Web of Things Semantic Interoperability in Smart Buildings

Laadhar, Amir; Dongo, Junior; Enevoldsen, Søren; Revaz, Frédéric; Gabioud, Dominique; Pedersen, Torben Bach; Meyer, Martin; Nielsen, Brian; Thomsen, Christian

Published in:
Procedia Computer Science

DOI (link to publication from Publisher):
10.1016/j.procs.2022.09.155

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 207 (2022) 997–1006

Procedia Computer Science

www.elsevier.com/locate/procedia

26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2022)

Web of Things Semantic Interoperability in Smart Buildings

Amir Laadhar[a,*], Junior Dongo[a], Søren Enevoldsen[a], Frédéric Revaz[b], Dominique Gabioud[b], Torben Bach Pedersen[a], Martin Meyer[b], Brian Nielsen[a], Christian Thomsen[a]

[a]*Department of Computer Science, Aalborg University, Aalborg, Denmark.*
[b]*Institute of Sustainable Energy, School of Engineering, University of Applied Sciences Western Switzerland Valais, Sion, Switzerland.*

Abstract

Buildings are the largest energy consumers in Europe and are responsible for approximately 40% of EU energy consumption and 36% of the greenhouse gas emissions in Europe. Two-thirds of the building consumption is for residential buildings. To achieve energy efficiency, buildings are being integrated with IoT devices through the use of smart IoT services. For instance, a smart space heating service reduces energy consumption by dynamically heating apartments based on indoor and outdoor temperatures. The W3C recommends the use of the Web of Things (WoT) standard to enable IoT interoperability on the Web. However, in the context of a smart building, the ability to search and discover building metadata and IoT devices available in the WoT ecosystems remains a challenge due to the limitation of the current WoT Discovery, which only includes a directory containing only IoT devices metadata without including building metadata. Integrating the IoT device's metadata with building metadata in the same directory can provide better discovery capabilities to the IoT services providers. In this paper, we integrate building metadata into the W3C WoT Discovery through the construction of a Building Description JSON-LD file. This Building Description is integrated into the W3C WoT Discovery and based on the domOS Common Ontology (dCO) to achieve semantic interoperability in smart residential buildings for the WoT IoT ecosystem within the Horizon 2020 domOS project. This integration results in a Thing and Building Description Directory. dCO integrates the SAREF core ontology with the Thing Description ontology, devices, and building metadata. We have implemented and validated the WoT discovery on top of a WoT Thing and Building Description Directory. The WoT Discovery implementation is also made available for the WoT community.

Keywords: Web of Things, Internet of Things, Web of Things Discovery, Semantic Interoperability, Semantic Web, Ontology ;

1. Introduction

According to the 2021 energy efficiency report of the European Union (EU), 75% of the existing buildings in the EU have been assessed as energy-inefficient. Two-thirds of the building energy consumption is for residential

* Corresponding author.
E-mail address: amir.laadhar@icloud.com

10.1016/j.procs.2022.09.155

buildings. Therefore, the domOS project focuses on residential buildings. Between 2000-2019, space heating is the dominant household consumption with 66% of the total consumption[1]. Driven by the availability of inexpensive IoT devices, buildings are being integrated with a variety of IoT devices to achieve energy efficiency through the use of smart IoT services. IoT services collect real-time data and communicate with different IoT sensors and actuators to effectively reduce energy consumption while ensuring household comfort. For instance, a smart space heating service integrates different IoT devices in a building. This service reduces energy consumption by dynamically controlling heating based on intelligent algorithms and different sensors such as room temperature and humidity and turns on the heater if the room temperature is below a given threshold. Therefore, the total heating time of a building can be reduced without affecting the household comfort while reducing energy consumption.

The Horizon 2020 domOS project[2] specifies an open and secure multi-service IoT ecosystem for smart buildings based on the WoT and deploys it implements in 5 demonstrator sites across Europe. One of the main challenges in the IoT landscape is achieving semantic interoperability between IoT data producers and IoT data consumers. To ensure semantic interoperability in the WoT, ontologies allow humans and machines to have a common understanding of the meaning of an IoT ecosystem through the use of a common vocabulary.

IoT consumers cannot be expected to be aware of every possible metadata of a smart building. It is necessary to rely on a semantic discovery capability that enables IoT consumers to search and discover the required metadata of a smart building. For instance, a new service provider would like to know the existing installed devices, and their energy characteristics in an apartment. W3C Web of Things (WoT) Thing Descriptions (TD) provide metadata that describes the affordances (properties, actions and events) and semantics of IoT devices. However, before a TD can be interpreted and used, it needs to be discovered. WoT Thing Description Directory provides a searchable repository of a collection of Thing Descriptions. W3C WoT Discovery specifies an API for a WoT Thing Description Directory service that provides a search mechanism for collections of WoT Thing Descriptions that can be dynamically registered. The Thing Description Directory serves with a prescribed API that allows the registration, management, and search of a database of Thing Descriptions. However, this directory has limited search capabilities, since it stores only Thing Descriptions without including the metadata of smart buildings. Metadata of a smart building can include the building space topology, the topology of energy meters, the relationships between IoT devices, the location of each device, and their ensured tasks.

To cope with the issues related to the lack of metadata in the W3C WoT Discovery, we specify and implement a Building Description Directory integrated with the Thing Description Directory in order to provide better semantic discovery capabilities. The Building Description is constructed based on the domOS Common Ontology (dCO), which integrates the Thing Description ontology, SAREF core ontology [5] and Building metadata. Therefore, IoT service providers are able to query a common directory in order to search for building metadata in the context of the WoT. The contributions of this paper are the following:

- A Building Description encoded in a JSON-LD format, which provides a powerful foundation to represent smart buildings knowledge in a machine-understandable way. The Building Description is constructed based on a schema defined by the domOS Common Ontology.

- A searchable Thing and Building Description Directory, with a prescribed API that allows, registration, management, and search of a database of Building Descriptions and Thing Description. The Building Description is integrated with Thing Descriptions in a semantic knowledge base in order to deliver the full search capabilities.

- An open-source implementation of the Thing and Building Description Directory is available to the WoT community.

[1] https://www.odyssee-mure.eu/publications/policy-brief/buildings-energy-efficiency-trends.html
[2] https://www.domos-project.eu/

2. Background Information

2.1. W3C Web of Things

The W3C Web of Things standard has been proposed to enable interoperability between the different platforms and applications for the Internet of Things. This is done by defining a WoT standard architecture [12] to describe the things (abstractions of physical or virtual entities) and to improve their usability while enabling interoperability.

The architecture is composed of four formally specified building blocks:

- The WoT Thing Description (TD) [7] is a JSON format that describes "Thing" metadata, protocols supported for interaction, properties, and the different communication interfaces available.

- The WoT Binding Templates [11] use a set of information to describe the communication protocols supported by a Thing. It enables a client to adapt to the underlying protocol the Thing can use.

- The WoT Scripting API [8] provides a convenient way to implement WoT applications that describe how to work (discover, operate, and expose) with a Thing using its TD. This component can be used to extend WoT at the gateways level to support new endpoint types.

- The WoT Security and Privacy Guidelines [15] define the general security requirements for the WoT.

The Thing Description is the fundamental building block of the WoT architecture. It is composed of: metadata to help describe the behavior of the Thing; Interaction Affordances specifying the interfaces to the Thing; data schemas that represent the data exchanged with the Thing; security configurations represent how the Thing supports privacy and security; finally, some links for the supported protocols are listed.

The WoT architecture specifies a directory service for Thing Descriptions called Thing Description Directory, which provides a Web interface for their registration and makes metadata of the things available to applications. A client application can then find the metadata needed to communicate with a thing by querying the Directory.

2.2. EU Horizon 2020 domOS project: an "Operating System" for Smart Buildings

H2020 EU domOS aims to demonstrate that increasing the energy efficiency and flexibility of buildings through the use of smart technologies is possible at a lower cost than deep renovation. To achieve this goal, domOS works on a technological and smart services axis. In the first axis, technology, an open, secure, privacy-enabled, multi-service IoT ecosystem for smart buildings is designed. If authorization is granted, multiple applications from different vendors can access building sensor data and control building setpoints, independently of the local communication network technology. The second axis, smart services, aims at developing and prototyping various smart services, mostly related to energy efficiency and flexibility. Thanks to the domOS IoT ecosystem, a smart service can interact with various smart appliances (e.g., photovoltaic inverter, electrical vehicle charging station, and heat pump), whatever their brands and models.

The domOS IoT ecosystem is based on the Web of Things and includes concepts presented in this paper: Building Descriptions, Thing Descriptions, Thing and Building Description Directory, and domOS Common Ontology. The Building Description acts as a digital nameplate for the building and gathers all relevant information about the building's topology, IoT devices, and energy system. It references Things Descriptions and describes the role of the things in the building operation.

A typical domOS scenario features the following steps: 1) Thing Descriptions for local devices and appliances are elaborated by customizing a device model-specific Thing Description template; 2) a Building Description is constructed based on the domOS Common Ontology, with references to Thing Descriptions; 3) an IoT application provider queries the Thing and Building Description directory to verify that it features the required monitoring and control points; 4) the application interacts with the house using the WoT interaction model.

2.3. W3C WoT discovery related work

Search and discovery techniques are used as a bridge between users and IoT ecosystems, but also as a middle-ware for the interconnection of any agent related to the IoT domain.

The Web of Things takes advantage of the Web architecture in the context of the Internet of Things: "things", which can be physical objects or virtual entities such as Web services, become resources that can be acted upon or queried via APIs (WoT scripting API [8]). To ensure that such “things” can be used without human intervention, they have to be formally described. To this end, the W3C standardized the Thing Description which is a specification that defines how to provide a JSON-LD representation of the affordances (properties, events and actions) of a “thing” via Web APIs. On top of this, W3C WoT Discovery provides a mechanism for discovering “thing” descriptions. These standards provide more autonomy for agents that make use of “things” connected to the Internet via Web standards. Specifically, WoT Discovery has to allow authenticated and authorized entities (and only those entities) to find WoT Thing Descriptions satisfying a set of criteria such as having certain semantics or containing certain interactions.

As the W3C WoT is a recent standard, few real-world applications and tools have been proposed so far in the literature. WoTify [10] is a W3C WoT platform that allows users to search for WoT projects, and download or contribute to shared ones. In the WoT Store approach [16], the authors propose a system for the distribution, discovery and installation of WoT applications. It includes a semantic search engine capable of discovering things through the use of a web portal capable of handling SPARQL queries. The approach of WoT Store has some parallelism with mobile application markets, like searching over a software catalog, or downloading and installing code directly from the store. Youngmin et al. [6] proposed a demo showing the possibility to query a W3C WoT sensor device from a mobile phone is sketched. Klotz et al. [9] proposed application of the W3C WoT architecture to the automotive industry; more specifically, the authors illustrate how to describe the car data with a semantic ontology, and how to make them available to external applications through the W3C WoT interaction patterns. McCool et al. [13] discussed security risks and vulnerabilities in the WoT metadata. Blank et al. [1] proposed a versioning mechanism of the Thing Descriptions and the relevance of life cycle mechanisms in Industry 4.0 scenarios, where there might be a constant change in the data structures exposed by each Thing. Serena et al. [17] proposed an ontology-driven discovery for the W3C WoT architecture, but few implementation details are provided. Moreover, the discovery process includes only Thing Descriptions, without any other type of metadata.

The W3C WoT Discovery working group made a great effort on evaluating the technology landscape relevant to the standardization initiative. There has been a great deal of development and evolution with regard to discovery in the IoT. However, semantic web technologies seem to have a minor representation in the landscape. The W3C WoT discovery proposes only a SPARQL endpoint that centralizes all thing descriptions. However, this is not sufficient when dealing with smart buildings, which includes a lot more metadata (e.g, building topology, building metadata, and relationship between IoT devices).

3. Example use cases

In this section, we present sample use cases related to semantic discovery. We consider a space heating service in a residential house located in Sion, Switzerland. The overall aim is to reduce losses in the in-building heat distribution system through a lowered temperature of the water-based internal heating distribution system. Involved things are 1) an air-to-water electrical heat pump with forward water temperature control capability; 2) per-room Smart Thermostatic Valves (STVs) monitoring temperature and enabling control of the fed-in heat. The smart space heating service algorithm calculates the optimal heat pump forward temperature and STVs position based on temperature sensors' measurement and temperature set points. Other services make use of the above-presented things but also further operate in parallel. One of the main challenges of IoT ecosystems in smart buildings is to enable IoT services to search for IoT devices and building metadata without having any prior knowledge about them. We present the following user stories related to the semantic discovery capability for the space heating service use case:

- The IoT service provider wants to know if the forward temperature of the heat distribution system can be controlled;
- The IoT service provider wants to know the list of room temperature sensors in a given apartment;
- The IoT service provider wants to know the list of rooms with a temperature above 26 degrees Celsius;
- The space heating service wants to read the sensor's values and assign the heat pump's a temperature set point.

In many cases, adding new services to an existing IoT ecosystem is a cumbersome task. Usually, new service providers do not understand the existing IoT ecosystems and the variety of its installed devices. Taking the example of a flexibility service of Electric Vehicle (EV) charging. For instance, a consumer arrives at home at 9 pm and wants to charge its EV at the lowest CO_2 emission or electricity price. Then, he wants to drive his EV tomorrow at 8 am with at least 80% battery charge. Within this 11-hour time flexibility, we can in each 1 hour time slice adjust the energy between 0 and 4KWh, giving an *energy flexibility* of 4KWh per slice. Over the whole period, the EV must be charged between 30KWh and 60KWh. The charging process can be flexible by moving the charging time. Therefore, a service provider would like to check if an IoT ecosystem contains the required IoT devices by the service. To illustrate the flexibility service use case for the semantic discovery process, we present the following user stories:

- The IoT service provider wants to know the maximum and minimum energy charging capacity of an EV;
- The IoT service provider wants to know the state of the EV charging in terms of the consumed time and KWh;
- The IoT service provider wants to know if a building contains the required IoT devices (e.g., energy meter, EV charger controller) for the EV energy flexibility service.

All these user stories, would not be possible to answer if all the metadata of IoT devices and building metadata are not stored in a central directory to allow their querying. Therefore, in this work, we propose to integrate metadata and store all of the Thing Descriptions and a Building Description in a single directory, which is a semantic knowledge base in order to ensure the WoT Discovery of all the required information in a smart building.

4. Specification of the Thing and Building Description Directory

In this section, we specify the Thing and Building Description Directory, which enables the semantic discovery for smart buildings. Therefore, we firstly present the domOS Common Ontology (dCO), which is used to construct Building Descriptions, and to annotates Thing Descriptions. This ontology integrates SAREF core ontology with Thing Description Ontology, IoT devices, Units of Measurements Ontology and building metadata. The Thing and Building Description Directory stores the RDF serialization of the Thing Descriptions and Building Descriptions in a semantic knowledge base via an API. The API allows the CRUD (Create, Read, Update, Delete) operations of Thing and Building description, and the semantic discovery of the metadata stored in the semantic knowledge base. The semantic discovery can be accomplished using a SPARQL API.

4.1. domOS Common Ontology

Conforming to the NeOn ontology engineering methodology, we identified a set of competency questions (CQ), (e.g., what are the metadata of a residential building) as part of our requirement specification document (ORSD). The ORSD is made in collaboration with our partners from the 5 domOS demonstration sites[3]. The purpose of the dCO is to integrate the WoT Thing Descriptions with smart buildings. After interviewing partners from the five demonstration sites, we have identified the following use cases of the dCO: (i) Semantic annotation of Thing Descriptions (ii) Construction of Building Descriptions (iii) Semantic validation of Thing Descriptions and Building Descriptions, and, (iv) Semantic Discovery of metadata from a Semantic Knowledge Base.

dCO is based on a modular design to enhance the maintainability of the ontology. More detailed graphical representations of the modules are provided in the dCO ontology documentation[4]. The reused ontologies and their prefixes are mentioned in Figure 1. In the following, we state the dCO modules:

dCO core module: this module reuses the SAREF core ontology and allows the integration of the other modules to ensure semantic interoperability of the WoT ecosystems in smart buildings. This module is integrated with the thing description module and the device module using a subsumption relationship. The core module is integrated with the building description module and the units of measurement module respectively using the property `vicinity:isLocatedAt` and `saref:isMeasuredIn`.

The building description module contains the representation of building-related metadata and building topology. In this module, we reuse some concepts from Building Topology Ontology (BOT) [14] of the W3C Linked Building Data Group. We also define a set of metadata related to the characteristics of a building. A `saref:Device` can be located at a `bot:Space`. We define four types of spaces: `bot:Building`, `dogont:Floor`, `dco:Apartment`, and `iottaxolite:Room`. A building must contain at least one floor, which can contain at least one apartment. An apartment can contain one or more rooms. A room can be categorized into several types (e.g., kitchen, living room).

[3] https://www.domos-project.eu
[4] https://www.w3id.org/dco

Thing Description module: this module reuses and extends the Thing Description ontology [2] to model the WoT Thing Description interaction affordances, i.e., properties, actions and events. This module integrates the W3C WoT Things Descriptions ontology with the core module.

Device module: this module represents the set of devices in an IoT ecosystem. Devices are structured in categories (subclasses), that reflect different types of devices (e.g, `saref:Meter`).

Units of Measurements module: this module aims to represent all the units of measurement of the IoT devices. This module reuses the Units of Measurement Ontology. An `saref:Measurement` is a measurement using a unit of measurement (`saref:unitOfMeasurement`), which is structured in categories that reflect the different types of unit of measurements (e.g, `om:energyUnit`).

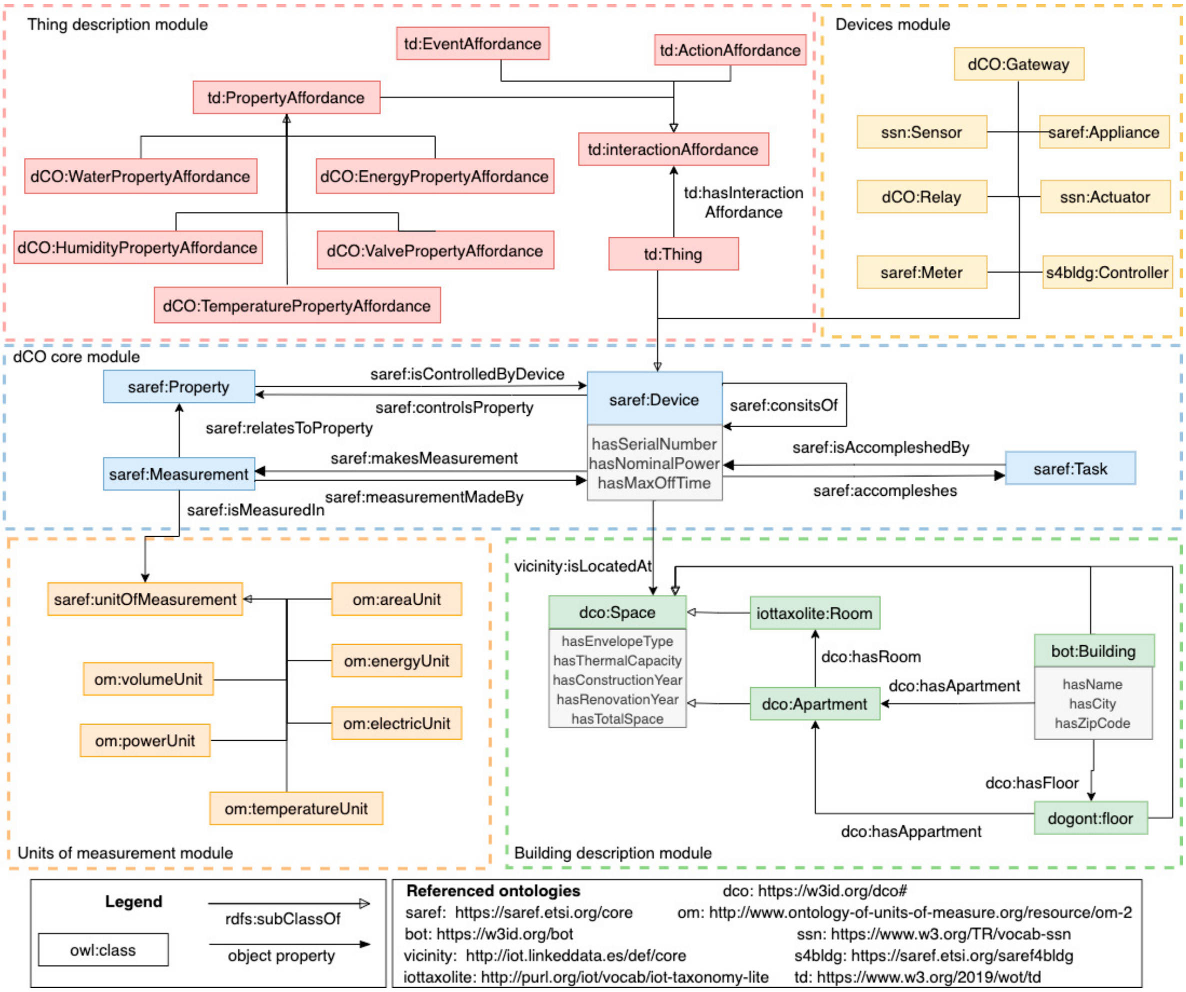

Fig. 1. dCO design overview

4.2. *Building Description Specification*

We have designed the Building Description based on the identified requirements related to the Building Description and documented them as competency questions in the ORSD. The Building Description features the following elements:

- Metadata of building spaces (i.e., building, apartments, rooms), such as the thermal capacity mega watt-hour per square meter, the construction year, and the total space in a square meter. Each space can include one or more IoT devices.

- Buildings topology defines the structure of a building. The relationships between spaces, i.e., a floor can contain one or more apartments.

- Tasks: is the goal which a device or a set of devices is designed to accomplish, for example, a heat pump, a water boiler, and a water temperature sensor accomplish together the task of water heating. A space can be exposed to one or more tasks. For instance, a living room can be exposed to a space heating task.

- Devices metadata (i.e., hardware revision, nominal power, maximum on-time, power source). The relationships between devices (e.g, the heat pump has a heat pump relay, the water temperature sensor measures the water temperature of a boiler, the power meter sensor measures the power of the heat pump)

In Figure 2, we present the ontological schema of the Building Description. The concepts of the schema should be used to construct the Building Description JSON-LD file. Not all the concepts must be included in the Building Description. This depends on the use case. For instance, a building description can be made without including rooms. An IoT device can be simply located in an apartment without indicating the exact room.

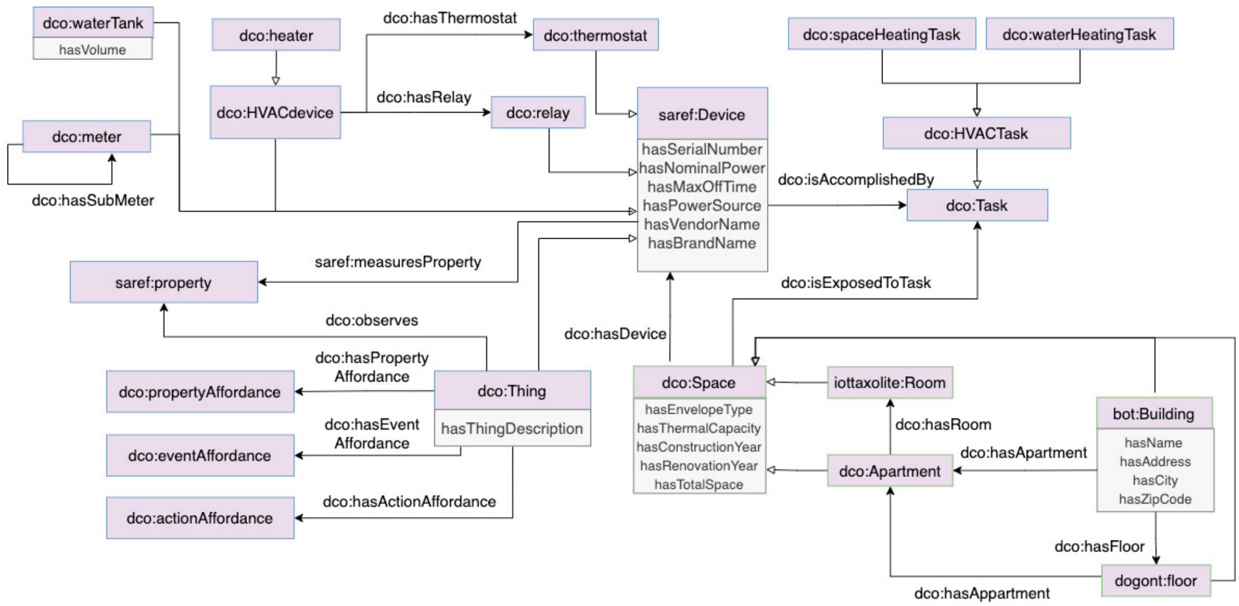

Fig. 2. Building Description Design

In Figure 3, we present a JSON-LD example of a Building Description. The illustrated example is for a single-family house, annotated using `dco:singleFamilyHouse`. A Building Description can also represent a multi-storey building. A building description has four main components: specific building metadata (line 1 to 13), building topology (line 15 to 30) indicating the device location in each space, the building tasks (line 32 to 38), and the metadata of each device (line 40 to 59). The building metadata component includes general information about the building, such as its address, total space, the existing floors, and the renovation year. The building topology component presents the structure of spaces in a building. It begins by presenting the existing floors, the spaces (e.g., apartments, rooms) of each floor, the set of devices located in each space and the exposed tasks of each space. A space (`dco:space`) may by exposed of a `dco:task` through the `dco:isExposedToTask` relation. For example, the ground floor is exposed to the space heating task (`dco:spaceHeatingTask`) and the water heating task (`dco:waterHeatingTask`). The installed sensing and actuating capabilities are captured by placing things in the defined building spaces. This is accomplished through the use of the property (`dco:hasDevice`). For example, the living room (line 25), has an ambient temperature sensor (Line 27). The Tasks component (Line 32 to 38) indicates all the tasks and the associated devices for each task. This is represented using "`dco:isAccomplishedBy`". The Devices component (Line 40 to 59) includes all the devices and their metadata. A device can be a WoT Thing (e.g, `waterTemperatureSensor01`) or a device not associated with a WoT Thing (e.g., heatPump01). Devices types are defined in the device hierarchy of the ontology (e.g. of types `dco:supplyTemperatureSensor`, `dco:heatPumpAppliance`). Each WoT device has a reference (`dco:hasThingDescription`) to the actual WoT thing description that implements the instance of the device type. This decouples the abstract, semantically well-defined device types from the actual devices that supply the data.


```
{ "@context":{"dco": "https://w3id.org/dco#"},
  "@id": "singleFamiliyHouse01",
  "@type": "dco:SingleFamiliyHouse",
  "dco:hasName": "Sion building",
  "dco:hasEnvelopeType": "dco:lightIsolation",
  "dco:hasThermalCapacity": [{"@value": "11234", "dco:unit":"dco:megaWattHourPerSquareMeter"}],
  "dco:hasConstructionYear": "1972",
  "dco:hasCity": "Sion",
  "dco:hasZipCode": "9220",
  "dco:hasAddress": "55 Route de la liberte",
  "dco:hasRenovationYear": "1989",
  "dco:hasTotalSpace": "846",
  "dco:hasFloor": "groundFloor01",

  "Floors":{
    "groundFloor01":{
     "@type": "dco:groundFloor",
      "dco:hasSpace": [
        {"@id": "Kitchen01",
         "@type": "dco:Kitchen"},
        {"@id": "ControlRoom01",
         "@type": "dco:ControlRoom",
         "dco:hasDevice":["heatPump01","waterBoiler01","smartMeter01","powerMeter01",
         "gateway01","heatPumpRelay01","waterTemperatureSensor01"]},
        {"@id": "LivingRoom01",
         "@type": "dco:LivingRoom",
         "dco:hasDevice":"ambientTemperatureSensor01"},
        {"@id": "Bedroom01",
         "@type": "dco:Bedroom"}],
      "dco:isExposedToTask":["spaceHeatingTask01","waterHeatingTask01"]}},

  "Tasks": {
    "spaceHeatingTask01": {
     "@type": "dco:spaceHeatingTask",
     "dco:isAccomplishedBy":["heatPump01","heatPumpRelay01","ambientTemperatureSensor01","powerMeter01"]},
    "waterHeatingTask01": {
     "@type": "dco:waterHeatingTask",
     "dco:isAccomplishedBy":["heatPump01","heatPumpRelay01","waterTemperatureSensor01"]}},

  "Devices": {
    "heatPump01": {
        "@type": "dco:heatPump",
        "dco:hasHeatPumpRelay":"heatPumpRelay01",
        "dco:hasObservablePropery":[{"HeatPumpPower01":{"@type":"dco:powerObservableProperty"}}],
        "dco:hasNominalPower": [{"@value": "3500","dco:unit": "dco:watt"}],
        "dco:hasMaximumOffTime": [{"@value": "0.5","dco:unit": "dco:hour"}],
        "dco:hasMinimumOnTime": [{"@value": "8","dco:unit": "dco:hour"}]},

    "waterBoiler01": {
       "@type": "dco:waterBoiler",
       "dco:hasObservablePropery":[{"waterTemperature01":{"@type":"dco:waterTemperatureProperty"}}],
       "dco:hasVolume": [{"@value": "0.8","dco:unit": "dco:squareMeter"}]},

    "waterTemperatureSensor01": {
       "@type": "dco:waterTemperatureSensor",
       "waterTemperaturePropertyAffordance01":{"@type":"dco:waterTemperaturePropertyAffordance","dco:unit": "dco:Celsius"},
       "dco:observes":[{"waterTemperature01":{"@type":"dco:waterTemperatureProperty"}}],
       "dco:hasThingDescription":"https://domos.oiken.ch/tdd/waterTemperatureSensorID"}}
}
```


Fig. 3. Building Description Example

4.3. Thing and Building Description Directory Specification

In addition to storing and querying Thing Descriptions we also want the same capability for Building Descriptions. Simply storing the documents in a document database would not be sufficient to make use of the semantic information embedded within. All WoT Thing Descriptions are JSON-LD documents with a context embedded that maps the contents of the document to the RDF data model. By also storing the RDF representation as triples in a triple store we gain the ability to use semantic querying technologies such as SPARQL. Because the description documents are augmented with the annotations from the domOS Common Ontology it affords the ability to query in terms of the ontology to answer question such as: what is the maximum and minimum energy charging capacity of an EV?

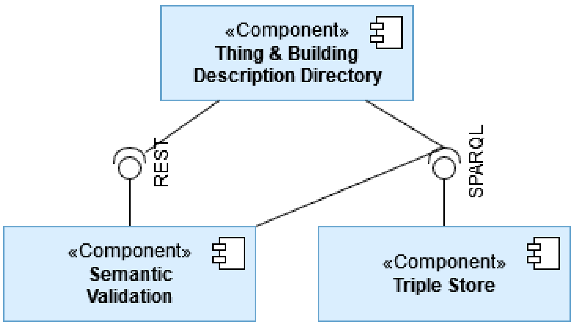

Fig. 4. Organization of the components for Thing Discovery

In Figure 4, we propose the the structure of the TBDD (Thing & Building Description Directory). The TBDD stores WoT Thing Descriptions (TDs) and the Building Descriptions (BDs). For semantic search the triple store stores the RDF triples serialization of the Thing and Building Descriptions. The Semantic Validation component verifies that a proposed document or change to be introduced contains is compliant with dCO and also there are no inconsistencies in the semantic representation as a whole. This can be verified using Shapes Constraint Language (SHACL).

Endpoint	Method	Description
/td	GET	Retrieve all TDs.
/bd	GET	Retrieve all BDs.
/td/{id}	GET	Retrieve the TD with the given id.
/td/{id}	PUT	Insert or replace the TD with given {id}. The id must match the one inside the TD
/td	POST	Inserts a TD with no id. The resulting id is in the response.
/td/validate	POST	Returns a list of validation errors (possible 0) for the TD without inserting or updating it.
/search/sparql?query={query}	GET	Runs a SPARQL discovery query against the stored TDs and BDs
/bd/{id}	GET	Retrieves the Building Description with the given id.
/bd	POST	Inserts a BD with no id. The resulting id is in the response.
/bd/{id}	PUT	Replaces the Building Description with the given id.
/bd/validate	POST	Returns a list of validation errors (possible 0) for the BD without inserting or updating it.

Fig. 5. HTTP endpoints for the Things & Building Description Directory

Table 5 shows the endpoints available for interacting with the TBDD component. The endpoints prefixed with '/td' are used for the Things Description part of the component. The operations, while a subset of the entire API (except /td/validate), are all compliant with WoT Thing Description Directory API and the endpoints themselves could be described in a Thing Description of type DirectoryDescription. The API for interfacing with the building description is prefixed with '/bd'. The semantic validation component has a simple API accepting either a Thing Description or Building Description and validates that the proposed description satisfy all ontological constraints defined by SHACL.

5. Implementation of the Thing and Building Description Directory

This section presents the implementation of the Thing and Building Description Directory specification. We also present technical information about systems and software design decisions. We have implemented the Thing and Building Description Directory in 5 demonstrator sites across Europe of the domOS project. This directory allows CRUD operations of Thing Descriptions and Building Descriptions, their syntactic validation, semantic validation, and semantic discovery using an API interface. We made the implementation of the Thing and Building Description Directory and the WoT Discovery available for the WoT community on GitHub (https://github.com/mistersound/tdd-bdd-api). We implemented the directory and the WoT Discovery and made it available using a Swagger interface (https://mistersound.github.io/TBDD/). Things Description and Building Descriptions are synchronously stored, updated, and deleted from a document database and a triple store. The document database (MongoDB) allows the CRUD operation of the JSON-LD Thing Descriptions. The scalable RDF triple store, Apache Jena Fuseki, stores the RDF transformation of the JSON-LD Things Description and Building Descriptions in order to allow semantic search via a SPARQL endpoint. We have implemented the Thing and Building Description Directory using Javascript (Nodejs). The TDD part of the directory is compliant with the WoT Discovery [4]. Regarding The syntactic validation, we have created JSON-LD schemes to validate the syntax Thing Description and Building Descriptions. Regarding the semantic validation of the description files, they are serialized into RDF, then validated the RDF against the ontology schema using SHACL. We have automatically generated SHACL shapes using ASTERA [3]. We have tested and validated the implementation in an agile development process in collaboration with our stakeholders. This has been done based on input from the 5 demo sites across Europe which are part of the domOS project.

6. Conclusion and Future Work

IoT services are employed for improving energy efficiency in smart buildings. These IoT services reduce the energy consumption of buildings without compromising household comfort. However, the semantic heterogeneity of IoT ecosystems presents a major bottleneck to implementing IoT services. As a result, IoT developers cannot be aware of the existing available devices and building metadata on the Web of Things. It is necessary to rely on a semantic discovery capability that enables IoT consumers to search and discover the required IoT devices and metadata of a smart building. However, the current W3C WoT Discovery does not integrate building metadata and recommends only the use of a Thing Description Directory. This directory integrates only Thing Descriptions, which can contain limited semantic annotations and does not contextualize IoT devices in smart buildings. Consequently, we specify a Thing and Building Description Directory, in order to leverage an advanced WoT Discovery and semantic queries in smart buildings. This Directory stores and integrates Things Descriptions and Building Descriptions and exposes them to IoT consumers via semantic queries. Building Descriptions are metadata files constructed based on the domOS Common Ontology (dco). This ontology integrates Thing Description ontology, SAREF core ontology, Units of Measurement Ontology, IoT devices metadata, and building metadata.

In future work, we also plan to expand the Building Description in order to support more metadata (e.g., energy flexibility of IoT devices). The expansion of the Building Description should be eventually supported by the domOS Common Ontology, which should be extended as well. We also plan to build building a Web of Things knowledge graph integrating data produced from IoT devices with metadata stored in the Thing and Building description directory. This knowledge graph will allow more advanced semantic queries of IoT data in the context of smart buildings. This allows the total decoupling of the Web Of Things ecosystems from the actual IoT devices infrastructure. The knowledge graph should include access and identity control management in order to comply with GDPR regulations. Moreover, the knowledge graph shall be employed as a data catalog in order to integrate in order to provide an inventory of IoT data combined with metadata management capabilities such as metadata curation.

Acknowledgements

Research funded by the European Union through the Horizon 2020 domOS project (grant agreement 894240).

References

[1] Blank, M., Lahbaiel, H., Kaebisch, S., Kosch, H., 2018. Role models and lifecycles in iot and their impact on the w3c wot thing description, in: Proceedings of the 8th International Conference on the Internet of Things, pp. 1–4.
[2] Charpenay, V., Käbisch, S., 2020. On modeling the physical world as a collection of things: The w3c thing description ontology, in: European Semantic Web Conference, Springer. pp. 599–615.
[3] Cimmino, A., Fernández-Izquierdo, A., García-Castro, R., 2020. Astrea: automatic generation of shacl shapes from ontologies, in: European Semantic Web Conference, Springer. pp. 497–513.
[4] Cimmino, A., McCool, M., Tavakolizadeh, F., Toumura, K., 2021. Web of Things (WoT) Discovery. Technical Report. W3C Working Group.
[5] Daniele, L., . SAREF4ENER: An Extension of SAREF for the Energy Domain Created in Collaboration with Energy@ Home and EEBus Associations. Technical Report. Technical Report.
[6] Ji, Y., Ok, K., Choi, W.S., 2018. Web of things based iot standard interworking test case: demo abstract, in: Proceedings of the 5th Conference on Systems for Built Environments, pp. 182–183.
[7] Käbisch, S., Kamiya, T., McCool, M., Charpenay, V., 2019. Web of things (wot) thing description. candidate recommendation. w3c.
[8] Kis, Z., Peintner, D., Aguzzi, C., Hund, J., Nimura, K., 2020. Web of Things (WoT) Scripting API. Technical Report. W3C Working Group.
[9] Klotz, B., Datta, S.K., Wilms, D., Troncy, R., Bonnet, C., 2018. A car as a semantic web thing: Motivation and demonstration, in: 2018 Global Internet of Things Summit (GIoTS), IEEE. pp. 1–6.
[10] Korkan, E., Hassine, H.B., Schlott, V.E., Käbisch, S., Steinhorst, S., 2019. Wotify: A platform to bring web of things to your devices .
[11] Koster, M., Korkan, E., 2019. Web of things (wot) binding templates. candidate recommendation. w3c.
[12] Kovatsch, M., Matsukura, R., Lagally, M., Kawaguchi, T., Toumura, K., Kajimoto, K., 2019. Web of things (wot) architecture. W3C Recommendation .
[13] McCool, M., Reshetova, E., 2018. Distributed security risks and opportunities in the w3c web of things, in: Workshop on Decentralized IoT Security and Standards (DISS).
[14] Rasmussen, M.H., Lefrançois, M., Schneider, G.F., Pauwels, P., 2021. Bot: The building topology ontology of the w3c linked building data group. Semantic Web , 1–19.
[15] Reshetova, E., McCool, M., 2019. Web of things (wot) security and privacy guidelines. candidate recommendation. w3c.
[16] Sciullo, L., Gigli, L., Trotta, A., Di Felice, M., 2020. Wot store:managing resources and applications on the web of things. Internet of Things .
[17] Serena, F., Poveda-Villalón, M., García-Castro, R., 2017. Semantic discovery in the web of things, in: International Conference on Web Engineering, Springer. pp. 19–31.