CHARIOT - Specification Language for IoT Services and Data

Deliverable 4.1-2-3

WP4: Semantic Services & Data



Version	0.1
Due Date	30.04.2018
Delivery Date	30.04.2018
Nature	Report

Table of Contents

1. Introduction 4
2. Requirements 4
3. Related Work 5
   3.1 Functionality 5
   3.2 Quality of Service 6
   3.3 Context 6
4. Semantic Model 7
   4.1 Functional Description 7
   4.2 Quality of Service Description 8
   4.3 Context 9
Appendix 11
   A. Semantic Model 11

List of Figures

Figure 1: Overview of the Semantic Services & Data Work Package
Figure 2: Visual Representation of the Semantic Model
Figure 3: QoS Model

List of Tables

Table 1: Environment and related Sensors & Devices
Table 2: Simulated Sensors and Devices

Table of Acronyms

Acronym	Meaning
OWL	Web Ontology Language
QoS	Quality of Service
IoT	Internet of Things
DSL	Domain-Specific Language
SCO	Smart City Object
DS	Directory Service

1. Introduction

WP4-specific introduction (see Fig 1). When we speak about *IoT entity*, we mean both simples and complex devices as well as virtual services.

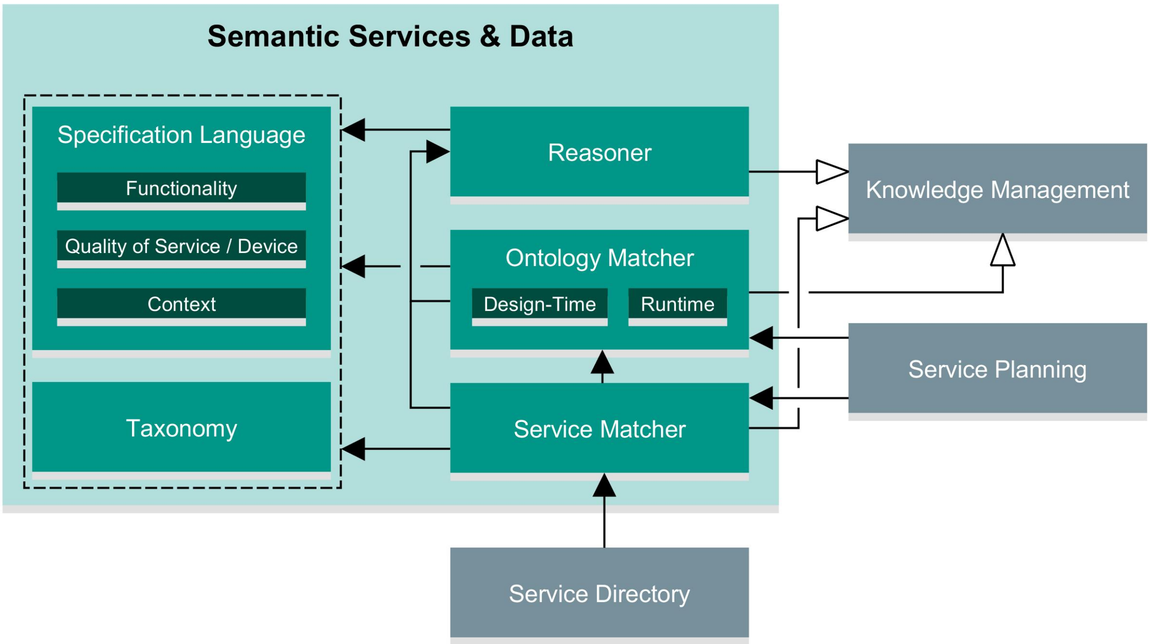

Figure 1: Overview of the Semantic Services & Data Work Package

2. Requirements

The IoT is a highly heterogeneous and dynamic environment that contains trillions of devices and virtual services. On the other hand, the functionalities provided by the individual entities overlap and are not known a priori, which complicates and impedes the efficient use of them. In conclusion, a technical representation that takes into account the mentioned attributes and problems has to solve the following challenges:

Functionality. We define IoT services by the functionality they provides to other entities; hence, the representation of such an entity has to reflect this: All information that are needed by a calling entity, such as an orchestrator, has to be accessible; *WP 4.1* covers this topic.

Data. Additional to functional attributes IoT entities can also be described by data that cannot be defined using a service model. Utilizing this data a better filtering of services that match a specific request is possible. We call this information data or context; *WP 4.3* covers this topic.

Heterogeneous Environment. The domain of IoT is not defined precisely and consists of several sub domains. The model has to give an appropriate, flexible way of defining the underlying data model of an entity so that even a requester from a different domain can utilize it.

Dynamic Environment. The technical representation has to be adaptable to a permanently changing environment. This means that the data model used by the entity cannot be statically defined, but has to be dynamic. This additionally requires a representation that can be semantically understood by the requesting entity.

Tremendous Amount of available services & devices. The individual entities provide a certain functionality that may be similar or even equal to other ones; in a city-scale environment this can even be thousands of devices or services. The technical representation has to provide suitable mechanisms that allows for further filtering to find the best matching SCO's. Among others, this can be achieved by quality of service (QoS) filtering; *WP 4.2* covers this topic.

3. Related Work

3.1 Functionality

Speaking about a reasonable way of sharing knowledge and modelling services and data in a heterogeneous and dynamic environment, the usage of ontologies seems to be the best fit; domain and data models described in an ontology provide a sophisticated semantic description that can be parsed by an application. The service-oriented computing community developed well-established standards to describe and invoke functionalities, such as WSDL/SOAP and REST services, but these standards do not suffice a smart city. Semantic descriptions of software services provide the needed additional information layer. Several approaches have been proposed in the last years. Amongst others, these are OWL-S[1], WSMO[2] and SAWSDL[3], which extend usual information by preconditions and effects and define input and output information in a more expressive ontology language, such as the Web Ontology Language (OWL)[4].

Nambi et al.[5] describe a semantic knowledge base for IoT and use, similar to our approach, the OWL-S ontology to describe services. However, the authors focus on describing the

[1] Martin, David, et al. "OWL-S: Semantic markup for web services." *W3C member submission* 22 (2004): 2007-04.

[2] Domingue, John, Dumitru Roman, and Michael Stollberg. "Web service modeling ontology (WSMO) - An ontology for semantic web services." (2005): 9-10.

[3] Kopecký, Jacek, et al. "Sawsdl: Semantic annotations for wsdl and xml schema." *IEEE Internet Computing* 11.6 (2007).

[4] McGuinness, Deborah L., and Frank Van Harmelen. "OWL web ontology language overview." *W3C recommendation* 10.10 (2004): 2004.

[5] Nambi, SN Akshay Uttama, et al. "A unified semantic knowledge base for IoT." *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014.

domain holistically with especially contextual information, and therefore create a rather complex multi-layered ontology; furthermore, orchestration of services is only brought up briefly. An approach that is not using OWL-S is described in the ontology-based resource description model (ORDM)[6]: It models sensors as services, but again focuses on a lightweight, compact representation.

Another way aside ontologies to describe a domain in a technical form that can later be used for implementations is to use so-called domain-specific languages (DSL); while they share a lot of attributes, they aim at different aspects of the problem domain[7]. There exist approaches for DSL's for the IoT domain[8,9]: While they provide the typical DSL benefit of efficient programming in the problem domain, they are e.g. restricted by the *Closed-World Assumption* that says that all knowledge that is true is known as true.

3.2 Quality of Service

3.3 Context

The IoT-Lite ontology[10] models the *Internet of Things (IoT)* domain in a descriptive way. The lightweight approach gives a good insight in how to efficiently classify the different accessible devices in a smart environment and what properties such a model should provide at least, but lacks at describing a functional view that can be called by a requester.

Perera et al.[11] made an extensive survey on context-aware computing in the context of IoT. Here, context is cited as "[...] any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves." [12] The survey gives a good introduction into the topic and heavily influenced our context model. They emphasize two different perspectives on the context model: The

---

6 Wang, Shulong, et al. "Sensing as Services: Resource-Oriented Service Publishing Method for Devices in Internet of Things." *Wireless Personal Communications* 95.3 (2017): 2239-2253.

7 Sutii, A. M., T. Verhoeff, and M. G. J. van den Brand. "Ontologies in domain specific languages: a systematic literature review." (2014).

8 Sneps-Sneppe, Manfred, and Dmitry Namiot. "On web-based domain-specific language for internet of things." *Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2015 7th International Congress on*. IEEE, 2015.

9 Negash, Behailu, et al. "DoS-IL: A Domain Specific Internet of Things Language for Resource Constrained Devices." *Procedia Computer Science* 109 (2017): 416-423.

10 Bermudez-Edo, Maria, et al. "IoT-Lite: a lightweight semantic model for the Internet of Things." *Ubiquitous Intelligence & Computing, 2016 Intl IEEE Conferences*. IEEE, 2016.

11 C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos: "Context Aware Computing for The Internet of Things: A Survey," *IEEE Commun. Surv. Tutor.*, pp. 1–41, 2013.
12 G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles: "Towards a Better Understanding of Context and Context-Awareness," *Proc. First Int. Symp. Handheld Ubiquitous Comput. (HUC '99)*, pp. 304–307, 1999.

conceptual perspective distinguishes between primary knowledge, which is directly given in the model, and secondary knowledge, which is derived from primary knowledge. The operational perspective looks on how the data is retrieved: It can be static, sensed, profiled (changing at low frequency) or derived. In another article the same authors also show their own approach for a context model that strongly focuses on the context of the calling entity to provide the content that is most probably currently needed.[13] Bettini et al. discuss several techniques for modelling and reasoning of context, amongst others ontological approaches, independently from a specific domain.[14] Interesting approaches on how to utilize the concept of context and context-awareness are given by Liu et al.[15] and Yanwei et al.[16]

4. Semantic Model

Our research revealed that there is no ontology that provides the possibility to describe services and data of the IoT domain in a shared knowledge base. Thus, in our approach we extend the OWL-S ontology, an extension to OWL that enables for describing semantic web services. It features the possibility to describe services with inputs, outputs, preconditions and effects (IOPE) using one or more ontologies. Furthermore, since in the IoT domain the devices with all their properties (context information, or data) is also very important, we also extend the ontology IoT-Lite; this ontology is a W3C submission member and an approach to provide a lightweight, descriptive ontology for the IoT domain. Using both OWL-S and IoT-Lite, both used and tested ontologies, we expect to have a solid basis for our own ontology.

4.1 Functional Description

Our extension integrates devices and, moreover, extends the concept of a service by attributes we identified as needed by a Smart City Object, e.g., the location or the accessibility. An overview of our ontology can be seen in Fig. 2: Great attention was paid to the interlocking of the presentation of services and the offering devices. The model basically allows for two different views on the domain:

1. The **functional**, service-oriented view, that involves OWL-S, an ontology established in the semantic web research community. This is the classical view.
2. The **descriptive** view, that involves IoT-Lite, an ontology that focuses on the core concept of the IoT. This view can be subdivided into a device-oriented and a surroundings-oriented view and anticipates the context extension/model.

[13] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “CA4IOT: Context awareness for Internet of Things,” in Proceedings - 2012 IEEE Int. Conf. on Green Computing and Communications, GreenCom 2012, Conf. on Internet of Things, iThings 2012 and Conf. on Cyber, Physical and Social Computing, CPSCom 2012, 2012, pp. 775–782.

[14] C. Bettini et al., “A survey of context modelling and reasoning techniques,” Pervasive Mob. Comput., vol. 6, no. 2, pp. 161–180, Apr. 2010.

[15] Y. Liu, B.-C. Seet, and A. Al-Anbuky, “An Ontology-Based Context Model for Wireless Sensor Network (WSN) Management in the Internet of Things,” J. Sens. Actuator Networks, vol. 2, no. 4, pp. 653–674, Sep. 2013.

[16] S. Yanwei, Z. Guangzhou, and P. Haitao, “Research on the context model of intelligent interaction system in the Internet of Things,” IT Med. Educ. (ITME), 2011 Int. Symp., vol. 2, pp. 379–382, 2011.

The CHARIOT ontology interconnects the two mentioned ontologies to fulfil the requirements of the CHARIOT project by using established and tested existing ontologies (reusing ontologies is seen as a best practice). By using an ontology the proposed model is not fixed and can easily be extended to fulfil upcoming requirements.

The central classes in the ontology are *Service*, which is a subclass of the OWL-S classes *Service*, *ServiceProfile*, *Process* and *Result,* and *Device*, which is directly taken out of the IoT-Lite ontology. This means that in terms of OWL-S CHARIOT services represent both the service as well as the underlying process with its result and, on the other hand, guarantees that CHARIOT services can be used by systems that understand OWL-S services.

Virtual services can be expressed using this ontology, however, the significant part is to also represent devices. Since we model the domain in terms of their functionality, we distinguish between *Actuator Service* and *Sensor Service* that only differ in terms of what they provide: An actuator changes the world's state, thus has an effect, a sensor measures a kind of quantity, thus has a single output.

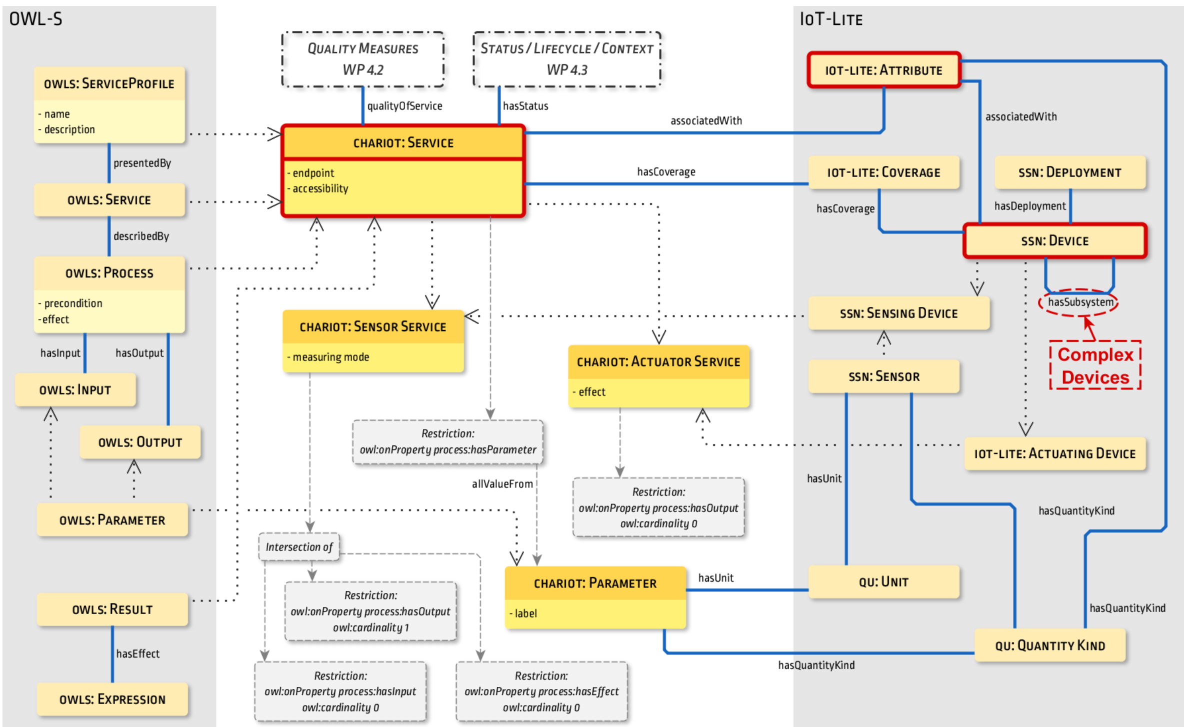

Figure 2: Visual Representation of the Semantic Model

4.2 Quality of Service Description

To address the filtering challenge (see **Requirements**), the design of the model allows for modelling QoS parameters assigned to the services: For entities that are functional equivalent, QoS parameters can be used to further narrow down the useful services and devices.

4.3 Context

Another important non-functional attributes that allow for a better selection of services and devices are so-called context information that describe the entities more comprehensive, but also less purposeful then the functional description. Our context model distinguishes between primary and secondary context: According to research on this topic the primary context is the basic, but most important information that comprises of location, identity, time and activity. The location is the geospatial information of the entities such as the region, for which a SCO provides its functionality, and of course the location for physical devices. Identity is a broad term; in our model we understand under identity every piece of information that is unique to the device or service, such as ID, hardware attributes (labeling, manufacturer, RAM size, etc). Time as a context information is obvious as (most of) the context is time-variable: Activity ist the most time-variable kind of information and comprises from our understanding information such as device stat (on/off), sensor values, etc. Secondary context on the other hand is derived from primary context (e.g. temporal sequence of locations gives velocity).

Another important categorization of context information is the operational perspective[17]: How is the information gained? The categories proposed [17] and used by our model are *Static*, *Sensed*, *Profiled* (changing at low frequency) and *Derived.*

Following the beforehand mentioned categories of context, our context model follows this guideline:

- Primary information, excluding Time, that is static and profiled

[17] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, “Context Aware Computing for The Internet of Things: A Survey,” IEEE Commun. Surv. Tutor., vol. X, no. X, pp. 1–41, 2013.

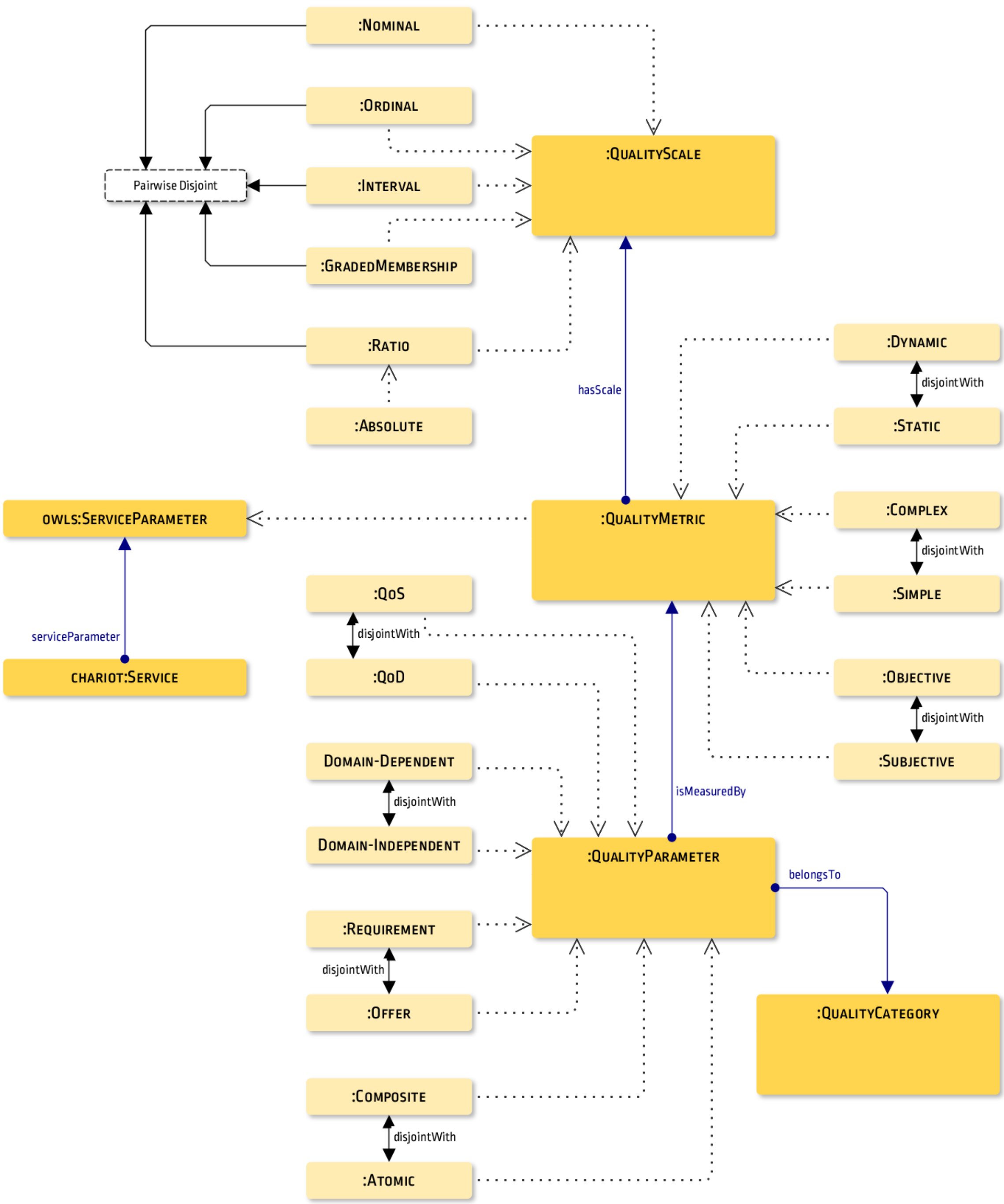

Figure 3: QoS Model

Appendix

A. Semantic Model

Following is attached the semantic model of the functional description of services and data in the Manchester Syntax[18]:

```
Prefix: : <http://chariot-km.dai-lab.de/ontologies/iot-services/iot-services.owl#>
Prefix: dc: <http://purl.org/dc/elements/1.1/>
Prefix: expression: <http://www.daml.org/services/owl-s/1.2/generic/Expression.owl#>
Prefix: iot-lite: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#>
Prefix: ns: <http://creativecommons.org/ns#>
Prefix: owl: <http://www.w3.org/2002/07/owl#>
Prefix: process: <http://www.daml.org/services/owl-s/1.2/Process.owl#>
Prefix: profile: <http://www.daml.org/services/owl-s/1.2/Profile.owl#>
Prefix: qu1: <http://purl.org/NET/ssnx/qu/qu#>
Prefix: qu: <http://purl.oclc.org/NET/ssnx/qu/qu#>
Prefix: qualityofservice: <http://isco-dev.aot.tu-berlin.de/ontologies/QualityOfService.owl#>
Prefix: rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
Prefix: rdfs: <http://www.w3.org/2000/01/rdf-schema#>
Prefix: service: <http://www.daml.org/services/owl-s/1.2/Service.owl#>
Prefix: skos: <http://www.w3.org/2004/02/skos/core#>
Prefix: ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
Prefix: swrl: <http://www.w3.org/2003/11/swrl#>
Prefix: swrla: <http://swrl.stanford.edu/ontologies/3.3/swrla.owl#>
Prefix: swrlb: <http://www.w3.org/2003/11/swrlb#>
Prefix: terms: <http://purl.org/dc/terms/>
Prefix: this: <http://chariot-km.dai-lab.de/ontologies/iot-services/iot-services.owl>
Prefix: wgs84_pos: <http://www.w3.org/2003/01/geo/wgs84_pos#>
Prefix: xml: <http://www.w3.org/XML/1998/namespace>
Prefix: xsd: <http://www.w3.org/2001/XMLSchema#>



Ontology: <http://chariot-km.dai-lab.de/ontologies/iot-services/iot-services.owl>
<http://chariot-km.dai-lab.de/ontologies/iot-services/1.0.0/iot-services.owl>
Import: <http://isco-dev.aot.tu-berlin.de/ontologies/QualityOfService.owl>
Import: <http://purl.oclc.org/NET/UNIS/fiware/iot-lite#>
Import: <http://purl.oclc.org/NET/ssnx/qu/qu>
Import: <http://purl.org/NET/dc_owl2dl/elements>
Import: <http://www.daml.org/services/owl-s/1.2/Process.owl>
Import: <http://www.daml.org/services/owl-s/1.2/Profile.owl>
Import: <http://www.daml.org/services/owl-s/1.2/Service.owl>

Annotations:
        dc:creator "Christian Kuster",
        dc:title "IoT Services",
        owl:versionInfo "1.0.0"

AnnotationProperty: dc:creator


AnnotationProperty: dc:title


AnnotationProperty: owl:versionInfo


AnnotationProperty: rdfs:comment


AnnotationProperty: rdfs:label


AnnotationProperty: swrla:isRuleEnabled


Datatype: rdf:PlainLiteral
```

[18] Horridge, Matthew, et al. "The Manchester OWL syntax." OWLed. Vol. 216. 2006.

Datatype: xsd:anyURI

Datatype: xsd:boolean

Datatype: xsd:string

ObjectProperty: hasAccessibility

SubPropertyOf:
owl:topObjectProperty

Domain:
Service

Range:
Accessibility

ObjectProperty: hasAttribute

Domain:
Service

Range:
iot-lite:Attribute

ObjectProperty: hasCoverage

Domain:
Service

Range:
iot-lite:Coverage

ObjectProperty: hasQuantityKind

Domain:
Parameter

Range:
qu1:QuantityKind

ObjectProperty: hasUnit

Domain:
Parameter

Range:
qu1:Unit

ObjectProperty: iot-lite:hasQuantityKind

ObjectProperty: isAttributeFor

Domain:
iot-lite:Attribute

Range:
Service

ObjectProperty: measuringMode

Domain:
Service

Range:
MeasuringMode

ObjectProperty: owl:topObjectProperty

ObjectProperty: process:hasEffect

ObjectProperty: process:hasInput

ObjectProperty: process:hasOutput

ObjectProperty: process:hasParameter

ObjectProperty: profile:hasOutput

ObjectProperty: qualityMeasure

Domain:
Service

Range:
qualityofservice:QualityMetric

ObjectProperty: service:supports

ObjectProperty: status

Domain:
Service

DataProperty: label

Domain:
Parameter

Range:
xsd:string

DataProperty: process:parameterType

Class: Accessibility

EquivalentTo:
{OpenAccess , RestrictedAccess}

Class: ActuatorService

EquivalentTo:
process:hasOutput exactly 0 owl:Thing

SubClassOf:
Service,
iot-lite:ActuatingDevice

DisjointWith:
SensorService

Class: MeasuringMode

EquivalentTo:

{ConstantMeasuringMode , ContinuousMeasuringMode , IntervalMeasuringMode , RequestMeasuringMode}

Class: Parameter

SubClassOf:
process:Parameter

Class: SensorService

EquivalentTo:
(process:hasEffect exactly 0 owl:Thing)
and (process:hasInput exactly 0 owl:Thing)
and (process:hasOutput exactly 1 owl:Thing)

SubClassOf:
Service,
ssn:SensingDevice

DisjointWith:
ActuatorService

Class: Service

Annotations:
rdfs:label "Service"@en,
rdfs:label "Dienst"@de,
rdfs:comment "A service is a smart object that does not represent a device, but serves abstract information."@en

EquivalentTo:
(service:supports min 1 owl:Thing)
and (hasAccessibility exactly 1 owl:Thing)

SubClassOf:
process:AtomicProcess,
process:Result,
profile:Profile,
service:Service,
process:hasParameter only Parameter

Class: iot-lite:ActuatingDevice

Class: iot-lite:Attribute

Class: iot-lite:Coverage

Class: owl:Thing

Class: process:AtomicProcess

Class: process:Parameter

Class: process:Result

Class: profile:Profile

Class: qu1:QuantityKind

Class: qu1:Unit

Class: qualityofservice:QualityMetric

Class: service:Service

Class: ssn:SensingDevice

Individual: ConstantMeasuringMode

Types:
MeasuringMode

Individual: ContinuousMeasuringMode

Types:
MeasuringMode

Individual: IntervalMeasuringMode

Types:
MeasuringMode

Individual: OpenAccess

Types:
Accessibility

Individual: RequestMeasuringMode

Types:
MeasuringMode

Individual: RestrictedAccess

Types:
Accessibility

Individual: param1

Types:
Parameter

Facts:
process:parameterType "http://purl.org/iot/vocab/m3-lite#Temperature"^^xsd:anyURI

Individual: sensor1

Types:
SensorService

Facts:
profile:hasOutput param1

Rule:
SensorService(?<http://chariot-km.dai-lab.de/ontologies/iot-services/iot-services.owl#service>), process:hasOutput(?<http://chariot-km.dai-lab.de/ontologies/iot-services/iot-services.owl#service>, ?<http://chariot-km.dai-lab.de/ontologies/iot-services/iot-services.owl#output>), process:parameterType(?<http://chariot-km.dai-lab.de/ontologies/iot-services/iot-services.owl#output>, ?<http://chariot-km.dai-lab.de/ontologies/iot-services/iot-services.owl#type>), iot-lite:hasQuantityKind(?<http://chariot-km.dai-lab.de/ontologies/iot-services/iot-services.owl#service>, ?<http://chariot-km.dai-lab.de/ontologies/iot-services/iot-services.owl#quantitykind>) -> swrlb:resolveURI(?<http://chariot-km.dai-lab.de/ontologies/iot-services/iot-services.owl#quantitykind>, ?<http://chariot-km.dai-lab.de/ontologies/iot-services/iot-services.owl#type>)