks to an easily understandable interface, the proposed tool can be used to perform the following tasks in a graphical way:
– *SSN browsing*. A geographic area of interest can be downloaded from OSM server. Available sensors, registered on CoAP sinks, can be showed on the map on the left panel (A);
– *Semantic-based sensor discovery*. A semantic-based CoAP request can be send to discover sensors in the environment. The right panel (C) allows to customize the request with semantic attributes described in Section III. Particularly, a reference location point can be selected by clicking on the map. Then it is possible to choose the semantic task to perform (*e.g.*, discovery via Concept Covering), the semantic threshold and the maximum discovery range. Finally, the semantic description can be composed by simply selecting (with drag-and-drop operations) from the UI panel (B) properties and classes defined in the reference ontology.

Let us suppose *a car is driving on the SS96, a highway near Bari, in the morning. The road presents a low-flowing with a poor traffic density (90 vehicles per hour). Possible dangers are due to several crossroads*. As shown in Figure 3, the car is near *RSU1*. RSU1 sends a discovery request, using concepts defined in the domain ontology, as described in Table

[7]Java OpenStreetMap (OSM) editor, http://josm.openstreetmap.de/

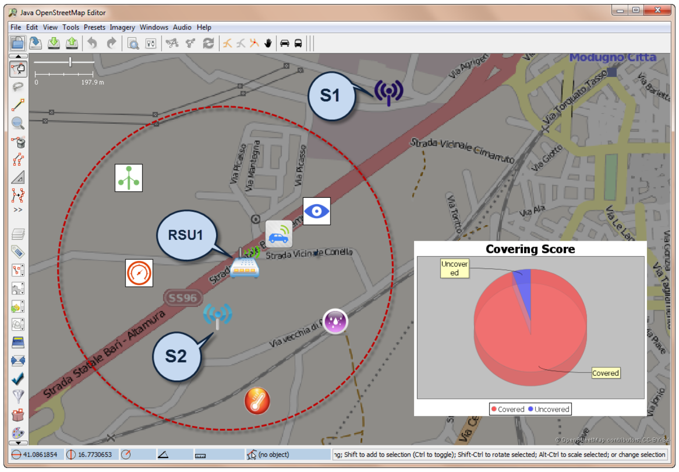

Fig. 3. Covering result

Parameter	Meteorological event			
	Fog	Wind	Rain	Snow
temp 0 m (°C)	-	-	$\geq$6	-
temp 0÷599 m (°C)	$t - t_d \leq 4$	-	-	$\leq$5
temp 600÷1499 m (°C)	-	-	-	5÷10
temp $\geq$1500 m (°C)	-	-	-	$\leq$0
visibility(m)	$\leq$1000	-	-	-
wind speed (km/h)	-	$\geq$100	-	-
humidity (%)	-	-	80÷100	-
pressure (mbar)	-	-	970÷1000	-

t_d: dew point temperature

TABLE II. CRITERIA FOR METEOROLOGICAL EVENT DETECTION

I[8]. The GET request also includes three query parameters: (i) RSU reference location P, in terms of geographic coordinates; (ii) the *maximum distance* d from P, defining the area in which sensors must be located; (iii) the *minimum threshold* t, to retrieve only device sets with a covering percentage score higher than t. In this example, RSU1 look for a set of devices positioned near SS96 with a maximum distance of **800m** from the reference point P and a coverage threshold value of **90%**. After a distance-based filtering, each sink node near RSU1 –*i.e.*, *S1* and *S2*– solves the CCoP exploiting the embedded reasoning engine [27]. The concept expressions for some of the sensor instances inside the measurement area, showed in Figure 3, are reported in Table I. Sink node (*S2*) returns the set of compatible resources S_c whose intersection best covering the request D. According to Table I, S_c is composed of LM70Sensor, Hts2030Sensor, BMP085Sensor, BitLineBLVSensor and FS11Sensor. Nevertheless, as reported by the pie chart in Figure 3, this set does not fully cover the request so an uncovered part U is returned. In particular, LM70Sensor does not completely satisfy the required features in terms of temperature measurement capabilities.

Now RSU1 can query/observe the retrieved sensors registered to *S2*. For example, let us suppose that, after a period of observation, the mining process produces the following mean values for the parameters gathered by the sink: sea level temperature: 7.09°C; temperature between 0÷599 m: 1.98°C; relative humidity: 80.52%; wind speed: 19.69 km/h; atmospheric pressure: 971.51 mbar; visibility: 254.38 m. Based on studies and laws on meteorological event detection, we designed a classifier able to detect one of the weather conditions reported in Table II. In the example, the classifier identifies *Fog* and *Rain* events due to high values of humidity and low values of pressure. Each detected event is annotated w.r.t. the reference ontology as subclass of the *Weather* concept. This high level semantic-based characterization of the weather conditions becomes a query for further matchmaking processes based on the environmental conditions. In the proposed case study, RSU1 waits for vehicles equipped with a wireless interface that come into its radio range. Each vehicle will send a semantic description containing main characteristics of its safety devices. Suppose that two different vehicles, described in DL formalism as follows, drive nearby RSU1:

BMWX5 $\equiv Vehicle \sqcap SUV \sqcap \exists hasSpeed \sqcap \forall hasSpeed.(ModerateSpeed) \sqcap \exists hasLamp \sqcap \forall hasLamp.(XenonLamp \sqcap FogLamp) \sqcap \exists hasSecureDevice \sqcap \forall hasSecureDevice.(ABS \sqcap ESP) \sqcap \exists hasPneumatic \sqcap \forall hasPneumatic.(SnowTire)$.

Fiat600 $\equiv Vehicle \sqcap EconomyCar \sqcap \exists hasSpeed \sqcap \forall hasSpeed.(HighSpeed) \sqcap \exists hasLamp \sqcap \forall hasLamp.(Headlight) \sqcap \exists hasPneumatic \sqcap \forall hasPneumatic.(TraditionalTire)$.

RSU1 will perform a matchmaking task between a vehicle description and events previously detected, described in terms of safety requirements a car must adopt to limit risk level for that particular weather conditions:

Fog $\equiv Weather \sqcap \exists hasSpeed \sqcap \forall hasSpeed.(ModerateSpeed) \sqcap \exists hasLamp \sqcap \forall hasLamp.(FogLamp) \sqcap \exists hasSecureDevice \sqcap \forall hasSecureDevice.(ABS)$.

Rain $\equiv Weather \sqcap \exists hasSpeed \sqcap \forall hasSpeed.(ModerateSpeed) \sqcap \exists hasPneumatic \sqcap \forall hasPneumatic.(RibbedTire) \sqcap \exists hasSecureDevice \sqcap \forall hasSecureDevice.(ABS \sqcap ESP)$.

For each event, RSU1 applies the rankPotential [26] algorithm and returns a rank value (*RV*) representing the risk level for that vehicle: very low ($0 \leq RV \leq 1$), low ($RV = 2$), medium ($RV = 3$), high ($4 \leq RV \leq 5$), very high ($RV = 6$), ultra high ($RV \geq 7$). The BMW is the vehicle with the lowest risk level ($RV_{Fog} = 1$ and $RV_{Rain} = 1$), because it is equipped with snow tier (also useful in case of rain), fog lamp, ABS and ESP. The Fiat 600 is absolutely unsuitable ($RV_{Fog} = 6$ and $RV_{Rain} = 7$), due to its high speed and inadequate safety features.

VI. CONCLUSION AND FUTURE WORK

The paper proposes a complete SSN framework, supporting an advanced resource discovery exploiting a backward-compatible CoAP extension and logic-based matchmaking of semantically rich sensor and event annotations. Future work includes the extension of underlying logic toward $\mathcal{EL}/\mathcal{EL}^{++}$ for increasing allowed expressiveness of resource and request descriptions. Moreover, the proposed framework will be implemented on a large testbed with off-the-shelf devices to fully evaluate the approach in terms of computational cost and efficiency.

ACKNOWLEDGMENTS

The authors acknowledge partial support of MIUR projects RES NOVAE and BEfree@campus.

REFERENCES

[1] L. Ni, Y. Zhu, J. Ma, M. Li, Q. Luo, Y. Liu, S. Cheung, and Q. Yang, *Semantic Sensor Net: An Extensible Framework*, ser. Lecture Notes in Computer Science. Springer, 2005, vol. 3619, pp. 1144–1153.

[8]For the sake of readability, concept expressions for both request and sensors are summarized in textual form.

	Device Measurement Capabilities						
Description	Device Type	Visibility	Temperature	Operative Range	Pressure	Wind Speed	Humidity
Request	Sensor	LowAcc. LowFreq.	LowRes., LowAcc. HighLatency	LowMedium Altitude	LowAcc., MediumRes.	MediumRes. MediumAcc., LowPrec.	MediumAcc., MediumRes. HighFreq.
SE95Sensor	Temperature Sensor	-	HighAcc., HighFreq. HighRange, HighRes.	MediumHigh Altitude	-	-	-
LM70Sensor	Temperature Sensor	-	LowAcc., MediumFreq. HighRange	LowMedium Altitude	-	-	-
Hts2030Sensor	Humidity Sensor	-	-	-	-	-	MediumAcc., HighFreq. MediumRes., HighRange
BitLineBLVSensor	Anemometer Sensor	-	-	-	-	MediumAcc., LowRes. MediumRes., LowPrec.	-
BMP085Sensor	Barometer Sensor	-	-	-	LowAcc., MediumRes. LowRange, LowPrec.	-	-
FS11Sensor	Visibility Sensor	LowAcc. LowRange LowFreq.	-	-	-	-	-

TABLE I. REQUEST AND SENSORS DESCRIPTION

[2] S. Avancha, C. Patel, and A. Joshi, “Ontology-driven adaptive sensor networks,” in *First Annual International Conference on Mobile and Ubiquitous Systems, Networking and Services*, 2004, pp. 194–202.

[3] M. Compton, H. Neuhaus, K. Taylor, and K. Tran, “Reasoning about sensors and compositions,” *Proc. Semantic Sensor Networks*, p. 33, 2009.

[4] G. Jiang, W. Chung, and G. Cybenko, “Semantic agent technologies for tactical sensor networks,” in *2003 SPIE Conference on AeroSense*, 2003, pp. 21–25.

[5] M. Ruta, F. Scioscia, and E. Di Sciascio, “Enabling the Semantic Web of Things: framework and architecture,” in *Sixth IEEE International Conference on Semantic Computing (ICSC 2012)*, IEEE. IEEE, sep 2012, pp. 345–347.

[6] N. Kushalnagar, G. Montenegro, and C. P. Schumacher, “IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals,” RFC Editor, Fremont, CA, USA, RFC 4919, Aug. 2007.

[7] C. Bormann, A. Castellani, and Z. Shelby, “Coap: An application protocol for billions of tiny internet nodes,” *Internet Computing, IEEE*, vol. 16, no. 2, pp. 62–67, 2012.

[8] S. Colucci, T. Di Noia, A. Pinto, A. Ragone, M. Ruta, and E. Tinelli, “A Non-Monotonic Approach to Semantic Matchmaking and Request Refinement in E-Marketplaces,” *International Journal of Electronic Commerce*, vol. 12, no. 2, pp. 127–154, 2007.

[9] A. Ragone, T. Di Noia, E. Di Sciascio, F. M. Donini, S. Colucci, and F. Colasuonno, “Fully Automated Web Services Discovery and Composition through Concept Covering and Concept Abduction,” *International Journal of Web Services Research (JWSR)*, vol. 4, no. 3, pp. 85–112, 2007.

[10] M. Compton, P. Barnaghi, L. Bermudez, R. Garcia-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog *et al.*, “The SSN ontology of the W3C semantic sensor network incubator group,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 17, pp. 25–32, Dec. 2012.

[11] W. Colitti, K. Steenhaut, N. De Caro, B. Buta, and V. Dobrota, “REST Enabled Wireless Sensor Networks for Seamless Integration with Web Applications,” in *8th IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS 2011)*, 2011, pp. 867–872.

[12] A. Dimitrios, G. Vasileios, G. Dimitrios, and C. Ioannis, “Employing Internet of Things technologies for building automation,” in *17th IEEE Conference on Emerging Technologies & Factory Automation (ETFA 2012)*, 2012, pp. 1–8.

[13] T. Berners-Lee, J. Hendler, and O. Lassila, “The semantic Web,” *Scientific American*, vol. 284, no. 5, pp. 28–37, 2001.

[14] D. J. Russomanno, C. R. Kothari, and O. A. Thomas, “Building a Sensor Ontology: A Practical Approach Leveraging ISO and OGC Models,” in *The 2005 International Conference on Artificial Intelligence*, 2005, pp. 637–643.

[15] C. Bizer, T. Heath, and T. Berners-Lee, “Linked data-the story so far,” *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 5, no. 3, pp. 1–22, 2009.

[16] H. Patni, C. Henson, and A. Sheth, “Linked Sensor Data,” in *Collaborative Technologies and Systems (CTS), 2010 International Symposium on*. IEEE, 2010, pp. 362–370.

[17] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, M. Pagel, M. Hauswirth, M. Karnstedt *et al.*, “SPITFIRE: Toward a Semantic Web of Things,” *Communications Magazine, IEEE*, vol. 49, no. 11, pp. 40–48, 2011.

[18] D. Le-Phuoc, H. Q. Nguyen-Mau, J. X. Parreira, and M. Hauswirth, “A middleware framework for scalable management of linked streams,” *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 16, pp. 42–51, Nov. 2012.

[19] K.-N. Tran, M. Compton, and R. G. Jemma Wu, “Semantic Sensor Composition,” in *3rd International Workshop on Semantic Sensor Networks. Proceedings of the 9th International Semantic Web Conference (ISWC 2010)*, ser. CEUR Workshop Proceedings, D. R. D. Taylor K., Ayyagari A., Ed., vol. 668. CEUR-WS, nov 2010, pp. 33–48.

[20] C. Perera, A. Zaslavsky, P. Christen, M. Compton, and D. Georgakopoulos, “Context-aware Sensor Search, Selection and Ranking Model for Internet of Things Middleware,” in *IEEE 14th International Conference on Mobile Data Management (MDM 2013)*, jun 2013, to appear.

[21] K. Taylor and L. Leidinger, “Ontology-driven complex event processing in heterogeneous sensor networks,” *The Semanic Web: Research and Applications*, pp. 285–299, 2011.

[22] M. Ruta, T. Di Noia, E. Di Sciascio, and F. Donini, “Semantic-Enhanced Bluetooth Discovery Protocol for M-Commerce Applications,” *International Journal of Web and Grid Services*, vol. 2, no. 4, pp. 424–452, 2006.

[23] M. Ruta, F. Scioscia, T. Di Noia, and E. Di Sciascio, “A hybrid ZigBee/Bluetooth approach to mobile semantic grids,” *Computer Systems Science and Engineering*, vol. 25, no. 3, pp. 235–249, May 2010.

[24] M. Ruta, F. Scioscia, G. Loseto, F. Gramegna, A. Pinto, S. Ieva, and E. Di Sciascio, “A logic-based CoAP extension for resource discovery in semantic sensor networks,” in *5th International Workshop on Semantic Sensor Networks (SSN 2012)*, ser. CEUR Workshop Proceedings, C. O. Henson C., Taylor K., Ed., vol. 904. CEUR-WS, nov 2012, pp. 17–32.

[25] F. Baader, D. Calvanese, D. Mc Guinness, D. Nardi, and P. Patel-Schneider, *The Description Logic Handbook*. Cambridge University Press, 2002.

[26] M. Ruta, E. Di Sciascio, and F. Scioscia, “Concept Abduction and Contraction in Semantic-based P2P Environments,” *Web Intelligence and Agent Systems*, vol. 9, no. 3, pp. 179–207, 2011.

[27] M. Ruta, F. Scioscia, E. Di Sciascio, F. Gramegna, and G. Loseto, “Mini-ME: the Mini Matchmaking Engine,” in *OWL Reasoner Evaluation Workshop (ORE 2012)*, ser. CEUR Workshop Proceedings, I. Horrocks, M. Yatskevich, and E. Jimenez-Ruiz, Eds., vol. 858. CEUR-WS, 2012, pp. 52–63.

[28] M. Kovatsch, S. Mayer, and B. Ostermaier, “Moving Application Logic from the Firmware to the Cloud: Towards the Thin Server Architecture for the Internet of Things,” in *Proceedings of the 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS 2012)*, Jul. 2012.