Koordinira/Coordinated by:
Fakultet organizacije i informatike/
Pavlinska 2/ 42000 Varaždin
www.foi.unizg.hr

Open IoT ontology

D 6.1 Open ontologija IoT uređaja – komponenta 1

D 6.1 Open ontology of IoT devices – 1. Component

D 6.2 Open ontologija IoT uređaja – komponenta 2

D 6.2 Open ontology of IoT devices – 2. component

Authors: Darko Andročec

Date: April 20th 2016

Revised: January 11th, 2017

Place: Faculty of Organization and Informatics - FOI, Varaždin

1. Introduction

The main aim of this document is to describe the development of the open Internet of things (IoT) ontology. This is part of the objective 6 of the project (Definition of the semantic annotations of IoT devices and their services – thing as a service). The ontology was developed incrementally. The first component of the ontology include elements for semantic annotation of IoT devices, and the second component defines concepts for semantic annotation of things as a service, their data types, and cloud services. All concepts are defined in the IoT OWL ontology that will be publicly available at project's web site.

2. Existing IoT ontologies

There are some existing works in current literature that deal or define some types of IoT ontologies. Mrissa et al. [1] proposed a framework built on the notion of avatar that can reason about functionalities and capabilities and

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

to propose Web of Things applications based on available devices. The main classes of their ontology are Appliance, Capability, and Functionality. Each functionality is described in terms of composing functionalities and implementing capabilities in a common shared ontology [1]. Christophe [2] developed a concept called virtual object made of properties allowing to associate FOAF vocabulary, describe a physical location, and describe temporal elements, associate time constraints with groups of users, associate a date to any produced content.

Wang et al. [3] proposed a sensing network element ontology description model for Internet of Things. The main concepts in their ontology are sensor, physical capability (location, battery, platform, size, weight, working status, etc.), observation value (value, its precision, frequency, response model, sensing range), and measurement value range (measurement unit, quality, sampling method).

Perera et al. [4] proposed a context-aware sensor search, selection and ranking model. They have developed an ontology based context framework for sensors which allows capturing and modelling context properties related to sensors. They used W3C Semantic Sensor Network Ontology [5] to model the sensor descriptions and context properties. The SSN ontology is capable of modelling a significant amount of information about sensors (its capabilities, performance, conditions of use, etc.) and it includes the most common context properties. Furthermore, the properties can be extended unlimitedly.

Kostelnik et al. [6] presented the middleware for a networking of physical devices, sensors, or components to provide higher value-added solutions to the end users. Their LinkSmart middleware aims at service/sensor data fusion and integration of these low-level data into business process workflow sequences. This was accomplished by combining the use of ontologies (based on the SSN ontology) with semantic web services. The ontology includes a generic Event concept, models of event results and event stimulus, and the connection to the service that triggers the event.

The W3C defined an OWL 2 ontology [5] named the Semantic Sensor Network (SSN) ontology to describe the capabilities and properties of sensors, the act of sensing and the resulting observations. Also, concepts for operating and survival ranges and their performance is included. The paper [5] is cited a lot in relevant research paper, and many IoT interoperability projects use or extend this ontology to semantically describe sensors. The SSN ontology consists of 41 concepts and 39 object properties. Observation are fully described by the SSN ontology, while feature and property are empty concepts, so for particular application the ontology must be extended via subclassing or equivalence relations. Sensor in the SSN is anything that observes. Central to the SSN ontology is the Stimulus-Sensor-Observation ontology design pattern and the SSN ontology has been developed as a minimal, common ground for heavy-weight ontologies for the Semantic Sensor Web. Stimuli are changes or states in an environment that a sensor can detect and use to measure a property [5]. Sensors are physical objects that observe by transforming incoming stimuli into another representation [5]. An observation can link the act of sensing, the event that is the stimulus, the sensor, a method, a result, an observed feature, and property, placing all in an interpretative context [5]. The ontology models accuracy, detection limit, drift, frequency, latency, measurement range, precision, response time, resolution, sensitivity, and selectivity as measurement capabilities. Observations are contexts for interpreting incoming stimuli. This context includes observed feature, property,

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

observing sensor, result, and method. The SSN ontology does not include other entities that exist in IoT such as physical objects, actuators and complex systems [7].

Mathew et al. [8] produced the ontology for things on the Web. They claim things can be classified into four fundamental dimensions: identity (use of an appropriate identification systems), processing (functions which allow things to be controlled or managed), communication (thing's communication interface to enable interactions among things), and storage (the type and amount of information that a thing retains). They classified things as core (it is uniquely identified within a given context), primitive (fuzzy – uniquely identified and has process information, plug – uniquely identified and has a communication interface, fat – uniquely identified and has storage capability but does not have processing capability or communication interface), complex, and smart things.

Wang et al. [9] have developed the description ontology using a knowledge-driven approach to capture the most important concepts and their relationships in the IoT domain. The linked data principle is fundamental for their design, some concepts are linked with external domain ontologies. Their description ontology consists of seven main modules: IoT Services (IoT services are defined as subclasses of the Service class defined in OWL-S); Service Test (verifying functional and non-functional cpaabilites of IoT services); QoS and quality of information (QoI); Deployment, Systems and Platform; Observation and Measurement (reuse of concepts related to observation and measurement from the SSN ontology); IoT Resources (this module extends SSN ontology to add Actuator, Iot Gateway, etc.); and Entity of Interest and Physical Locations. They emphasized that IoT services tend to be less reliable and there is the need for testing during the whole service lifecycle and additional mechanism for service adaptation and recomposition are needed.

He et al. [10] proposed a smart Web service based on the context of things, which is implemented using their Thing-REST style. Their Thing-REST style manage the context of things and user context, and can mashup web services through tree structures (for example, chain, select, and merge) to implement smart web services. The ontologies and domain knowledge are used to unify and understand the context of things. The context of a thing includes knowledge about both the situation around the thing and the thing itself. The semantic context of a thing is created by people and is generally unified by ontology services on the Semantic Web. In its simplest form, the basic abstract of a single sensor is a 2-tuple (physical parameter, value). They differ two types of actions towards things: people actions and actuator actions. Chain structures in Thing-REST represent a sequence of web services chained together, where the output of one Web service is the input of another Web service. Midterm results are stored in XML files. So, semantic web services (and related standards) and their composition are not used in their work, custom XML files and programming are used to mashup different services. They created the use case of smart plant-watering service that mashes up four heterogeneous Web services.

Nambi et al. [7] presented a unified semantic knowledge base for IoT that uses several ontologies. Resource ontology represents an entity (sensors, actuators, physical objects, composite objects) in IoT, and it was developed as extension of the SSN ontology. Location ontology extends GeoNames ontology and its aim is to add geospatial semantic information for things. Context and Domain ontologies represent contextual information and

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

specific knowledge about domains like smart home, smart transportation, etc. Policy ontology is used to provide information on how to accomplish a service requested by an actor in dynamic environments. Their service ontology is built upon OWL-S and it describes how services can be defined, represented and modeled.

Spalazzi et al. [11] extended the SSN ontology with concepts and roles that describe actuators. They proposed the Actuator-Stimulus-Operation pattern in order to describe actuators. An actuator is an object that can modify a property in environment, stimulus is able to modify a property, and an operation represents the activity that is accomplished by the actuator. They also created domain-related concepts to the IoT ontology for the earth-quake emergency scenario.

3. Selected ontology development methodology, tool and language

For the purpose of this research, the Ontology Development 101 [12] methodology was selected. This methodology was chosen among others, because it is the simplest and it is really focused on the results, i.e. building the first ontology version very fast and then refining it according to requirements. Ontology Development 101 is designed as a simple iterative methodology and a starting guide for new ontology designers to develop their own ontologies. Furthermore, it is also well aligned with the used tool (Protégé) and it provides working examples for this ontology editor. The open-source tool Protégé was selected because it is free and currently most used tool for ontology development. As an illustration, Protégé has more than 300,000 registered users at the moment. Web Ontology Language (OWL) was chosen because it has the needed expressive power and is most widely used language for ontologies in the papers in the field of computer science and research projects related to this field of study.

Now, the main steps of the selected ontology will be listed. Noy and McGuinness [12] claim that the development of the ontology includes defining classes and their hierarchy, defining their properties and instances. The ontology development process is iterative, an initial version is built, this version is checked in applications or by experts, and it is refined until usable ontology is obtained. There are seven steps in Ontology Development 101 methodology [12]:

1. Determine the domain and scope of the ontology – First step includes defining ontology's domain and scope by using competency questions (questions that the ontology should be able to answer).
2. Consider reusing the existing ontologies – Checking whether the existing ontologies can be refined and extended.
3. Enumerate important terms in the ontology – Write down all the possible relevant terms without worrying about the overlap between concepts.
4. Define the classes and the class hierarchy – Using top-down or bottom-up approach, or the combination of the two, to define classes and their hierarchy.
5. Define the properties of classes – slots – Here the internal structure of concepts is defined using data and object properties.

6. Define the facets of the slots – The value type, allowed values, domain, range, and cardinality of slots should be defined.
7. Create instances – The individual instances of classes should be defined and their slot values should be filled.

As part of their published document, Noy and McGuinness [12] showed how to create sample Wine ontology using the above mentioned steps. In the next section, the Ontology Development 101 methodology is used to create open IoT ontology.

4. Open IoT ontology

4.1. Domain and scope of the ontology

In the first step of Ontology Development 101 [12] guide, the domain and scope of the model should be limited. The representation of IoT devices and things as a services is determined as the domain of the ontology. This ontology will be used to semantically annotate things as a service. The information in the ontology should provide answers to the following questions: What are the concepts to describe IoT devices and things as a services? How to support mappings of data types among the heterogeneous things and existing cloud services?

4.2. Reusing the existing ontologies

First, the work of the other authors was considered and checked if there was a possibility to refine and extend the existing ontologies for the domain and scope determined in the previous step. As a basis for our ontology, we have used concepts defined in the W3C defined Semantic Sensor Network (SSN) ontology [5] described in more details in Section 2 of this report. The paper [5] is cited a lot in relevant research paper, and many existing IoT interoperability projects use or extend this ontology to semantically describe sensors. We have also used the actuator concepts from Semantic Actuator Network Ontology developed by Spalazzi et al. [11]. This ontology describes actuators and operations, and related concepts. It extends the Semantic Sensor Network ontology developed by the W3C SSN Incubator Group.

4.3. Important terms for the ontology

A list of all the relevant terms was identified in this step, without worrying about the overlap between the concepts or considering whether the concepts were OWL classes or properties. Excel spreadsheets were used to list all relevant terms, one sheet per one relevant document. Table 1. lists important terms per source.

Table 1 Important terms for the ontology

Source	Identified important terms
SSN ontology	Entity, Feature of Interest, Input, Output, Abstract, Formal entity, Set, Region, Amount, Observation value, Physical attribute, Social attribute, Space region, SpatioTemporalRegion, Time interval, Event, Action, Process, Deployment-related Process, Deployment, Sensor Input, Stimulus, InformationEntity, Information

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

	object, Sensor Data Sheet, Sensor Output, Information realization, Object, Agent, Person, Natural person, Social person, Physical agent, Social agent, Collective agent, Community, Group, Organization, Personification, Social person, Physical object, Physical agent, Physical artifact, Designed artifact, DesignedSubstance, Device, Sensing Device, Physical body, Biological object, Organism, Chemical object, Substance, Functional substance, DesignedSubstance, Physical place, Platform, Sensor, Sensing device, System, Device, Social object, Collection, Collective, Configuration, Type collection, Concept, Event type, Task, Local concept, Parameter, Unit of measure, Role, Description, Contract, Design, Diagnosis, Goal, Method, Narrative, Norm, Plan, Project, Workflow, Relation, Pattern, Social relation, Right, Theory, Information object, Place, Situation, Classification, Observation, Plan execution, Workflow execution, Transition, Quality, Property, Condition, Measurement Property, Accuracy, Detection limit, Drift, Frequency, Latency, Measurement Range, Precision, Resolution, Response time, Selectivity, Sensitivity, Measurement Capability, Operating Property, Maintenance Schedule, Operating Power Range, Operating Range, Survival Property, Battery Lifetime, System Lifetime, Survival Range
SAN ontology	Operation Value, Actuator Output, Actuator Input, Actuator, Acting Device, Acting, Operation, Change Capability, Change Property
BonSAI ontology	Context, Actuator, ActuatorSensor, MultiSensor, Operation, ActuatorOperation, SensoryOperation, Parameter
openIoT ontology	Cloud Access, Anonymous Access, Credential Access, Cloud Service Status, Service Disabled, Service Enabled, Context, MobilityContext, MeasurementCapability, Service Attribute, Cloud computing service, Cloud computing user, VirtualSensorTask, Confidentiality, CostEfficiency, EaseOfDeployment, Reliability, ResourceOptimization, SensorUtility, Survivability

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

Surrey ontology	QualityOfInformation, AccuracyQoI, CompletenessQoI, CorrectnessQoI, CostQoI, CredibilityQoI, FreshnessQoI, LatencyQoI, PrecisionQoI, ProvenanceQoI, SecurityQoI,TimelinessQoI, QualityOfService, AvailabilityQoS, NetworkQualityOfService, ReliabilityQoS, RobustnessQoS, SecurityQoS, TimelinessQoS, IoTResource, Actuator, GateWay, IoTServer, RFIDtag, Sensor, IoTService, REST Service, SOAP Service
Mrissa et. al	Appliance, Capability, ActuatorCapability, ProcessingCapability, SensorCapability, Functionality
IoT protocols (from http://www.postscapes.com/internet-of-things-protocols/)	IoT protocols, IoT data protocols, IoT discovery protocols, IoT infrastructure protocols, IoT transport protocols
IoT security problems (derived from OWAPS IoT - https://www.owasp.org/index.php/IoT_Security_Guidance)	IoT security problems, insecure cloud interface, insecure mobile interface, insecure network services, insecure software firmware, insecure web interface, insufficient security configurability, lack of transport encryption, poor physical security, privacy concerns

4.4. Classes and their hierarchy

From the list created in the previous step, the terms describing independent objects were selected to present classes in the ontology. In OWL, classes are used to group individuals that have something in common and that represent sets of individuals. A class can have subclasses, so the classes were organized into a hierarchical taxonomy. A total of 173 classes were defined that are organized in 20 top level classes (Figure 1). All classes are systematically specified in Table 2.

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

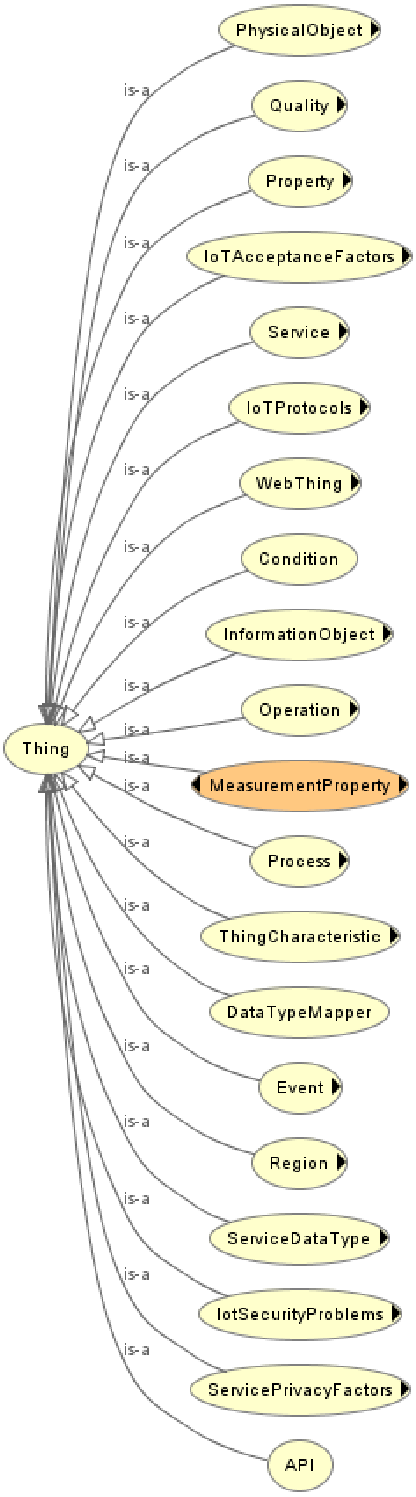

Figure 1 The main hierarchy of the IoT ontology

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

Table 2 Classes in the ontology

Class	Super class	Description
API	Thing	It represents vendors' Application Programming Interfaces (APIs).
DataTypeMapper	Thing	Its instances are used for data type mappings.
DesignedArtifact	Thing	Designed artifact
Device	DesignedArtifact	A device is a physical piece of technology - a system in a box. Devices may of course be built of smaller devices and software components (i.e. systems have components).
Acting Device	Device	An acting device is a device that implements acting.
Sensing Device	Device	A sensing device is a device that implements sensing.
Event	Thing	Event
Actuator Output	Event	Actuator output
Sensor Input	Event	An Event in the real world that 'triggers' the sensor. The properties associated to the stimulus may be different to eventual observed property. It is the event, not the object that triggers the sensor.
Stimulus	Event	An Event in the real world that 'triggers' the sensor. The properties associated to the stimulus may be different to eventual observed property. It is the event, not the object that triggers the sensor.
Feature of Interest	Thing	A feature is an abstraction of real world phenomena (thing, person, event, etc).
InformationObject	Thing	Information object
Actuator Input	InformationObject	An actuator requires an information (a value you want to change), the value itself being represented by an OperationValue.
Sensor Data Sheet	InformationObject	A data sheet records properties of a sensor. A data sheet might describe for example the accuracy in various conditions, the power use, the types of connectors that the sensor has, etc. Generally a sensor's properties are recorded directly (with hasMeasurementCapability, for example), but the data sheet can be used for example to record the manufacturers specifications verses observed capabilites, or if more is known than the manufacturer specifies, etc. The data sheet is an information object about the

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

		sensor's properties, rather than a direct link to the actual properties themselves.
Sensor Output	InformationObject	A sensor outputs a piece of information (an observed value), the value itself being represented by an ObservationValue.
Input	Thing	Any information that is provided to a process for its use [MMI OntDev].
Method	Thing	Method
Process	Method	A process has an output and possibly inputs and, for a composite process, describes the temporal and dataflow dependencies and relationships amongst its parts. [SSN XG]
Acting	Process	Acting is a process that results in the modification or control of the value of a phenomenon.
Sensing	Process	Sensing is a process that results in the estimation, or calculation, of the value of a phenomenon.
Object	Thing	Object
Operation	Thing, Situation	Operation
ActuatorOperation	Operation	Operation that results in a change of the world's state i.e. hasEffect object property has a value.
SensoryOperation	Operation	Operation that returns a parameter.
Output	Thing	Any information that is reported from a process. [MMI OntDev]
PhysicalObject	Thing	Physical object
Actuator	PhysicalObject	An actuator can do (implements) acting: an actuator is any entity that can follow an acting method and thus control some *Property* of a *FeatureOfInterest*. Actuator may be physical device or any other thing can follow an Acting Method to control a *Property*.
Acting Device	Actuator, Device	An acting device is a device that implements acting.
Platform	PhysicalObject	An Entity to which other Entities can be attached - particularly Sensors and other Platforms. For example, a post might act as the Platform, a buoy might act as a Platform, or a fish might act as a Platform for an attached sensor.
Sensor	PhysicalObject	A sensor can do (implements) sensing: that is, a sensor is any entity that can follow a sensing method and thus observe some *Property* of a *FeatureOfInterest*. Sensors may be physical devices, computational methods, laboratory setup with a person following a

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

		method, or any other thing that can follow a *Sensing Method* to observe a *Property*.
Sensing Device	Sensor, Device	A sensing device is a device that implements sensing.
System	PhysicalObject	System is a unit of abstraction for pieces of infrastructure (and we largely care that they are) for sensing. A system has components, its subsystems, which are other systems.
Device	System	A device is a physical piece of technology - a system in a box. Devices may of course be built of smaller devices and software components (i.e. systems have components).
Process	Thing	Process
Deployment-related Process	Thing	Place to group all the various Processes related to Deployment. For example, as well as deployment, installation, maintenance, deployment of further sensors and the like would all be classified under *DeploymentRelatedProcess*.
Deployment	Deployment-related Process	The ongoing Process of Entities (for the purposes of this ontology, mainly sensors) deployed for a particular purpose. For example, a particular Sensor deployed on a Platform, or a whole network of Sensors deployed for an observation campaign. The deployment may have sub processes, such as installation, maintenance, addition, and decommissioning and removal.
Quality	Thing	Quality
Property	Quality	An observable Quality of an Event or Object. That is, not a quality of an abstract entity as is also allowed by DUL's Quality, but rather an aspect of an entity that is intrinsic to and cannot exist without the entity and is observable by a sensor.
Change Capability	Property	Collects together operation properties (accuracy, range, precision, etc) and the environmental conditions in which those properties hold, representing a specification of an actuator's capability in those conditions. The conditions specified here are those that affect the operation properties, while those in *OperatingRange* represent the actuator's standard operating conditions, including conditions that don't affect the operation.
Change Property	Property	An identifiable and observable characteristic of an actuator's operations or ability to make operations.
Condition	Property	Used to specify ranges for qualities that act as conditions on a system/sensor's operation. For example, wind speed of 10-60m/s is expressed as a condition linking a quality, wind speed, a unit of measurement, metres per second, and a set of values, 10-60, and

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

		may be used as the condition on a MeasurementProperty, for example, to state that a sensor has a particular accuracy in that condition.
Measurement Property	Property	An identifiable and observable characteristic of a sensor's observations or ability to make observations.
Accuracy	Measurement Property	The closeness of agreement between the value of an observation and the true value of the observed quality.
Detection limit	Measurement Property	An observed value for which the probability of falsely claiming the absence of a component in a material is Î², given a probability Î± of falsely claiming its presence.
Drift	Measurement Property	A, continuous or incremental, change in the reported values of observations over time for an unchanging quality.
Frequency	Measurement Property	The smallest possible time between one observation and the next.
Latency	Measurement Property	The time between a request for an observation and the sensor providing a result.
Measurement Range	Measurement Property	The set of values that the sensor can return as the result of an observation under the defined conditions with the defined measurement properties. (If no conditions are specified or the conditions do not specify a range for the observed qualities, the measurement range is to be taken as the condition for the observed qualities.)
Precision	Measurement Property	The closeness of agreement between replicate observations on an unchanged or similar quality value: i.e., a measure of a sensor's ability to consistently reproduce an observation.
Resolution	Measurement Property	The smallest difference in the value of a quality being observed that would result in perceptibly different values of observation results.
Response time	Measurement Property	The time between a (step) change in the value of an observed quality and a sensor (possibly with specified error) 'settling' on an observed value.
Selectivity	Measurement Property	Selectivity is a property of a sensor whereby it provides observed values for one or more qualities such that the values of each quality are independent of other qualities in the phenomenon, body, or substance being investigated.
Sensitivity	Measurement Property	Sensitivity is the quotient of the change in a result of sensor and the corresponding change in a value of a quality being observed.

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

Measurement Capability	Property	Collects together measurement properties (accuracy, range, precision, etc) and the environmental conditions in which those properties hold, representing a specification of a sensor's capability in those conditions. The conditions specified here are those that affect the measurement properties, while those in *OperatingRange* represent the sensor's standard operating conditions, including conditions that don't affect the observations.
Operating Property	Property	An identifiable characteristic of the environmental and other conditions in which the sensor is intended to operate. May include power ranges, power sources, standard configurations, attachments and the like.
Maintenance Schedule	Operating Property	Schedule of maintenance for a system/sensor in the specified conditions.
Operating Power Range	Operating Property	Power range in which system/sensor is expected to operate.
Operating Range	Property	The environmental conditions and characteristics of a system/sensor's normal operating environment. Can be used to specify for example the standard environmental conditions in which the sensor is expected to operate (a *Condition* with no *OperatingProperty*), or how the environmental and other operating properties relate: i.e., that the maintenance schedule or power requirements differ according to the conditions.
Survival Property	Property	An identifiable characteristic that represents the extent of the sensors useful life. Might include for example total battery life or number of recharges, or, for sensors that are used only a fixed number of times, the number of observations that can be made before the sensing capability is depleted.
Battery Lifetime	Survival Property	Total useful life of a battery.
System Lifetime	Survival Property	Total useful life of a sensor/system (expressed as total life since manufacture, time in use, number of operations, etc.).
Survival Range	Property	The conditions a sensor can be exposed to without damage: i.e., the sensor continues to operate as defined using *MeasurementCapability*. If, however, the *SurvivalRange* is exceeded, the sensor is 'damaged' and *MeasurementCapability* specifications may no longer hold.
QualityOfService	Quality	Quality of service
AvailabilityQoS	QualityOfService	Availability
NetworkQualityOfService	QualityOfService	Network quality of service

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

Delay	NetworkQualityOfService	Delay
Jitter	NetworkQualityOfService	Jitter
PacketLossRate	NetworkQualityOfService	Rate of packet loss
Throughput	NetworkQualityOfService	Throughput
PrivacyQoS	QualityOfService	Privacy
ReliabilityQoS	QualityOfService	Reliability
RobustnessQoS	QualityOfService	Robustness
SecurityQoS	QualityOfService	Security
TimelinessQoS	QualityOfService	Timeliness
Region	Thing	Region
Observation Value	Region	The value of the result of an Observation. An Observation has a result which is the output of some sensor, the result is an information object that encodes some value for a Feature.
Operation Value	Region	The value you want to change or set trough an Operation. An Operation requires a desired result which is the input for some actuator, the desired result is an information object that encodes some value for a Feature.
Service	Thing	Service
CloudService	Service	Cloud service
IaaSService	CloudService	Infrastructure as a service
PaaSService	CloudService	Platform as a service
SaaSService	CloudService	Software as a service
RESTService	Service	REST Service
SOAPService	Service	SOAP Service
ThingAsAService	Service	Thing as a service
ServiceDataType	Thing	Data types for input and outputs of the services.
ComplexServiceDataType	ServiceDataType	Complex service data type.
SimpleServiceDataType	ServiceDataType	Simple service data type.

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

Situation	Thing	Situation
Observation	Situation	An *Observation* is a *Situation* in which a *Sensing method* has been used to estimate or calculate a value of a *Property* of a *FeatureOfInterest*. Links to *Sensing* and *Sensor* describe what made the *Observation* and how; links to *Property* and *Feature* detail what was sensed; the result is the output of a *Sensor*; other metadata details times etc.
IoTAcceptanceFactors	Thing	IoT acceptance factors
Age	IotAcceptanceFactors	Age
BehavioralIntention	IotAcceptanceFactors	Behavioral intention
EffortExpectancy	IotAcceptanceFactors	Effort expectancy
Experience	IotAcceptanceFactors	Experience
FacilitatingConditions	IotAcceptanceFactors	Facilitating conditions
Gender	IotAcceptanceFactors	Gender
PerformanceExpectancy	IotAcceptanceFactors	Performance expectancy
SocialInfluence	IotAcceptanceFactors	Social influence
UseBehavior	IotAcceptanceFactors	Use behavior
VoluntarinessOfUse	IotAcceptanceFactors	Voluntariness of use
IoTProtocols	Thing	IoT protocols derived from http://www.postscapes.com/internet-of-things-protocols/.
IoTDataProtocols	IotProtocols	IoT data protocols
AMQP	IoTDataProtocols	Advanced Message Queuing Protocol
CoAP	IoTDataProtocols	Constrained Application Protocol
DDS	IoTDataProtocols	Data-Distribution Service for Real-Time Systems
JMS	IoTDataProtocols	Java Message Service
LLAP	IoTDataProtocols	Lightweight Local Automation Protocol
LWM2M	IoTDataProtocols	Lightweight M2M
M3DA	IoTDataProtocols	Mihini/M3DA

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

MQTT	IoTDataProtocols	Message Queuing Telemetry Transport
MQTT-SN	IoTDataProtocols	MQTT for sensor networks
Reactive_Streams	IoTDataProtocols	Reactive Streams
SSI	IoTDataProtocols	Simple Sensor Interface
STOMP	IoTDataProtocols	The Simple Text Oriented Messaging Protocol
XMPP	IoTDataProtocols	Extensible Messaging and Presence Protocol
XMPP-IoT	IoTDataProtocols	XMPP-IoT
IoTDiscoveryProtocols	Thing	IoT discovery protocols
HyperCat	IoTDiscoveryProtocols	HyperCat - An open, lightweight JSON-based hypermedia catalogue format for exposing collections of URIs.
mDNS	IoTDiscoveryProtocols	mDNS (multicast Domain Name System) - Resolves host names to IP addresses within small networks that do not include a local name server.
PhysicalWeb	IoTDiscoveryProtocols	Physical Web - The Physical Web enables you to see a list of URLs being broadcast by objects in the environment around you with a Bluetooth Low Energy (BLE) beacon.
UPnP	IoTDiscoveryProtocols	Universal Plug and Play
IoTInfrastructureProtocols	Thing	IoT infrastructure protocols
Aeron	IoTInfrastructureProtocols	Efficient reliable UDP unicast, UDP multicast, and IPC message transport.
CCN	IoTInfrastructureProtocols	Content-Centric Networking
DTLS	IoTInfrastructureProtocols	Datagram Transport Layer
IPv6	IoTInfrastructureProtocols	IPv6, is an Internet Layer protocol for packet-switched internetworking and provides end-to-end datagram transmission across multiple IP networks.
LOWPAN	IoTInfrastructureProtocols	6LoWPAN is an acronym of IPv6 over Low power Wireless Personal Area Networks. It is an adaption layer for IPv6 over IEEE802.15.4 links. This protocol operates only in the 2.4 GHz frequency range with 250 kbps transfer rate.
NanoIP	IoTInfrastructureProtocols	NanoIP, which stands for the nano Internet Protocol, is a concept that was created to bring Internet-like networking services to embedded and sensor devices, without the overhead of TCP/IP.

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

Koordinira/Coordinated by:
Fakultet organizacije i informatike/
Pavlinska 2/ 42000 Varaždin
www.foi.unizg.hr

		NanoIP was designed with minimal overheads, wireless networking, and local addressing in mind.
QUIC	IoTInfrastructureProtocols	(Quick UDP Internet Connections, pronounced quick) supports a set of multiplexed connections between two endpoints over User Datagram Protocol (UDP), and was designed to provide security protection equivalent to TLS/SSL, along with reduced connection and transport latency, and bandwidth estimation in each direction to avoid congestion.
ROLL	IoTInfrastructureProtocols	ROLL / RPL (IPv6 routing for low power/lossy networks)
TSMP	IoTInfrastructureProtocols	Time Synchronized Mesh Protocol
UDP	IoTInfrastructureProtocols	User Datagram Protocol
uIP	IoTInfrastructureProtocols	The uIP is an open source TCP/IP stack capable of being used with tiny 8- and 16-bit microcontrollers.
IoTTransportProtocols	Thing	IoT transport protocols
ANT	IoTTransportProtocols	ANT is a proprietary wireless sensor network technology featuring a wireless communications protocol stack that enables semiconductor radios operating in the 2.4 GHz Industrial, Scientific and Medical allocation of the RF spectrum ("ISM band") to communicate by establishing standard rules for co-existence, data representation, signalling, authentication and error detection.
Bluetooth	IoTTransportProtocols	Bluetooth works in the 2.4 GHz ISM band and uses frequency hopping.
Cellular	IoTTransportProtocols	GPRS/2G/3G/4G cellular
DigiMesh	IoTTransportProtocols	DigiMesh is a proprietary peer-to-peer networking topology for use in wireless end-point connectivity solutions.
EC-GSM-IoT	IoTTransportProtocols	EC-GSM-IoT (Extended Coverage-GSM-IoT) - Enables new capabilities of existing cellular networks for LPWA (Low Power Wide Area) IoT applications. EC-GSM-IoT can be activated through new software deployed over a very large GSM footprint, adding even more coverage to serve IoT devices.
Eddystone	IoTTransportProtocols	Eddystone - A protocol specification that defines a Bluetooth low energy (BLE) message format for proximity beacon messages.
EnOcean	IoTTransportProtocols	EnOcean is a an energy harvesting wireless technology which works in the frequencies of 868 MHz for Europe and 315 MHz for North America

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

Ethernet	IoTTransportProtocols	Ethernet
IEEE_802.15.4	IoTTransportProtocols	IEEE 802.15.4 is a standard which specifies the physical layer and media access control for low-rate wireless personal area networks (LR-WPANs).
ISA100.11a	IoTTransportProtocols	SA100.11a is a wireless networking technology standard developed by the International Society of Automation (ISA).
LORaWAN	IoTTransportProtocols	Network protocol intended for wireless battery operated Things in regional, national or global network.
LTE-MTC	IoTTransportProtocols	LTE-MTC (LTE-Machine Type Communication) - Standards-based family of technologies supports several technology categories, such as Cat-1 and CatM1, suitable for the IoT.
NB-IoT	IoTTransportProtocols	NB-IoT (Narrow-Band IoT) A technology being standardized by the 3GPP standards body.
NFC	IoTTransportProtocols	Based on the standard ISO/IEC 18092:2004, using inductive coupled devices at a center frequency of13.56 MHz.
RPMA	IoTTransportProtocols	RPMA (Random phase multiple access) A technology communication system employing direct-sequence spread spectrum (DSSS) with multiple access.
Weightless	IoTTransportProtocols	Weightless is a proposed proprietary open wireless technology standard for exchanging data between a base station and thousands of machines around it (using wavelength radio transmissions in unoccupied TV transmission channels) with high levels of security.
WiFi	IoTTransportProtocols	WiFi
WiMax	IoTTransportProtocols	WiMax is based on the standard IEEE 802.16 and is intended for wireless metropolitan area networks.
WirelessHart	IoTTransportProtocols	WirelessHART technology provides a robust wireless protocol for the full range of process measurement, control, and asset management applications.
ZigBee	IoTTransportProtocols	The ZigBee protocol uses the 802.15.4 standard and operates in the 2.4 GHz frequency range with 250 kbps.
IotSecurityProblems	Thing	IoT security problems derived from OWAPS IoT (https://www.owasp.org/index.php/IoT_Security_Guidance).
InsecureCloudInterface	IotSecurityProblems	Covers cloud APIs or cloud-based web interfaces.
InsecureMobileInterface	IotSecurityProblems	Insecure mobile interface

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

InsecureNetworkServices	IotSecurityProblems	Insecure network services
NetworkServicesVulnerable-ToDenailOfService	InsecureNetworkServices	Network services vulnerable to Denial of Service (DoS) attacks
PortsExposedToTheInternet-ViaUPnP	InsecureNetworkServices	Ports exposed to the Internet via UPnP
UnnecessaryPortsAreOpen	InsecureNetworkServices	Unnecessary ports are opened
InsecureSoftwareFirmware	IotSecurityProblems	Insecure software firmware
DeviceUpdatesNotSigned	InsecureSoftwareFirmware	Device updates are not digitally signed
DeviceUpdatesTransmitted WithoutEncryption	InsecureSoftwareFirmware	Device updates transmitted without encrpytion
UpdateServersAreNotSecured	InsecureSoftwareFirmware	Update servers are not secured
InsecureWebInterface	IotSecurityProblems	Insecure web interface
DefaultUsernamesAnd-Passwords	InsecureWebInterface	Web interface uses default usernames and passwords.
NoAccountLockout	InsecureWebInterface	No account lockout
XSS_CSRF_SQLi_Vulnerabilities	InsecureWebInterface	XSS, CSRF, and SQLi vulnerabilities
InsufficientAuthentication-OrAuthorization	IotSecurityProblems	Insufficient authentication or authorization
InsecurePasswordRecovery	InsufficientAuthentication-OrAuthorization	Insecure password recovery
NoTwoFactorAuthentication	InsufficientAuthentication-OrAuthorization	There is no two factor authentication.
WeakPasswords	InsufficientAuthentication-OrAuthorization	Weak passwords
InsufficientSecurity-Configurability	IotSecurityProblems	Insufficient security configurability

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

EncryptionOptionsAreNot-Available	InsufficientSecurity-Configurability	Encryption options are not available
NoOptionToEnableSecurity-Logging	InsufficientSecurity-Configurability	No option to enable security logging
PasswordSecurityOptionsAre-NotAvailable	InsufficientSecurity-Configurability	Password security options are not available
LackOfTransportEncryption	IotSecurityProblems	Lack of transport encryption
NotAvailableOrNotProperly-ConfiguredSSL_TLS	LackOfTransportEncryption	Not available or not properly configured SSL/TLS
SensitiveInformationPassed-InClearText	LackOfTransportEncryption	Sensitive information passed in clear text
UseOfProprietaryEncryption-Protocols	LackOfTransportEncryption	Use of proprietary encryption protocols
PoorPhysicalSecurity	IotSecurityProblems	Poor physical security
AccessToOperatingSystems-ThroughRemoveMedia	PoorPhysicalSecurity	Access to operating systems through remove media
InabilityToLimitAdministrative-Capabilities	PoorPhysicalSecurity	Inability to limit administrative capabilities
UnnecessaryExternalPortsLike-USBPorts	PoorPhysicalSecurity	Unnecessary external ports like USB ports.
PrivacyConcerns	IotSecurityProblems	Privacy concerns
CollectingTooMuchPersonal-Information	PrivacyConcerns	Collecting too much personal information
EndUserIsNotGivenAChoiceTo-AllowCollectionOfCertain-TypesOfData	PrivacyConcerns	End user is not given a choice to allow collection of certain types of data
NotProperlyProtectionOf-CollectedInformation	PrivacyConcerns	Not properly protection of collected information.
ServicePrivacyFactors	Thing	Service privacy factors
CompensationForInformation-Provision	ServicePrivacyFactors	Compensation for information provision
ImproperAccess	ServicePrivacyFactors	Improper access

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

InformationCongruency	ServicePrivacyFactors	Information congruency
InformationSensitivity	ServicePrivacyFactors	Information sensitivity
InformationType	ServicePrivacyFactors	Information type
InformationUsage	ServicePrivacyFactors	Information usage
PerceivedBenevolence	ServicePrivacyFactors	Perceived benevolence
PerceivedCredibility	ServicePrivacyFactors	Perceived credibility
PerceivedIntegrity	ServicePrivacyFactors	Perceived integrity
PerceivedRisk	ServicePrivacyFactors	Perceived risk
PrivacyAttitude	ServicePrivacyFactors	Privacy attitude
ServiceProviderReputations	ServicePrivacyFactors	Service providers' reputations
ServiceQuality	ServicePrivacyFactors	Service quality
UnauthorizedSecondaryUse	ServicePrivacyFactors	Unauthorized secondary use
UserEducation	ServicePrivacyFactors	User's education
WebThing	Thing	http://model.webofthings.io/#web-things-model - A Web Thing (or simply Thing) is a digital representation of a physical object accessible via a RESTful Web API. Examples of Web Things are: an Arduino board, a garage door, a bottle of soda, a building, a TV, etc. The API of the Web Thing can be hosted on the Thing itself or on an intermediate host in the network such as a Gateway or a Cloud service (for Things that aren't accessible through the Internet).
SemanticWebThing	WebThing	It additionally supports semantic annotations using this (open IoT) ontology.

4.5. Properties of classes

The properties of classes describe the internal structure of concepts. Properties specify how the instances of a class relate to other instances. Property cardinality defines how many values a property can have. The allowed classes for a property instance are called a range of a property, and the classes that the property describes are called the domain of the property [12]. A set of defined object properties, along with their corresponding domains, ranges is shown in Table 3.

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

Table 3 Object properties defined in the IoT ontology

Object property	Domain	Range	Description
definesOperation	API	Operation	
describes			
implemented by			A relation between the description of an algorithm, procedure or method and an entity that implements that method in some executable way. For example, between a scientific measuring method and a sensor that senses via that method.
detects			A relation from a sensor to the *Stimulus* that the sensor can detect. The *Stimulus* itself will be serving as a proxy for (see *isProxyOf*) some observable property.
for property			A relation between some aspect of a sensing entity and a property. For example, from a sensor to the properties it can observe, or from a deployment to the properties it was installed to observe. Also from a measurement capability to the property the capability is described for. (Used in conjunction with *ofFeature*).
has input	Operation	Input	
has output	Operation	Output	
hasLocation			
on platform			Relation between a *System* (e.g., a *Sensor*) and a *Platform*. The relation locates the sensor relative to other described entities entities: i.e., the Sensor s1's location is Platform p1. More precise locations for sensors in space (relative to other entities, where attached to another entity, or in 3D space) are made using DOLCE's Regions (*SpaceRegion*).
hasMappingFrom	DataTypeMapper	ServiceDataType	
hasMappingTo	DataTypeMapper	ServiceDataType	
hasPart			
deployment process part			Has part relation between a deployment process and its constituent processes.
has subsystem			Has part relation between a system and its parts.

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

hasParticipant			
deployed on platform			Relation between a deployment and the platform on which the system was deployed.
deployed system			Relation between a deployment and the deployed system.
hasQuality			
has property	FeatureOfInterest	Property	A relation between a FeatureOfInterest and a Property of that feature.
has change capability	Actuator	ChangeCapability	Relation from an Actuator to a ChangeCapability describing the properties of the actuator.
has change property			Relation from a *ChangeCapability* to a *ChangeProperty*. For example, to an accuracy (see notes at *ChangeCapability*).
has measurement capability			Relation from a *Sensor* to a *MeasurementCapability* describing the measurement properties of the sensor.
has measurement property			Relation from a *MeasurementCapability* to a *MeasurementProperty*. For example, to an accuracy (see notes at *MeasurementCapability*).
has operating property			Relation from an *OperatingRange* to a *Property*. For example, to a battery lifetime.
has operating range			Relation from a *System* to an *OperatingRange* describing the normal operating environment of the *System*.
has survival property			Relation from a *SurvivalRange* to a *Property* describing the survival range of a system. For example, to the temperature extreme that a system can withstand before being considered damaged.
has survival range			A *Relation* from a *System* to a *SurvivalRange*.
quality of observation			*Relation* linking an *Observation* to the adjudged quality of the result. This is of course complimentary to the *MeasurementCapability* information recorded for the *Sensor* that made the *Observation*.
quality of operation			*Relation* linking an *Operation* to the adjudged quality of the result. This is of course complimentary to the *ChangeCapability* information recorded for the *Actuator* that made the *Operation*.
hasRegion			

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

end time			
has Value			
observation result time			The result time is the time when the procedure associated with the observation act was applied.
observation sampling time			The sampling time is the time that the result applies to the feature-of-interest. This is the time usually required for geospatial analysis of the result.
operation desired result time			The result time is the time when the procedure associated with the operation act was applied.
operation sampling time			The sampling time is the time that the desired result applies to the feature-of-interest. This is the time usually required for geospatial analysis of the result.
start time			
hasServiceQuality	Service	QualityOfService	
in condition			Describes the prevailing environmental conditions for *MeasurementCapabilites*, *OperatingConditions* and *SurvivalRanges*. Used for example to say that a sensor has a particular accuracy in particular conditions. (see also *MeasurementCapability*)
includesEvent			
includesObject			
observedBy			
operatedBy			
is produced by			Relation between a producer and a produced entity: for example, between a sensor and the produced output.
is required by	Actuator	Input	Relation between an actuator and the required input.
isDescribedBy			
implements			A relation between an entity that implements a method in some executable way and the description of an algorithm, procedure or method. For example, between a *Sensor* and the scientific measuring method that the *Sensor* uses to observe a *Property*.

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

isLocationOf			
attached system			*Relation* between a *Platform* and any *Systems* (e.g., *Sensors*) that are attached to the *Platform*.
isObjectIncludedIn			
made observation			*Relation* between a *Sensor* and *Observations* it has made.
made operation			*Relation* between an *Actuator* and *Operation* it has made.
isParticipantIn			
has deployment			Relation between a *System* and a *Deployment*, recording that the *System/Sensor* was deployed in that *Deployment*.
in deployment			Relation between a *Platform* and a *Deployment*, recording that the object was used as a platform for a system/sensor for a particular deployment: as in this *PhysicalObject* is acting as a *Platform inDeployment Deployment*.
isProxyFor			A relation from a *Stimulus* to the *Property* that the *Stimulus* is serving as a proxy for. For example, the expansion of the quicksilver is a stimulus that serves as a proxy for temperature, or an increase or decrease in the spinning of cups on a wind sensor is serving as a proxy for wind speed.
isQualityOf			
is property of			Relation between a *FeatureOfInterest* and a *Property* (a *Quality* observable by a sensor) of that feature.
isRegionFor			
isSettingFor			
controlled property			Relation linking an *Operation* to the *Property* that was controlled. The controlledProperty should be a *Property* (*hasProperty*) of the *FeatureOfInterest* (linked by *featureOfInterest*) of this operation.
feature of interest			A relation between an observation and the entity whose quality was observed. For example, in an observation of the weight of a person, the feature of interest is the person and the quality is weight.
observation result			*Relation* linking an *Observation* (i.e., a description of the context, the *Situation*, in which the observation was made)

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

			and a *Result*, which contains a value representing the value associated with the observed *Property*.
observed property			Relation linking an *Observation* to the *Property* that was observed. The *observedProperty* should be a Property (*hasProperty*) of the *FeatureOfInterest* (linked by *featureOfInterest*) of this observation.
operation desired result			Relation linking an *Operation* (i.e., a description of the context, the *Situation*, in which the operation was made) and a desired result, which contains a value representing the value associated with the controlled *Property*.
models			A relation from an actuator to the stimulus that the actuator can control. The stimulus itself will be serving as a proxy for (see *isProxyOf*) some editable property.
modify			Relation between a *Actuator* and a *Property* that the actuator can change.
observes			Relation between a *Sensor* and a *Property* that the sensor can observe. Note that, given the DUL modelling of Qualities, a sensor defined with 'observes only Windspeed' technically links the sensor to particular instances of Windspeed, not to the concept itself - OWL can't express concept-concept relations, only individual-individual. The property composition ensures that if an observation is made of a particular quality then one can infer that the sensor observes that quality.
of feature			A relation between some aspect of a sensing entity and a feature. For example, from a sensor to the features it can observe properties of, or from a deployment to the features it was installed to observe. Also from a measurement capability to the feature the capability is described for. (Used in conjunction with *forProperty*).
satisfies			
acting method used			A control procedure is a detailed description of a control according to one or more control principles and to a given control method, based on a control model and including any calculation to obtain desired result.
sensing method used			A (measurement) procedure is a detailed description of a measurement according to one or more measurement principles and to a given measurement method, based on a

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

			measurement model and including any calculation to obtain a measurement result.
usesPhysicalObject	Operation	PhysicalObject	
isRepresentedBy	PhysicalObject	WebThing	Physical object is represented by web thing.

4.6. Creating instances

The last step in the methodology devised by Noy and McGuinnes [12] is filling in the values for instances. It requires the creation of individual instances of each relevant class. For now, all hierarchy data is contained in concepts defined as OWL classes and object properties, so no individuals were created.

5. Conclusion

This document presents the process of the development of the Internet of thing ontology using prominent Ontology Development 101 methodology [12]. The ontology will be used (and possible refined) in next steps of our research, and will be used to semantically annotated various smart things. The ontology is publicly available at https://github.com/dandrocec/IoTOntology.

6. References

[1] M. Mrissa, L. Medini, and J.-P. Jamont, “Semantic Discovery and Invocation of Functionalities for the Web of Things,” in 2014 IEEE 23rd International WETICE Conference, 2014, pp. 281–286.

[2] B. Christophe, “Semantic Profiles to Model the ‘Web of Things,’” in 2011 Seventh International Conference on Semantics Knowledge and Grid (SKG), Beijing, China, 2011, pp. 51–58.

[3] X. Wang, H. An, Y. Xu, and S. Wang, “Sensing Network Element Ontology Description Model for Internet of Things,” in 2015 2nd International Conference on Information Science and Control Engineering (ICISCE), Shanghai, 2015, pp. 471–475.

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.

[4] C. Perera, A. Zaslavsky, C. H. Liu, M. Compton, P. Christen, and D. Georgakopoulos, “Sensor Search Techniques for Sensing as a Service Architecture for the Internet of Things,” IEEE Sens. J., vol. 14, no. 2, pp. 406–420, Feb. 2014.

[5] M. Compton, P. Barnaghi, L. Bermudez, R. García-Castro, O. Corcho, S. Cox, J. Graybeal, M. Hauswirth, C. Henson, A. Herzog, V. Huang, K. Janowicz, W. D. Kelsey, D. Le Phuoc, L. Lefort, M. Leggieri, H. Neuhaus, A. Nikolov, K. Page, A. Passant, A. Sheth, and K. Taylor, “The SSN ontology of the W3C semantic sensor network incubator group,” Web Semant. Sci. Serv. Agents World Wide Web, vol. 17, pp. 25–32, Dec. 2012.

[6] P. Kostelnik, M. Sarnovsky, and K. Furdik, “The Semantic Middleware for Networked Embedded Systems Applied in the Internet of Things and Services Domain,” Scalable Comput. Pract. Exp., vol. 2, no. 3, 2011.

[7] S. N. A. U. Nambi, C. Sarkar, R. V. Prasad, and A. Rahim, “A unified semantic knowledge base for IoT,” in 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, 2014, pp. 575–580.

[8] S. S. Mathew, Y. Atif, Q. Z. Sheng, and Z. Maamar, “Web of Things: Description, Discovery and Integration,” in 2011 International conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, Dalian, 2011, pp. 9–15.

[9] W. Wang, S. De, R. Toenjes, E. Reetz, and K. Moessner, “A Comprehensive Ontology for Knowledge Representation in the Internet of Things,” in 2012 IEEE 11th International Conference onTrust, Security and Privacy in Computing and Communications (TrustCom), Liverpool, 2012, pp. 1793–1798.

[10] J. He, Y. Zhang, G. Huang, and J. Cao, “A smart web service based on the context of things,” ACM Trans. Internet Technol., vol. 11, no. 3, pp. 1–23, Jan. 2012.

[11] L. Spalazzi, G. Taccari, and A. Bernardini, “An Internet of Things ontology for earthquake emergency evaluation and response,” 2014, pp. 528–534.

[12] Noy NF, McGuinness DL. Ontology Development 101: A Guide to Creating Your First Ontology [Internet]. Stanford University; 2001 [cited 2013 Jun 30]. Available from: http://www-ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness.pdf

Ovaj je rad financirala Hrvatska zaklada za znanost projektom IP-2014-09-3877.
This work has been fully supported by the Croatian Science Foundation under the project IP-2014-09-3877.