Proceedings of the Federated Conference on
Computer Science and Information Systems pp. 949–955
ISBN 978-83-60810-22-4

Service Modelling for the Internet of Things

Suparna De, Payam Barnaghi
Centre for Communication Systems Research, University of Surrey, Guildford GU2 7XH. United Kingdom
Email: {S.De, P.Barnaghi}@surrey.ac uk

Martin Bauer
NEC Laboratories Europe, Software and Services Research Division, Kurfursten-Anlage 36, D-69115 Heidelberg, Germany
Email: Martin.Bauer@neclab.eu

Stefan Meissner
Centre for Communication Systems Research, University of Surrey, Guildford GU2 7XH. United Kingdom
Email: S.Meissner@surrey.ac.uk

Abstract—**The Internet of Things envisions a multitude of heterogeneous objects and interactions with the physical environment. The functionalities provided by these objects can be termed as 'real-world services' as they provide a near real-time state of the physical world. A structured, machine-processible approach to provision such real-world services is needed to make heterogeneous physical objects accessible on a large scale and to integrate them with the digital world. This paper presents a semantic modeling approach for different components in an IoT framework. It is also discussed how the model can be integrated into the IoT framework by using automated association mechanisms with physical entities and how the data can be discovered using semantic search and reasoning mechanisms.**

I. Introduction

THE vision of the Internet of Things (IoT) relies on the provisioning of real-world services. The services are provided by a plethora of heterogeneous objects that are directly related to the physical world. Advancements in networking technologies and device capabilities enable a large number of physical world objects to have the communication and computation capabilities to connect and interact with their surrounding environment. The data and/or services offered by such objects can provide information about the physical world and allow interaction with it. These real-world data/services need to be defined and made available in a homogeneous way to allow integration of the data from different sources and to support autonomous reasoning and decision making mechanisms. Existing research initiatives have focussed on sensor (and actuator) middleware architectures that offer sensor measurement data services on the Web and/or at the application level. To extend this to heterogeneous physical world objects' data, this paper identifies the following requirements: a) identification of the various possible concepts in the IoT framework and their structured representation b) an access mechanism that offers a homogeneous interface to heterogeneous IoT objects with diverse capabilities, and c) automated machine-interpretability of the various interactions and integration with existing applications. This is necessary in order to integrate the physical world objects with the digital world and facilitate horizontal collaboration with existing software services.

This paper describes work undertaken in the context of the IoT-A project, IoT-A: Internet of Things – Architecture (http://www.iot-a.eu/public) contract number: 257521.

The information model presented in this paper captures the components of the IoT domain and provides a formal representation to the interactions. The paper is organised as follows: Section II presents relevant state-of-the-art in the IoT domain and sensor modeling. The proposed information models are detailed in Section III. The applicability of the models to infer associations with physical objects and to be utilized in a search framework is presented in Section IV. The implications of the modeling approach are discussed in Section V. Section VI concludes the paper and discusses the future work.

II. Related Work

Research initiatives and standardization activities in areas allied to the IoT vision have mainly focused on sensor descriptions and observation data modeling. The SENSEI project [1] aimed at realizing ambient intelligence in future networks and service environments by developing a framework of universal service interfaces for wireless sensor and actuator networks (WSANs). The core modeling concept considered in SENSEI is 'resource', with all sensors, actuators, and processors being modeled as resources [2]. A resource model captures resource functionalities, and where and how they can be accessed, in a conceptual view. The concrete instantiation of this information is contained in the resource description, which is published in a resource directory that acts as a service repository. Resources are described by a number of keywords. The syntax and semantics of the interfaces are captured in the advanced resource description, which is an ontology including concepts such as location, type (Sensor, Processor, Actuator), and operations of a resource. For each operation, it specifies the inputs that a resource takes in order to provide an output, the pre-conditions and post-conditions derived from invoking an operation and the temporal availability of the operation. The SENSEI resource model forms the basis of the models proposed in this paper, which are extended to encompass possible key concepts of the IoT domain.

There have been a number of works focusing on representation models for sensor data using ontologies, such as [3], [4]. OntoSensor [3] constructs an ontology-based descriptive specification model for sensors by excerpting parts of SensorML [5] descriptions and extending the IEEE Suggested Upper Merged Ontology (SUMO) (http://www.ontologyportal.org/). However, it does not provide a descriptive model for observation and measurement data. The work presented in [4] proposes an ontology-based model for service oriented sensor data and networks. However, it does not specify how to represent and interpret complex sensor data. The SensorData Ontology developed in [6] is built based on Observations & Measurements and SensorML specifications defined by the OGC Sensor Web Enablement (SWE) [7].

W3C's Incubator Group on Semantic Sensor Networks (SSN) (http://www.w3.org/2005/Incubator/ssn/) has introduced an ontology [8] to describe sensors and sensor networks. The ontology represents a high-level schema model to describe sensor devices, their capabilities, platform and other related attributes in the semantic sensor networks and the sensor Web applications. The SSN ontology, however, does not include modeling aspects for features of interest, units of measurement and domain knowledge that are related to sensor data and need to be associated with the sensor data to support autonomous data communications and efficient reasoning and decision making processes. In fact, the SSN ontology describes sensor devices, observation and measurement data and the platform aspects; however extensions to other components in the IoT domain are not specified in the ontology.

The CSIRO sensor ontology [9] was the precursor of the W3C SSN sensor ontology. It provides a semantic description of sensors in terms of the sensor grounding (platform, dimensions, calibration, power-source and access mechanism) and operation specification (operation, process and results). Concepts for sensor measurements are not part of the ontology. Moreover, similar to the SSN ontology, concepts for domain knowledge, units of measurement, location etc. are not included. Thus, more modeling concepts are needed to link the sensor descriptions to sensor measurements and then to the observed entity in the IoT domain. Sensor observations and measurements are modeled in the SemSOS O&M-OWL ontology [10]. The key concepts modeled are observation, process, feature (abstraction of real-world entity) and phenomenon (property of a feature that can be sensed or measured). The O&M concepts are aligned to SensorML and the feature and phenomenon concepts pertain to the weather domain. A similar approach to separate the observations from the entity being observed is presented in the SEEK Extensible Observation Ontology (OBOE) [11], which has a core observation ontology, a units extension, and a further extension for domain use (coastal ecosystems). Each observation is modeled to have a measurement, which is that of an entity's characteristic. An entity is supposed to serve as an extension point into domain models, with one particular example provided for a coastal ecosystem domain. The concepts in the OBOE ontology would require to be extended to include generic features of possible IoT entities. Also, placeholders to include sensor descriptions from other ontologies would be required.

The SemSerGrid4Env project has developed a service ontology that represents sensor web services provided by a sensor grid infrastructure [12]. In that model, Web Services are classified by the datasets they expose. SemSorGrid4Env considers that datasets conform to definitions such as OGC [7] or GeoJSON (http://geojson.org/geojson-spec.html). The service interface is defined according to ISO 19119 standard [13] specifying service operations together with their parameters. To annotate sensor observation values gathered by services with spatio-temporal meta-data, concepts from NASA's SWEET ontology (http://sweet.jpl.nasa.gov/) are used. To describe the physical phenomena observed by the sensor service, the concepts 'Property' and 'FeatureOfInterest' are borrowed from SSN sensor ontology. The SemSorGrid4Env Service ontology is suitable to describe sensor services about natural phenomena. To be able to describe arbitrary 'things' including human made artifacts, a more general description is needed.

Ontology Web Language for Services (OWL-S) [14] is a minimalistic approach for describing semantic Web Services. It is a service description framework that provides both rich expressive descriptions and well-defined semantics. OWL-S provides the main attributes to describe services and their functional attributes. It describes the characteristics of a service by using three top-level concepts, namely service profile, service-grounding, and service model. The profile is meant to be published to service repositories. It offers provider information, a functional description (inputs and outputs, preconditions and effects), and non-functional properties such as categorisation and quality rating. The service model describes the service's operation and enables invocation, composition, and monitoring of a service. It describes whether the service is atomic or composed of other atomic services. The grounding specifies how the service is invoked technically by the service consumer including a network address of the service endpoint. It also provides information about data-types used in the operations of services. It should be noted that although OWL-S uses Web Service Description Language (WSDL) [15] as its grounding mechanism, it is not restricted to WSDL as the only service technology. The OWL-S ontology is very flexible to use and thus it serves as upper ontology for the IoT-adapted Service Model proposed in this paper.

III. IoT Information Model

An IoT framework can benefit from structured models that detail various concepts and provide abstractions of the components and their attributes. This section defines the main abstractions and concepts that underlie the IoT domain and describes the relationships between them.

The main tenet of the IoT is extension of the Internet into the physical world, to involve interaction with a physical entity in the ambient environment. The entity constitutes 'things' in the Internet of Things and could be a human, animal, car, store or logistic chain item, electronic appliance or a closed or open environment. The 'entity' is the main focus of interactions by humans and/or software agents. This interaction is made possible by a hardware component, a 'device', which either attaches to an entity or is part of the environment of an entity so it can monitor it. The device allows the entity to be part of the digital world by mediating the interactions. The actual software component that provides information on the entity or enables controlling of the device, is a 'resource'. As implementations of resources can be highly dependent on the underlying hardware of the device, a 'service' provides a well-defined and standardised interface, offering all necessary functionalities for interacting with entities and related processes. The services expose the functionality of a device by accessing its hosted resources. Other services may invoke such low-level services for providing higher-level functionalities, for instance executing an activity of a specified business process. The relations between services and entities are modeled as associations. These associations could be static, e.g. in case the device is embedded into the entity; they could also be dynamic, e.g., if a device from the environment is monitoring a mobile entity. These identified concepts of the IoT domain and the relations between them are depicted in Figure 1.

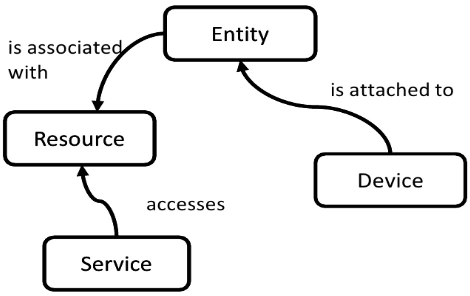

Fig. 1 IoT model: key concepts and interactions

The identified concepts need to be modeled in a format that provides interoperable and automated human and machine interpretable representations. The Semantic Web community has introduced formal definitions specified as ontologies that model different information in a domain, enable knowledge sharing and support automated reasoning. Specifically, the Web Ontology Language - Description Logic (OWL-DL), rooted in the decidable fragment of first-order logic, provides a powerful platform for a formal and machine-processible structure to represent the information that are collated from diverse sources.

Based on the identification above, of the main concepts in the IoT domain, this paper proposes a suite of ontologies that models entity, resources and IoT services. These will serve as a high-level model that references and builds upon existing vocabularies, as have been reviewed in section II. The concepts related to other relevant domains, such as sensors, observation and measurement and location, can be included from other ontologies. Where appropriate, properties are included to allow linking the proposed ontologies to external ontologies; for example, the global location URI of an entity could link to the relevant location instance in the GeoNames ontology (http://www.geonames.org/ontology/documentation.html), where the given location is more fully described. This enables reusability of ontologies and fosters modularity.

A. Entity Model

An entity can have certain aspects that need to be taken into account. For example, when one needs to know about the location of an entity or the features of interest that data is available for. The OWL-DL representation has been used to define the entity model. The entity ontology is available at http://purl.oclc.org/net/unis/EntityModel.owl. A diagram of the main attributes in the entity model is shown in Figure 2.

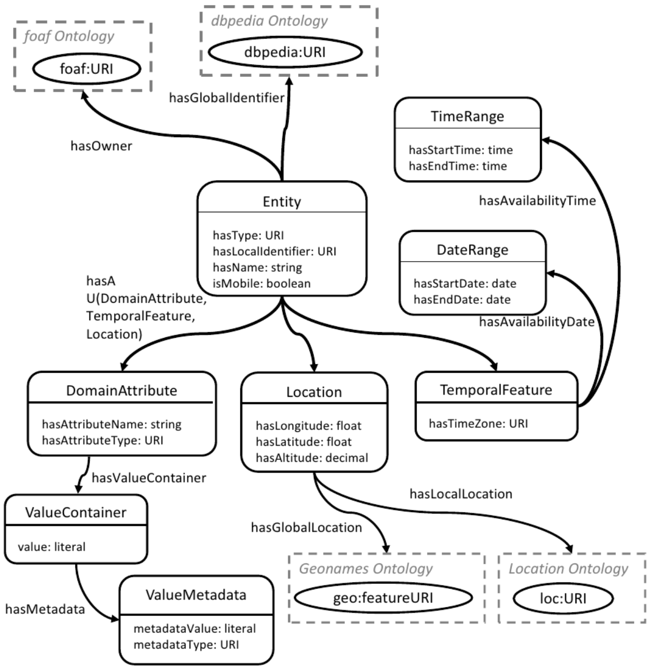

Fig. 2 The Entity model

An entity can have certain features, which include domain attributes, temporal features and location (Entity:hasA U(DomainAttribute, TemporalFeatures, Location)). Moreover, an entity instance can have multiple values for the domain, temporal or location feature. The observable features of an entity are specified by domain attributes that encapsulate the attribute name (hasAttributeName), attribute type (hasAttributeType) and one or more values in a value container (hasValueContainer). Each value container has the literal value specification (value), which is connected to metadata information. The metadata information can, for instance, be used to specify the units of measurement for the value, its timestamp or a notion of its quality. Temporal features are specified through time zone and through object properties to the time range (in terms of start and end time) and date range (start and end date) concepts. The location is defined in terms of the geographical coordinates (hasLatitude, hasLongitude, has Altitude). The location concept also has properties that link to global (hasGlobalLocation) and local location (hasLocalLocation) ontologies. The local location ontology provides detailed location description, such as rooms and buildings on a campus, whereas the global location ontology URI links the entity to existing high level location ontologies such as GeoNames, which provides toponyms or place names for cities, districts, countries and universities. Additionally, an entity has datatype properties that specify the URI of an owner (hasOwner) where the URI could point to a foaf ((http://www.foaf-project.org/docs) profile, a literal name (hasName) and a Boolean property to denote if the entity could be mobile (isMobile). An important attribute of an entity is the entity type (hasType). The local identifier (hasLocalIdentifier) property points to a local naming schema or literal representation of the entity and the global identifier (hasGlobalIdentifier) property is a placeholder to associate the entity to Linked Open Data (http://linkeddata.org) platform; for instance, to a DBpedia (http://dbpedia.org/) entry.

An illustrative example of an entity instance that implements the entity model is available at http://purl.oclc.org/net/unis/U38_Entity.owl. The instance is that of a room with ID 'RoomU38'. The entity type (http://www.owl-ontologies.com/LocationModel.owl#Room) and localIdentifier (http://www.owl-ontologies.com/ LocationModel.owl#U38) are mapped from a location ontology. The globalIdentifier links to the DBPedia entry for the institution of which the room is a part of, i.e. 'University of Surrey' in this case (http://dbpedia.org/resource/ University_of_Surrey). The local location (http:// http://surrey.ac.uk/ontologies/LocationModel.owl#BABuilding) is also specified from the location ontology and specifies the building location of the room. The globalLocation property links to the GeoNames feature URI of the town (http://www.-geonames.org/2647793/). The room has an ambient temperature attribute, with attribute type 'Temperature' (http://purl. oclc.org/NET/ssnx/qu/dim# Temperature). The attribute value is '17' and the associated metadata specifies that the unit of measurement is degreeCelsius, in terms of the metadata type (http://purl.oclc.org/NET/ssnx/qu/dim#TemperatureUnit) and metadataValue (http://www.qudt.org/qudt/owl/ 1.0.0/unit/Instances.html#DegreeCelsius).

B. Resource Model

A resource is the core software component that represents an entity in the digital world. Figure 3 details the resource description model. The resource model is available at http://purl.oclc.org/net/unis/ResourceModel.owl.

The resource concept has datatype properties that specify its name (hasName), an ID (hasResourceID) and a timezone defined in an external ontology (hasTimeZone). A resource also has a functional location property (hasResourceLocation) that links to the Location concept. This location could be the location of the device the resource runs on. The functional restriction denotes that a resource can only have a link to one location instance. The definition of the location concept is similar to the one defined in the entity model. The link to the resource type is denoted in terms of the type property (hasType) to the ResourceType concept. The resource type can be an instance of either of the following types: sensor, actuator, or tag. When the type is a sensor, the hasType property serves as a link to an instance of a sensor that conforms to an available sensor ontology

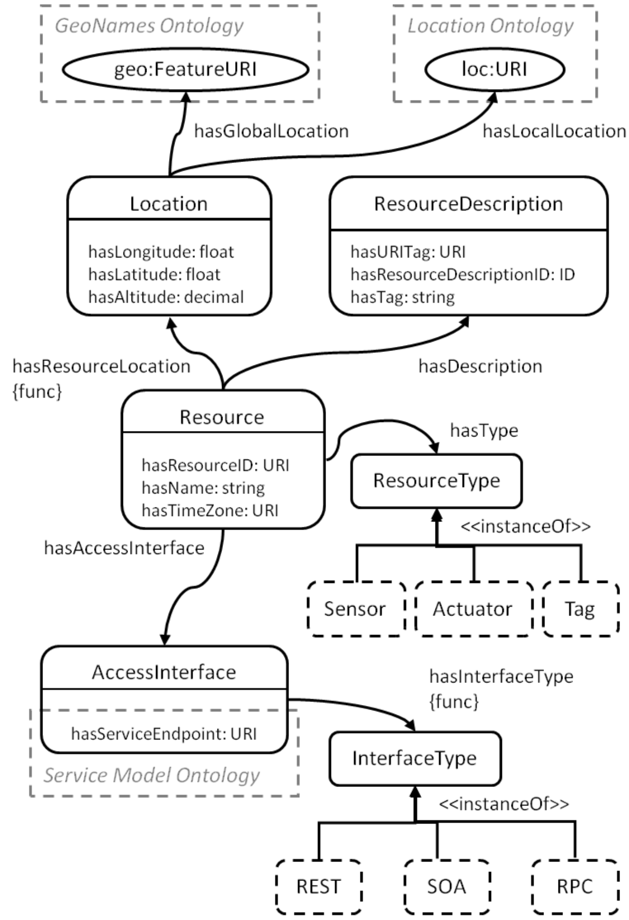

Fig. 3 The Resource model

(e.g. SSN sensor ontology). This allows linking the resource concept to external ontologies which define the related concepts without the need of repeating them in the proposed ontology suite. The interface to the resource (hasAccessInterface) is specified by the AccessInterface concept, which is further specified by an InterfaceType. The InterfaceType concept is defined as a set of instances which reflect technologies widely used in distributed systems, such as REST, SOAP, and RPC. The hasServiceEndpoint property links the resource model to the service model that exposes the resource functionalities to the IoT world.

Let 'U38_Temp_Sensor_Resource' be an example resource which hosts the temperature sensing capabilities in the location 'BaBuildingLocation'. The location has geographic properties of longitude, latitude, and altitude as well as links to a local ontology modeling the buildings on University of Surrey campus and to the GeoNames entry for Guildford that localises the resource on a global scale. The sensor resource is further described by the 'ResourceDescription_U38_temp_sensor' which contains a DBpedia classification of this resource and some tags describing the resource in plain text (temperature sensor in room 38 BA). The example resource is classified as 'Sensor' by the property hasType and it exposes the 'AccessInterface_U38_temp_sensor' to IoT-users which is declared as a RESTful interface by 'hasInterfaceType'. The access interface of this resource contains the locator of the service endpoint, which is part of the Service Model. The example resource presented here can be found at http://purl.oclc.org/net/unis/U38_Temp_Sensor_Resource.owl.

C. IoT Service Model

Resources are accessed by services which provide functionality to gather information about entities they are associated with or manipulate physical properties of their associated entities.

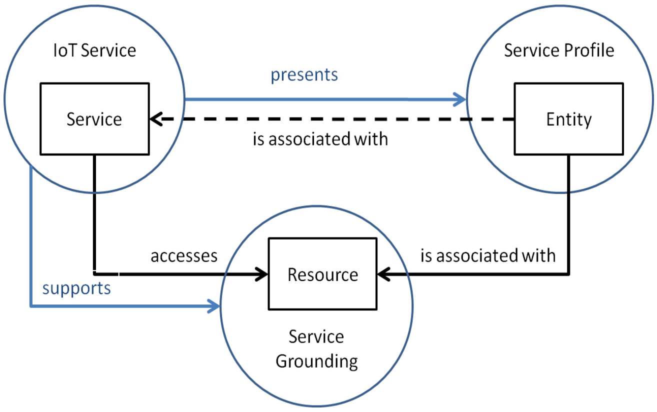

Fig. 4 The adapted OWL-S service ontology for IoT domain

The OWL-S specification has been designed as upper ontology for the Semantic Web Services. According to this specification, Semantic Web resources provide services which are described by their service profile, service model, and service grounding. Assuming potential IoT users are interested in information about the real world entities, they will search with terminology concerning entities of several domains. A search will return the service description containing a link to the resource offering the service that is able to satisfy user's information request. Thus, a service profile must contain information about the entity it is associated to as well as the link to the resource that provides the service about the entity. We use the OWL-S profile's object properties for this purpose. However, it must be noted that the association to an entity is not asserted (or may not be known at all) when the service is published; the link is asserted dynamically when an association is inferred. Mapping of OWL-S components to the identified IoT components (as demonstrated in Figure 1) is shown in Figure 4. The service profile describes services by their inputs, outputs, preconditions, and effects (IOPE). IoT sensing services provide output data service consumers are interested in (hasOutput). If a service needs any input to be processed by a resource it can be specified by a property (hasInput). Attributes of any entity can be used to describe the meaning of input and output parameters. Thus the IOPE properties of service profile link the Service Model to an Entity Model. Actuation services change properties of entities from an initial state to a desired state. The service profile's initial states are specified as precondition (hasPrecondition) and desired states are determined as resulting condition (hasEffect). These two object properties have a logic expression, a predicate, as range denoting a condition about an entity attribute. Such conditions, like 'equalTo' can be evaluated to true or false. A service will only be invoked if its precondition is evaluated to true.

We extend the existing profile with two more properties and their respective objects. 'ObservationArea' denotes the geographic area the service can observe (for sensors) or operate in (for actuators). With 'ObservationSchedule' it can be described when the service is able to operate and when it is planned to be out of work. The schedules can be used for maintenance, similar to SSN's OperatingRange or can be utilized for saving energy on the resource providing the services.

The resource is accessed over the Internet through a suitable interface, such as using a Web Service. The service endpoint is identified by a locator (URI) in the resource's AccessInterface. IoT users have access to this service endpoint the resource exposes, if not explicitly forbidden by privacy policies. The technical details that users need to know in order to access the service are specified in the service grounding. Since those details are dependent on the implementation of services and used technologies, they are not depicted in Figure 5. Typical information placed there are communication protocol, port number and the data types used for parameters that need to be sent to the service, as well as coming from the service, as depicted in Figure 5.

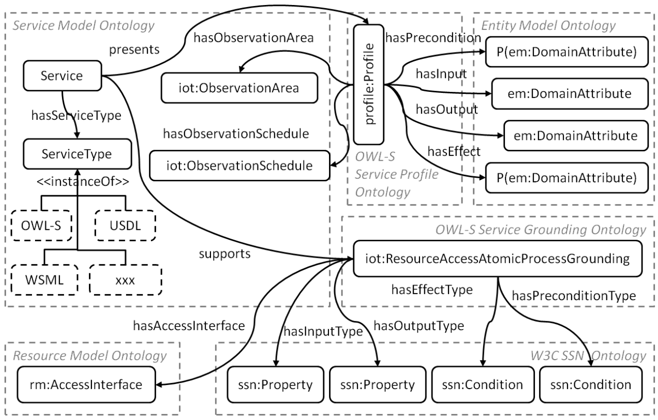

Fig. 5 Service Grounding

The ResourceAccessAtomicProcessGrounding specifies the mapping from domain specific entity attributes to properties observable by sensors. To each of the entity attributes assigned in the service profile an observation and measurement type can be assigned by their respective relations (hasInputType, hasOutputType, hasEffectType, and hasPreconditionType). The property hasInterfaceType determines the interface type as defined in the Resource Model. The IoT service model presented here is available at http://purl.oclc.org/net/unis/OWL-IoT-S.owl.

Let 'U38_Temp_Sensor_Resource' be the example resource that exposes the 'U38_TempSensor_Service'. This service has a type 'OWL-S' as specified by the hasServiceType property. The U38_TempSensorService_ Profile presents the service profile and supports the U38_TempSensorServiceProcessGrounding. The profile has links to U38_ObservationArea as well as U38_ObservationSchedule. The service output is described by the AmbientTempAttribute of the example entity 'RoomU38' which is defined using the Entity Model proposed in this paper. The link to the temperature sensor resource is established through the service grounding. The service grounding is realized by AccessInterface_ U38_temp_sensor which is part of the Resource Model for the example temperature sensor. The data type of the temperature measurement of this resource is determined by the range of property hasOutputType that is defined as a union of W3C SSN's 'Property' and a SENSEI Observation and Measurement type 'Temperature'.

The example service presented before is available at http://purl.oclc.org/net/unis/ U38_TempSensor_Service.owl.

IV. Using the Information models

A. Dynamic Associations

In the presented information models, physical entities and services that provide information or allow the interaction with the entities, are not connected through fixed links that are directly part of the entity or resource models, but instead are linked through separately modeled associations.

Having separate associations provides a higher level of flexibility. Services may be associated with multiple entities at the same time, e.g., a temperature sensor may provide the indoor temperature of a room and at the same time the ambient temperature of all the people who are currently in the room. As can be presumed from the example, the set of people in the room is changing, thus the valid associations can also change dynamically. For a small resource-constraint device providing the actual service, it might be a significant burden if it has to handle the resulting changes. Instead dynamic associations can be handled in a server infrastructure like a cloud, where communication and computing resources are plentiful. An additional advantage is that privacy can be better protected as services associated to people should not be visible to everybody, information that may again be harder to protect on a resource-constraint device.

In order to support dynamic associations, the associations first need to be discovered and then their validity has to be monitored. For this purpose, relevant aspects of both the entity and the device, which hosts the resource through which the service is provided, have to be monitored.

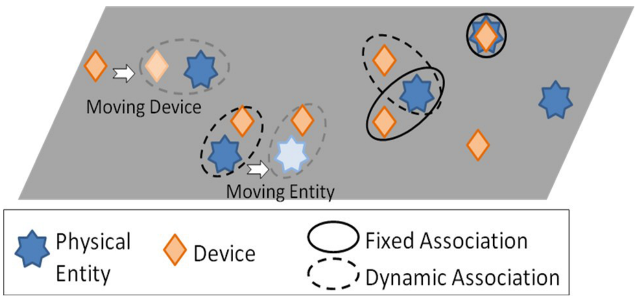

Fig. 6 Associations between physical entities and services provided through devices

Fig. 6 shows different associations between physical entities and services provided through devices. As both the physical entities and the devices can be mobile, the respective location or proximity of the entity and the device are relevant, but not necessarily sufficient indicators that a dynamic association is valid. Location information is explicitly modeled in both the Entity Model and the Resource Model, enabling both the specification of geographic coordinates as well as symbolic locations. Ownership or same movement patterns are examples for other relevant aspects that have to be taken into account for discovering dynamic associations.

An association also has to contain information about what aspect of the physical entity is being associated with the service. The ResourceType specifies what the service can do, e.g., provide information about the aspect in the case of a sensor, or change the aspect in case of an actuator.

B. Reasoning and Semantic Search

Utilizing the information represented in the form of the models and using them in IoT application and services also depends on finding relevant data and discovering entities, resources and/or services based on different scenarios. The semantic data can be represented in the form of Linked Data; i.e. links between entities, resources, service descriptions and also domain knowledge represented in the form of location ontologies, application data and resource in the Linked Open Data. In [16], we describe a Linked Data platform used for sensor descriptions that are represented and accessed in the form of linked data. Processing and reasoning large-scale semantic descriptions is also another important aspect to make the represented information more available to the end-users. In [17], we discuss a probabilistic machine learning mechanism to process semantic service descriptions for indexing and searching semantically described services. The introduced models provide similar type of descriptions so a similar method for indexing and searching the large-scale semantic data in the IoT domain can be adopted. Reasoning of resource, entity and service descriptions in relation to other data in the IoT domain and resources that describe application domain and environment attributes also enables to analyze the descriptions and supports autonomous communication and decision making processes. In [18], we have discussed some scenarios and concepts that utilize the sensor data and resource descriptions in the IoT domain.

V. Discussion

This paper focuses on describing the IoT component and data description models and captures relations between different data provider and data descriptor components in the IoT field. Our main objective is describing the entity, resource and service models for the IoT domain. We have also described how these models can be related to each other and can be also associated with the domain knowledge. The main advantage of introducing semantic models for the IoT component descriptions is providing interoperability in data and service levels. The models do not limit the data and/or service providers in what they can provide or provision; they, however, enable data/service providers to provision machine-interpretable data and descriptions such as what is provided, what the data/service is related to, where is the location of a data or a service provider, who is the provider. The models in general enable to describe spatial, temporal and thematic data related to data which is in line with the aspects that are also defined for the Semantic Sensor Web [19].

The semantic modeling and OWL/RDF descriptions solve the interoperability issues within the stakeholders that have agreed and/or provide data using the models. We have aligned our descriptions with the key players and existing standards and representation models in this domain. For other types of existing and future description models, it will be still possible to provide an alignment to map the descriptions across different IoT resource description frameworks. This however depends on the features that are described in different models and it would be applicable as long as the required and provided data can match to the designated attributes and assumptions that we have made in designing the models.

Timeliness of data and reasoning services is also another issue that needs to be considered while using semantic modeling and annotated data in the IoT domain. In large-scale deployments, identifying the relevant resources that can provide required data/services and reasoning with the domain knowledge can be a time consuming process. Effective utilization of these models depends on how efficiently the discovery and reasoning processes can perform as the number of components and the volume of descriptions increases.

Power and resource constraints and limited capabilities of the underlying devices is also another issue that should be considered when semantic data modeling is used in the IoT domain. In the introduced framework, we assume that the models are used to describe resources, entities and services and the semantic data is stored and utilized on powerful machines, e.g. gateway nodes or middleware components. This enables the devices to perform independently while the descriptions make their capabilities, descriptions and data more processible and interpretable for software agents and human users. The observation and measurement data can be also discussed in the middleware level and/or on the sensor node level within the capillary networks and then different techniques can be used to support effective communication of this data over lower power and low bandwidth devices and networks.

Manual versus automated annotations and associations processes is also another important issue in dealing with the detailed semantic models. The important question is that who will provide this semantic annotation and how this data for each component will be associated to other data and resources in the IoT domain and also to the existing data on the cyber world (i.e. the Web data). In [20], we discussed a middleware solution that uses predefined template models to provide semantic annotation for known types of sensors. A similar approach can be adapted for known types of resources, entities and services in the IoT domain. Association of the resources can be also supported by off-line reasoning processes that analyze the annotations and find the relation between different entities, resources and services based on different aspects such as location, type, and domain attributes.

VI. Conclusion and Future Work

The models proposed in this paper are designed based on our previous work and experiences in the SENSEI project and SSN ontology modeling and can support a general association between different components in the IoT domain. The models provide a semantic annotation framework so the legacy data can be also enhanced using these descriptions. The semantic annotation allows that the model data is represented as linked data and can be associated with the existing data on the Web and in particular Linked Open Data.

Future work will involve development of a resolution framework that allows searching the large scale data of the instances of the models in the IoT domain and will facilitate automated inference of dynamic associations.

References

[1] "SENSEI - Integrating the Physical with the Digital World of the Network of the Future," IST-FP7, 2010. http://www.sensei-project.eu/

[2] C. Villalonga, Ed., D2.5 Adaptive and Scalable Context Composition and Processing, *Public SENSEI Deliverable,* 2010.

[3] D. J. Russomanno, C. Kothari, and O. Thomas, "Sensor ontologies: from shallow to deep models," in *Proceedings of the Thirty-Seventh southeastern Symposium on System Theory,* 2005.

[4] J. H. Kim, K. Kwon, K. D.-H, and S. J. Lee, "Building a service-oriented ontology for wireless sensor networks," in *Proceedings of the Seventh IEEE/ACIS International Conference on Computer and Information Science,* 2008, pp. 649-654.

[5] OGC, "OpenGIS® Sensor Model Language (SensorML) Implementation Specification," *Open Geospatial Consortium, Inc.* 2007. http://www.opengeospatial.org/standards/sensorml.

[6] P. M. Barnaghi, S. Meissner, M. Presser, and K. Moessner, "Sense and sens'ability: Semantic data modelling for sensor networks," in *Proceedings of the ICT Mobile Summit,* 2009.

[7] OGC, "Open Geospatial Consortium (OGC) Sensor Web Enablement: Overview and High Level Architecture," *OGC white paper,* 2007.

[8] "W3C SSN Incubator Group Report". 2011. http://www.w3.org/2005/Incubator/ssn/wiki/Incubator_Report

[9] H. Neuhaus and M. Compton, "The Semantic Sensor Network Ontology: A Generic Language to Describe Sensor Assets," presented at the *Pre-Conference Workshop on Challenges in Geospatial Data Harmonisation (AGILE 2009),* Hannover, Germany, 2009.

[10] C. Henson, J. K. Pschorr, A. P. Sheth, and K. Thirunarayan, "SemSOS: Semantic Sensor Observation Service," in *Proc. of the 2009 International Symposium on Collaborative Technologies and Systems (CTS 2009),* Baltimore, MD, 2009.

[11] S. Bowers, J. S. Madin, and M. P. Schildhauer, "A Conceptual Modeling Framework for Expressing Observational Data Semantics," Q. Li et al. (Eds.): ER 2008, *LNCS* 5231, 2008, pp. 41–54.

[12] R. Garcia-Castro, C. Hill, and O. Corcho, "SemserGrid4Env Deliverable D4.3 v2 Sensor network ontology suite," Feb 28 2011.

[13] G. Percivall, "ISO 19119 and OGC Service Architecture," in *Proceedings of FIG XXII International Congress,* 2002.

[14] "OWL-S: Semantic Markup for Web Services ", *W3C Member Submission* 2004. http://www.w3.org/Submission/OWL-S.

[15] "Web Services Description Language (WSDL), 1.1", *W3C Note,* 2001. http://www.w3.org/TR/wsdl

[16] P. Barnaghi, M. Presser, and K. Moessner, "Publishing Linked Sensor Data," presented at the *Proc. 3rd International Workshop on Semantic Sensor Networks (SSN), in conjunction with the 9th International Semantic Web Conference (ISWC 2010),* 2010.

[17] G. Cassar, P. Barnaghi, and K. Moessner, "Probabilistic Methods for Service Clustering," presented at the *Proceedings of the 4th International Workshop on Semantic Web Service Matchmaking and Resource Retrieval,* 2010.

[18] W. Wang and P. Barnaghi, "Semantic annotation and reasoning for sensor data," presented at the *Proceedings of the 4th European conference on Smart sensing and context (EuroSSC2009),* Guildford, UK: Springer-Verlag, 2009.

[19] A. P. Sheth, C. Henson, and S. S. Sahoo, "Semantic sensor web," *IEEE Internet Computing,* vol. 12, pp. 78-83, 2008.

[20] F. Ganz, P. Barnaghi, F. Carrez, and K. Moessner, "Context aware management for sensor networks," in *Proceedings of the Fifth International Conference on COMmunication System softWAre and middlewaRE (COMSWARE11),* 2011.