

DogOnt as a viable seed for semantic modeling of AEC/FM

Dario Bonino [a,*], Luigi De Russis [b]
[a] *Istituto Superiore Mario Boella, Torino, Italy*
E-mail: bonino@ismb.it
[b] *Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino, Italy*
E-mail: luigi.derussis@polito.it

Abstract. Energy consumption and performance assessment of Smart Cities must consider different levels, various sub-domains, and several stakeholders. A comprehensive energy profile of a city, in fact, should work at the city, district, and building levels. At the same time and for each level, it should take into account both electrical and thermal consumptions, and gather these information from a plethora of different stakeholders (i.e., citizens, utilities, policy makers, and energy providers) and sensors. Current modeling approaches for this context address each level and domain separately, thus preventing a structured and comprehensive approach to a unified energy representation. Moreover, current approaches make difficult to keep the consistency between the energetic data through levels, sub-domains and across stakeholders. Starting from an analysis of ontologies at the state-of-the-art, this paper shows how DogOnt can be used as a foundation towards a shared and unified model for such a context. DogOnt was firstly developed in 2008 and withstand over 8 years of usage without major failures and shortcomings. We discuss successful design choices and adaptations, which kept the model up-to-date and increasingly adopted in domains ranging from home automation to energy representation in Smart Cities.

Keywords: Built Environment, Ontology, Smart City, AEC/FM, Energy Modeling

1. Introduction

Energy consumption and performance assessment of Smart Cities must consider different levels, various sub-domains, and several stakeholders. A comprehensive energy profile of a city, in fact, should work at the city, district, and building levels. At the same time and for each level, it should take into account both electrical and thermal consumptions, and gather these information from a plethora of different stakeholders (i.e., citizens, utilities, policy makers, and energy providers) and heterogeneous sensors.

In such a context, intelligent, and in particular, semantic-based approaches can be seen as viable solutions to extract sense from the vast sea of information made available by the large number of sensors spread all over the city, at different levels, and involving various stakeholders. Several research groups and companies are working on techniques deriving from the Semantic Web and the Artificial Intelligence to address the modeling of so many different aspects, be it at the application, sensing, or device level. Among the available initiatives, the most renowned encompass the Linked Open Data (LOD) initiative, acting at the application level, the Semantic Sensor Web initiative, which aims at addressing, at least partially, the diversity of sensors and sensors data, and the Semantic Big Data research field aiming at tackling the data cardinality and heterogeneity issue.

Linked Open Data (LOD) provides machine understandable, shared and open semantics for representing a wide set of knowledge domains in the world. Rather than focusing on a single, rigid and practically not-scalable representation model, the LOD approach in-

*Corresponding author. E-mail: bonino@ismb.it

tegrates more than 295 datasets[1] with over 31 billions of triples representing real data, from personal e-mail contacts to world nations, from medical topics to plane parts. The Semantic Sensor Web (SSW) specifically focuses on sensing networks, and provides means for modeling sensor devices (and their capabilities), systems and processes. One of the most important results of the SSW initiative is the Semantic Sensor Network (SSN) ontology, defined by the W3C SSN XG group which was active from 2009 to 2011 [1]. Finally, the Semantic Big Data initiative, aims at empowering big data solutions (e.g., Complex Event Processing) with semantic technologies such as ontologies, definitions and behavior (inference) rules. This allows to transform "raw" data events into meaningful information conforming to a formal semantics, which, in turn, supports better understanding of situations (states) by machines (agents), better understanding of relationships between events and declarative processing of events, and reaction to situations (event patterns).

Current modeling approaches and initiatives, however, are too general for the energy context (e.g., SSW) or too specific, since they aim at modeling each level and domain separately. In this way, a structured and comprehensive approach to a unified energy representation is not possible and, similarly, it is difficult to maintain the data consistency between the energetic information through levels and across sub-domains.

This paper builds upon the motivations sustaining semantics as a viable solution to effectively tackling the energy domain in Smart Cities with a unified model. Starting from an analysis of ontologies at the state-of-the-art, the paper discusses an ontology representation, DogOnt, that was firstly published 8 years ago [2] and was initially designed, and developed, to tackle interoperability issues in home automation networks. In the past eight years, it evolved to tackle representation issues emerging from residential, building, and factory automation solutions. Lately, it included primitives for dealing with distributed networks of sensors deployed as part of smart buildings. Nowadays, DogOnt empowers several research projects needing uniform, semantic access to environment sensors and actuators and successfully supports abstraction of several standards including both Internet of Things (e.g., ZigBee) and non-IoT (e.g., Modbus) technologies. We claim and demonstrate that DogOnt can be used as a foundation towards a shared and unified model for the energy modeling in Smart Cities.

The remainder of this paper is organized as follows: Section 2 provides an up-to-date overview of the current modeling panorama for energy consumptions and assessment in Smart City settings. Section 3 introduces the DogOnt model in its current form, by discussing the foundations and showing practical modeling examples, while Section 4 motivates and illustrates why DogOnt can be used as a unified model for this context. Finally, Section 5 provides final remarks and discusses the foreseen evolutions in the next 5 years.

2. Overview on AEC/FM modeling

Energy performance assessment and representation demands for models able to deal with an increasing variety of ground truth data, generated through heterogeneous monitoring networks and devices. The emergence of IoT approaches to smart cities is stressing the importance of a uniform, machine understandable, representation of the energy qualities of devices, rooms, buildings and, by extension, of districts and entire cities. This need is currently acknowledged by several research efforts, both industrial and academic, which aim at building domain ontologies to model energy consumption and performance. In the energy domain, ontologies are employed to define shared and common inter-language for performance evaluation, energy rating, device consumption profiling, etc. Approaches present in the literature, typically, address the energy domain by splitting the analysis along different forms of energy, i.e., electrical and thermal. On the one hand, this division permits to tackle the specificity of the single energy form and the related engineering domains. On the other hand, it prevents a structured and comprehensive approach to energy representation, at higher levels of detail, like at the district level.

2.1. Electrical sub-domain

Electric energy consumption is one of the most important aspects modeled in the smart environments (e.g., home and building) domain. Such an importance is related to the amount of "saving" that can be achieved by considering energy management as fundamental part of home and building automation. Approaches for modeling energy consumption in smart environments mainly address the problem under two complimentary point of views. The first aims at model-

[1] Such a figure refers to 2011, with the number of datasets steadily increasing in the last years.

ing instantaneous consumption, i.e., consumption levels associated to specific, observable states of devices and appliances. The last, instead, considers the overall consumption "profile" of a given electric device, i.e., the sequence of consumption levels associated to a complete "working" cycle.

As an example, consider a washing machine. The first approach finely models the machine consumption when spinning, heating water, drying clothes, etc. while inferring the current consumption according to the machine state. The second, instead, considers complete washing cycles (e.g., delicate washing) and models the energy consumption trend with respect to time, often in discrete steps.

PowerOnt [3] follows the first approach and provides a lightweight ontology that models consumption associated to specific states of devices. A rather coarse, yet modular, approach is used for defining three levels of consumption for each state, with increasing level of details. States are associated with a *typical consumption* (in Watt) which is derived from catalogs of device categories, e.g., "A class" fridges. Such a typical consumption can be better specified if the *nominal consumption* rate is available for the specific state. Finally, the model provides means to model the *actual consumption* of the device, in a given state, extracted through direct metering. No notion of time is included in the model, and no direct/explicit support to thermal energy is provided. However, the model is general enough to represent both thermal and electric energy, with a little extension.

The challenge of representing electric device consumption has been tackled in several initiatives driven by home automation standardization bodies. Among the others, the Energy@Home consortium[2], which was involved in the definition of the ZigBee Smart Energy [4] and Home Automation [5] specifications, tackled energy consumption modeling in terms of energy profiles, i.e., of sequences of consumption levels, which evolve in time depending on the device type/-operating cycle. Unfortunately, such profiles have not been formalized in terms of ontologies and they have only been modeled in terms of data-types associated to specific ZigBee clusters.

In the last years, the increasing need for standardization of energy consumption modeling, and representation, promoted the European initiative on Energy Using and Producing Products [6], which lead to the creation of the Smart Appliances Reference ontology (SAREF [7]), now an ETSI standard [8]. SAREF formalizes in OWL the "energy profile" concept developed in the ZigBee Alliance, thus providing a standard, machine understandable representation of energy consumption of devices, over time. Moreover, it models explicitly the observable states of devices[3] and is therefore directly linkable with PowerOnt. This offers a complete modeling of both instantaneous and temporal energy consumption. It must be noted that, although SAREF implicitly assumes that devices are "electrical" and that the associated consumption is related to the "electricity" form of energy, no formal constraints prevent modeling primitives to be exploited for representing thermal quantities. As such, SAREF can be considered a nice merger for the two sub-domains.

While SAREF tackles energy consumption modeling at the device level, ThinkHome [9] (that also exploits many of the DogOnt concepts for modeling devices) addresses energy representation with a more structured approach. In fact, it considers building information for supporting optimized control strategies striving for energy-efficient operation of smart environments. It achieved this goal by explicitly integrating data stored in Building Information Models (BIM). Both Industry Foundation Classes (IFC) concepts and Green Building XML specifications [10] are supported.

The common modeling base shared by SAREF, PowerOnt and ThinkHome, i.e., DogOnt, provides a strong hint on the viability of a unified energy modeling framework, based on ontologies, able to deal with different levels of detail from single devices to full homes and buildings, regardless of the energy form.

The latter aspect, which is worth citing in the electrical domain, regards consumption flexibility, i.e., the ability to perform temporal load switches depending on both internal (self-production) or external (active demand-response) constraints. In such a context, some attempts can be cited which tackle the flexibility challenge by exploiting a formal, ontology-based modeling. Among them, the MIRABEL project defines the FlexOffer ontology [11] and represents objects involved in energy flexibility systems and their relationships. It provides a conceptual framework where the flexibility concept is defined and set in relation with

[2] `http://www.energy-home.it`, last visited on April 05, 2017

[3] as its device modeling approach partly stems from the DogOnt ontology [2]

building information and smart grid data. FlexOffer is mainly intended as a tool for supporting IT and Energy stakeholders to handle supply and demand of energy, using a common inter-operation language. In addition, FlexOffer is partly integrated in SAREF, thus being easily reconducted to the SAREF modeling base ontology.

2.2. Thermal sub-domain

Ontologies addressing energy profiling under the thermal standpoint typically represent the temporal evolution of consumption, since instantaneous data is less relevant in environments where time constants are of the order of minutes or hours. In the thermal domain, most of the ontology-based models address energy performance evaluation in terms of multiple energy efficiency indexes, as prescribed by the European Energy Performance of Buildings Directive (EPBD), which imposes the adoption of measures for improving energy efficiency in buildings.

The Energy Efficiency Ontology (EEOnt) [12], for example, provides a semantics-rich, representation of energy data in terms of EPBD objectives, thus offering means to model buildings and energy efficiency in a unified way. Moreover, it provides tools for building energy assessment inventories, enabling the creation of formal, machine understandable and easily assessable certification schemes.

Similarly to most of the ontologies described for the electrical sub-domain, EEOnt builds upon the work done in DogOnt [2] and its extensions. Through DogOnt, appliance properties are exposed according to existing semantic models, while power consumption is modeled by introducing a specific Energy Profile ontology (i.e., PowerOnt [3]). EEOnt explicitly represents links between building components and corresponding energy efficiency indexes, which is clearly complimentary to the ThinkHome approach.

The SmartCoDE ontology model [13], instead, represents the thermal homologous of profile-based modeling of electric consumption. It provides a classification of Energy using Products (EuPs) into seven categories based on their compound temporal and energy behavior. Included categories are:(a) variable services; (b) thermal services, (c) schedulable services, (d) event-timeout services, (e) charge control, (f) complete control, and (g) custom control. Moreover, an energy management and a cost profile characterize each product. SmartCoDe mappings with SAREF exists and can be easily obtained [14].

2.3. City and district-level modeling

Systemic views of energy consumption are gaining momentum, thanks to an increasing demand for representing building energy profiles in the context of a wider district- or city-level vision. The Urban Energy Ontology (UEO)[4], elaborated in the SEMANCO project[5], among many similar initiatives, describes the domain of urban planning based on the SUMO upper-level ontology [15]. It includes concepts derived from diverse sources, and related to the domain of urban planning and energy management. UEO encompasses terms and attributes for describing regions, cities, district and buildings, energy consumption profiles and CO2 emission indicators, together with climate and socio-economic factors that influence energy consumption.

The CERISE CIM Profile for Smart Grids, i.e., the Common Information Model developed by the Cerise-SG project[6], addresses interoperability of information exchanged between smart grids, public authorities, and geographical information. The Cerise-SG project, in particular, developed semantic model transformation services bridging the gaps between modeling domains relevant to smart grids (e.g., as in Gridpedia[7]), and providing alignment and conflict resolution facilities. The Energy in Buildings Ontology[8] is another attempt to provide a systematic framework for city-level energy modeling. It provides a reference model for publishing energy performance data of public buildings in Italy, with a Linked Open Data approach. With respect to the previous models, and in addition to building-level representations, it addresses and represents energy flows incoming and outgoing from a building district.

3. DogOnt

3.1. Overview

The DogOnt ontology aims at offering a uniform, extensible model for all devices being part of a smart

[4] `http://www.semanco-tools.eu/urban-enery-ontology`, last visited on April 05, 2017

[5] `http://www.semanco-project.eu/`, last visited on April 05, 2017

[6] `http://ns.cerise-project.nl/energy/def/cim-smartgrid`, last visited on April 05, 2017

[7] an RDF/XML model for the smart grid domain, `http://gridpedia.org` (last visited on April 05, 2017)

[8] `http://www.planergy.it/file/EiBOv1.owl`, last visited on April 05, 2017

environment, no matter if at the home, building or district level. Its major focus is on device modeling, for all the aspects needed to abstract device "capabilities" from low-level idiosyncrasies and communication issues. This enables both abstract reasoning on devices, e.g., to find similar devices or to identify the most suitable output to which forward urgent notifications, and actual integration of different technologies, and paradigms. DogOnt was firstly introduced in 2008 [2] and was originally meant to represent home automation devices for interoperability support. In the past years, it underwent several reviews and amendments, and its scope was widened to include devices and technologies typically part of an indoor IoT network. If the original focus was more on modeling operational aspects enabling device control, the latest version, discussed in this paper, has moved to a more informed, modular and linked modeling approach which enables adoption of DogOnt-based representations at different abstraction layers. Device control and interoperability is still one of the pillars of the representation, but extensibility, modularity, and service-based representation of heterogeneous entities (IoT and non-IoT devices) empower the latest ontology, thus enabling modular integration and reconciliation of different specifications, e.g., the cluster-based ZigBee Home Automation model and the registry-based Modbus data representation. More attention is also devoted to the Linked Open Data initiative: the ontology is now listed in the Linked Open Vocabulary data set[9] and its connections with well-known ontologies (see Figure 1) are being improved day by day.

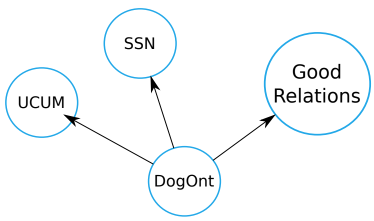

Fig. 1. DogOnt relations with well-known ontologies

From a very high-level perspective, the ontology is deployed along three main hierarchies of concepts, supported by four additional trees that better specify the knowledge encoded in the main topics. The three hierarchies are respectively rooted at *BuildingThing*, *Functionality* and *State* (Figure 2).

[9]`http://lov.okfn.org/dataset/lov/about/`, last visited on April 05, 2017

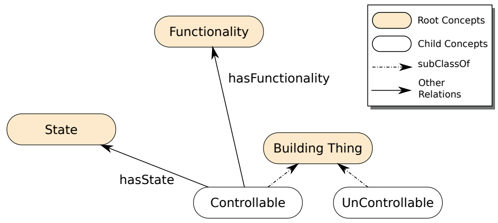

Fig. 2. The main representation pillars.

The *BuildingThing* hierarchy is one of the pillars of the DogOnt ontology and is completely devoted to the description of objects contained inside architectural spaces. These object are divided in *Controllable* and *UnControllable* entities. The former represent any device that can be somewhat controlled into a closed environment, i.e., it represents the nodes of any smart environment network. The latter, instead, represents all physical, inanimate objects contained in an indoor environment, including furniture, inner walls and partitions, etc. This hierarchy is strictly interconnected with the other two main pillars of the representation: the Functionality and the State trees of classes.

Functionality is the top concept of a class hierarchy that was originally designed to represent devices under an operational perspective. Each device was given a set of functionality which completely specified the device type, allowing - for example - classification reasoning. In the latest ontology evolution, the functionality representation has moved to a more service oriented approach[10], where each device offers a well known set of services, and sufficient conditions are provided to categorize devices as belonging to a specific class. However, modelers are free to represent entities offering an arbitrary set of services (functionality), not necessarily corresponding to actual device capabilities (e.g., virtual or high-level functionality such as energy management [14] or energy profiling).

Finally, concepts inheriting from the *State* class model the current condition of a device (Controllable, in DogOnt), using the Harel's state chart semantics [16] as reference model and allowing devices to assume multiple states at the same time, with different state values. For example, a smart microwave oven which is heating a frozen meal can be modeled as being in the "on", "defrosting", and "emitting microwaves" states at the same time. This enables higher

[10]In such a sense also the "Functionality" name is undergoing a serious review process to better reflect the new nature of modeled concepts.

flexibility in modeling, and permits to tackle different abstraction levels and different granularity depending on specific application cases.

On the formal standpoint, DogOnt is an OWL2 DL compliant ontology with $ALCHIQ(D)$[11] expressivity. It counts 896 classes and 6654 axioms. The current version (3.2.13) is released under the Apache 2.0 License and is reachable at the corresponding namespace[12] through content negotiation, as suggested by the W3C guidelines on RDF vocabulary publishing [17]. Table 1 summarizes the main ontology metrics.

Table 1
DogOnt metrics.

Metric	Value
Axioms	6654
Logical axioms count	5221
Class count	896
Object properties count	30
Data properties count	46
Individuals count	0
DL Expressivity	$ALCHIQ(D)$
SubClassOf axioms count	2595
Equivalent classes axioms count	2
Disjoint classes count	2425

As can easily be noticed, DogOnt adopts a modeling paradigm that maintains a clear separation between ontology schema and instances (0 instances in the main ontology). In such a way, environment descriptions are independent and slowly evolving knowledge (the schema) is well separated from quickly changing models (environment representations).

Subsequent paragraphs better detail the DogOnt model with respect to devices and surrounding environments.

3.2. Device modeling

Devices and sensors corresponding to physical objects, or behaving as (virtual) physical devices, are represented as subclasses of the main *Controllable* concept (equivalent to the *Device* class defined in the SSN ontology). Controllables are further specialized into *Appliances*, *HousePlants* and *NetworkComponent* (Figure 3), which respectively identify smart objects (e.g., fridges, washing machines, etc.), sensors and actuators[13], and physical layer components, i.e., devices whose main function is to guarantee physical communication of real devices (e.g., network controllers, gateways, etc.).

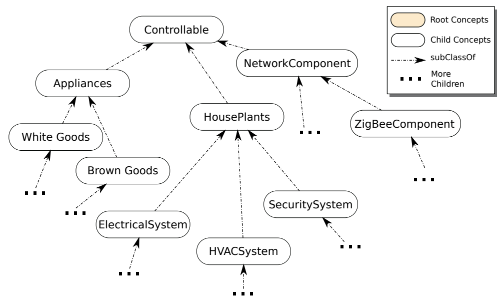

Fig. 3. Controllable subclasses.

Differently from the initial modeling approach, which was mainly descriptive and modeled device operations through single, shared instances of subclasses of the *Functionality* concept, the current ontology adopts a strong service-oriented approach where devices and operations (functionality classes) are associated by means of object properties (`dogont:hasFunctionality`). Every modeled device, in other words, is described as an entity having a (variable) set of functionality and states. While several device classes (over 400) are already described in the ontology, and their functionality predefined through `owl:someValuesFrom` restrictions (Figure 4 shows an example), modelers are free to create their own classes (and/or instances) by composing functionality and states through the `dogont:hasFunctionality` and `dogont:hasState` relations.

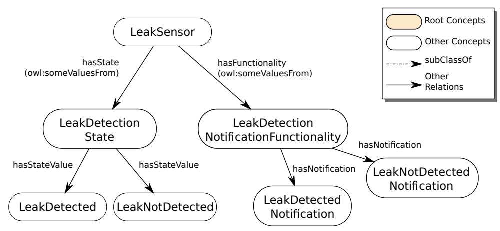

Fig. 4. An example of device class predefined in the ontology.

[11] *AL* indicates the base language allowing atomic negation, concept intersection, universal restrictions and limited existential qualifications; *C* means complex concept negation; *H* defines support for role hierarchies (subproperties); *I* provides inverse relationships; *Q* defines support for qualified cardinality restrictions and (*D*) indicates the capability to handle datatype properties and expressions.

[12] http://elite.polito.it/ontologies/dogont.owl

[13] The common ancestor concept name (*HousePlants*) is under revision as it is no more intended to represent devices belonging to house plants, only.

The set of named devices defined in DogOnt encompasses devices included in the early European Home System (EHS) taxonomy, in the ZigBee device specification for Smart Metering and Home Automation, plus several other entities typically occurring in actuation and metering infrastructures (e.g., in indoor IoT networks). Nevertheless, this structure can easily be extended to support generic device definition, by removing the indoor constraint, and by adopting a more general naming schema less bounded to typical indoor systems, e.g., by renaming the *Controllable* class to *ConnectedDevice*, and so on.

Predefined device classes, in DogOnt, are organized in the main three hierarchies reported in Figure 3, which are further subdivided in commonly used categories. *Appliance*s are split along the main white and brown goods categories, respectively referring to big appliances such as fridges, ovens, stoves and to small devices such as TVs, Hi-Fi systems, etc. *HousePlant*s are divided into sub-systems each pertaining a single, homogeneous field of application and include the electric, the Heating, Ventilating and Air Conditioning (HVAC), and the security (e.g., smoke or movement sensors) sub-systems. *NetworkComponent*s are eventually organized according to the physical network they represent, e.g., *ZigBeeComponent* for ZigBee networks, *ModbusComponent* for Modbus networks, *HueComponent* for the Philips Hue connected lighting system, and so on. While *Appliance*s and *HousePlant*s are completely independent from network specific information and can be freely adopted to abstract any physical device in terms of supported functionality and possible states, the concepts belonging to the *NetworkComponent* tree are designed to "attach" network-specific data to abstract devices (through multiple typing), thus enabling low level access to the underlying physical sensor (e.g., through a gateway software). Section 3.4 reports a complete, yet simple, modeling walkthrough to better clarify the notions introduced here.

The concepts hierarchy stemming from the *Functionality* root defines the possible services (or operations) that devices can provide. Such services are categorized on the kind of interaction they support / imply. In particular, 3 different types of interactions are considered: query, notification and control. They are modeled by the sub-trees of classes rooted at *QueryFunctionality*, *NotificationFunctionality* and *ControlFunctionality*, respectively.

Query functionality model all possible interrogations that a device could answer to. They represent the typical request-based (or polling-based) interaction between devices and applications aimed at gathering data at application-driven instants. They represent, in other words, those device services that provide data upon explicit request, e.g., to get the current power consumption from an electricity meter or to obtain the amount of cars counted by a vehicles counter sensor. Notification functionality, on the converse, represent event-driven interactions between devices and applications, i.e., they represent the ability of a device to autonomously notify new data, e.g., measures, current state, etc. Eventually, control functionality represent the actuation (and configuration) capabilities of a device. They permit to associate pre-defined set of commands (modeled by *Command* instances) to devices, thus allowing to completely model the device capabilities at an abstract, technology-independent level. For the sake of clarity, control functionality can be seen as abstract, shared interfaces that define how a device can be controlled by applications (or end-users).

Control and notification functionality are complemented by two auxiliary set of classes respectively rooted at *Command* and *Notification*, which are exploited to attach predefined set of commands (notifications) to functionality modeled in DogOnt. For instance, an *OnOffControlFunctionality* is defined as having at least one *OnCommand* and one *OffCommand* by means of suitable OWL restrictions (`owl:someValuesFrom`), as shown in Figure 5.

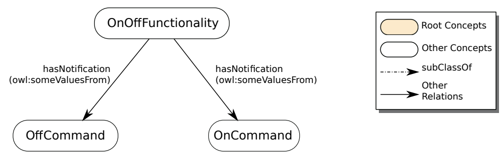

Fig. 5. Example of functionality modeling.

While concepts inheriting from the *Functionality* class model services offered by a given device, the current device state is represented by means of the hierarchy of concepts stemming from the root *State* class, and can assume several *StateValues* depending on its definition. State modeling, in DogOnt, follows the Harel's statechart semantics (Hierarchical FSMs) which provides support for complex state descriptions including parallel states, history states, clustering, and refinement. Such a semantics well adapts to complex behavior of real-world sensors and permits to represent complex devices as having multiple state values at the same time. For instance, a smart plug might be at the same time on, and measuring electric power, and so on.

It should be noted, however, that statecharts semantics does not imply exact modeling of any real device state, as this is often unfeasible. Real devices might, in fact, evolve through several, unknown, internal states which are of little interest for actual interaction in EAC/FM scenarios. Therefore, modeled states are typically a subset of actual device states, and mostly refer to observable conditions in which the device might be. Figure 6 reports an example of state modeling for color dimmable lamps.

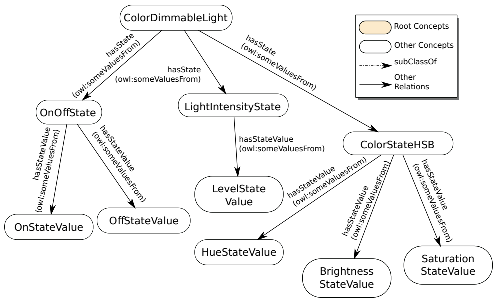

Fig. 6. State modeling for color dimmable lamps.

Both functionality and states are partitioned in descriptions of properties assuming real and discrete values, as defined in the former ontology versions. However, in the presented ontology, the formal representation of functionality and states assuming real values has been improved by accounting explicitly the associated unit of measures, thanks to a tighter integration with the well established UCUM / MUO ontologies[14].

3.3. Environment modeling

Concepts stemming from *BuildingEnvironment* and from the *UnControllable* branch of the *BuildingThing* hierarchy provide means to describe the environment hosting the modeled devices, and in particular to roughly represent the architectural spaces (i.e., rooms, etc.) containing the modeled network. Such a hierarchy has remained almost unchanged since 2008, therefore we provide a general overview of the adopted modeling paradigm, only, while interested readers may look at the original publication [2].

Environment modeling in DogOnt is rather abstract and mainly aimed at locating indoor devices at room granularity. Reflecting this general design goal the available concepts permit to represent: (a) Buildings as instances of the *Building* concepts. (b) Storeys, as part of multi-storey buildings. (c) Flats, either located on single or multiple storeys. (d) Rooms inside flats and other indoor locations (e.g. Garages) located outside flats; (e) Walls, ceilings, floors, partitions, doors and windows composing both rooms and building boundaries.

Positioning is addressed by simple containment relations, i.e., `dogont:IsIn` whereas dedicated relations are defined to represent environment composition in flats, rooms, etc. Figure 7 depicts a typical room definition.

```
<owl:NamedIndividual rdf:about="sh:bedroom">
  <rdf:type rdf:resource="&do;Bedroom"/>
  <do:contains rdf:resource="sh:Button_11_bedroom"/>
  <do:contains rdf:resource="sh:Button_down_17_sh1_bedroom"/>
  <do:contains rdf:resource="sh:Button_down_19_sh2_bedroom"/>
  <do:contains rdf:resource="sh:Button_up_16_sh1_bedroom"/>
  <do:contains rdf:resource="sh:Button_up_18_sh2_bedroom"/>
  <do:hasCeiling rdf:resource="sh:Ceiling_bedroom"/>
  <do:contains rdf:resource="sh:DoorActuator_d1_bed"/>
  <do:hasFloor rdf:resource="sh:Floor_bedroom"/>
  <do:contains rdf:resource="sh:InfraredSensor_IR1_bed"/>
  <do:contains rdf:resource="sh:Lamp_l1_bedroom"/>
  <do:contains rdf:resource="sh:MainsPowerOutlet_p3"/>
  <do:contains rdf:resource="sh:MainsPowerOutlet_p4"/>
  <do:contains rdf:resource="sh:MainsPowerOutlet_p5"/>
  <do:contains rdf:resource="sh:MainsPowerOutlet_p6"/>
  <do:contains rdf:resource="sh:ShutterActuator_sh1_bedroom"/>
  <do:contains rdf:resource="sh:ShutterActuator_sh2_bedroom"/>
  <do:contains rdf:resource="sh:SmokeSensor_smoke_bedroom"/>
  <do:contains rdf:resource="sh:Switch_allPlugsBedroom"/>
  <do:contains rdf:resource="sh:ToggleRelay_bedroom"/>
  <do:hasWall rdf:resource="sh:Wall_bed_north"/>
  <do:hasWall rdf:resource="sh:Wall_bed_west"/>
  <do:hasWall rdf:resource="sh:Wall_bedroom_bathroom"/>
  <do:hasWall rdf:resource="sh:Wall_bedroom_lobby_storage"/>
  <do:contains rdf:resource="sh:WindowActuator_w1_bedroom"/>
  <do:contains rdf:resource="sh:WindowActuator_w2_bedroom"/>
</owl:NamedIndividual>
```

Fig. 7. Example room modeling, in RDF/XML syntax.

As can easily be noticed, the modeling approach is very lightweight, however in future evolutions it might easily evolve to fully fledged environment representations exploiting Region Connection Calculus [18] and other well known, and widely recognized modeling paradigms.

3.4. Modeling Walk-through

To better clarify how DogOnt can be exploited to model devices and networks, and the services they offer, a sample modeling walk-through is reported in this section. Let assume a Smart Energy context in which a given indoor environment is hosting a sensing network to monitor energy consumption of appliances, smart home devices, sensors and actuators, e.g., for implementing Smart Grid or Demand-side Management Policies (e.g., as in the GreenCom EU project[15]). In

[14] `http://idi.fundacionctic.org/muo/`, last visited on April 04, 2017

[15] `http://www.greencom-project.eu`, last visited on April 05, 2017

such a context several smart plugs (using whichever communication technology, e.g., Z-Wave or ZigBee) are wirelessly interconnected to a network coordinator and provide information about current energy and power consumption. Despite the simplistic nature of the scenario, let consider the case in which several buildings are participating to the initiative each exploiting a different smart plug technology. In such a case, DogOnt easily supports abstraction of services and capabilities offered by involved devices and offers a common representation layer exploitable, for instance, to implement technology-independent Smart Energy Management Systems.

For the sake of simplicity, let us shrink down the problem to the representation of a single metering plug measuring the consumption of a traditional oven located in the kitchen of a given house participating in the project. As the smart-plug object is already modeled in DogOnt by means of the *MeteringPowerOutlet* concept definition, the modeling process simply consists in creating the individuals needed to represent the given plug, the oven, the room, and the house in which the plug is placed.

The modeling approach to follow, which finally leads to the result reported in Figure, follows a simple, yet general, set of representation steps:

1. identify the object to represent and the corresponding DogOnt concept (if exists);
2. model the object according to the DogOnt class definition including functionality and states, as imposed by DogOnt-defined constraints;
3. define individuals for additional functionality and states (not available in the pre-defined class specification);
4. model any object that is functional to the correct representation of the initial object (e.g., connected devices for the smart plug scenario);
5. model the environment(s) in which the objects are placed;
6. model any explicit relation between instances, e.g., the control relation between a switch and its corresponding actuator;
7. model the network-specific information allowing to interface real-devices, e.g. by adopting a gateway software.

3.4.1. Steps 1-3

The first three steps of the modeling methodology tackle the representation of the device in focus, i.e., of the sample smart plug. The earliest step in this phase involves a quick browsing of devices currently supported in DogOnt. As a general hint, in this phase, the more quick approach to browsing is “reasoning” by systems: the plug is part of a general electric system / plant, and it is something that can be controlled. The corresponding DogOnt concept, if available, should therefore be under the *Controllable* concept, possibly located in the *ElectricSystem* subtree, which in turn stems from the *HousePlants* class. By browsing the concepts immediately inheriting from the *ElectricSystem*, it is easy to notice 2 candidate subtrees, respectively rooted at *PowerDelivery* and *Meter*. Few hierarchy levels below, the two subtrees converge on the *PowerMeteringPowerOutlet* class, which perfectly matches the sample smart plug; for the sake of simplicity we assume here that the plug is only able to measure the instantaneous power absorbed by connected electrical loads.

At this point, the modeler shall concentrate on the class definition, where mandatory relations and properties are defined through suitable OWL2 restrictions (constraints). In the *PowerMeteringPowerOutlet* case, these restrictions (either locally defined or inherited through all the concept ancestors) define the plug (see Figure 8) as having:

- an *OnOffFunctionality*, i.e., the ability to be turned on and off;
- an *OnOffNotificationFunctionality*, i.e., the ability to autonomously generate events about the current activation state of the plug, e.g., to detect external control events;
- a *SinglePhaseActivePowerMeteringFunctionality*, i.e, the ability to measure currently consumed power and to be queried about current consumption;
- a *SinglePhaseActivePowerMeteringNotificationFunctionality* i.e., the ability to autonomously generate events about the current consumption of connected electrical loads;
- an *OnOffState*, modeling the ability to be providing power to connected devices, or not;
- a *SinglePhaseActivePowerMeasurementState*, representing the currently measured consumption value, and the relative unit of measure.

A suitable instance shall be created for each concept involved into such existential constraints, and the process must be recursively repeated on each of the newly created models. It must be noticed the complete absence of any technology-specific detail in the representation generated so far.

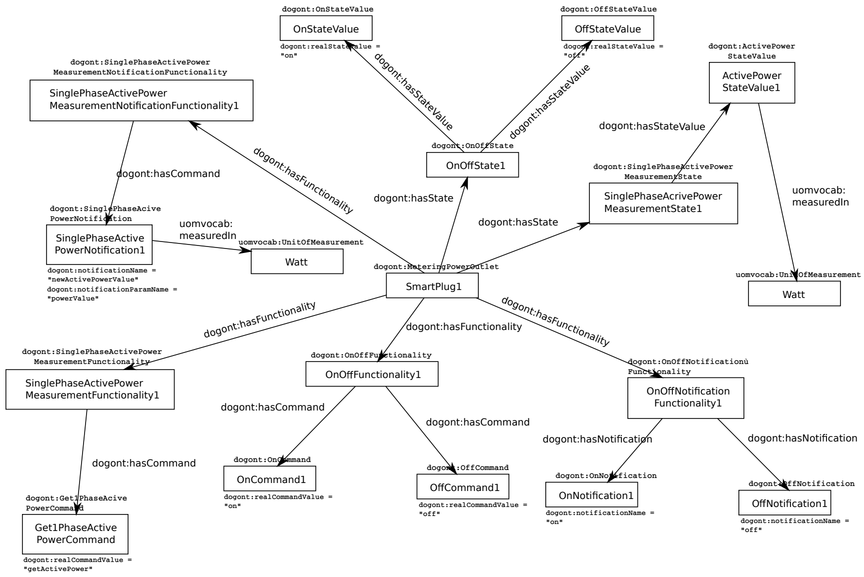

Fig. 8. Modeling approach applied to a smart plug, steps from 1-3.

3.4.2. Step 4

The fourth modeling step involves the analysis of the existing relations between the device in focus (the smart plug) and the other devices present in the same smart environment context. Relations modeling in DogOnt is quite lightweight, and mainly 3 relation families are represented: control,connection and metering. The former represents the fact that a device can control / be controlled by another device, e.g., a switch that controls a lamp. The second is specifically related to instances of classes stemming from the *PowerDelivery* concept and represents the fact that a device is connected to a power delivery object to draw the electric energy needed to provide its own functions. The latter, allows modeling the process of measuring physical quantities of interest, over a set of devices: for example, it permits to specify which set of electrical loads are monitored by a given power meter.

In this modeling step (4th) these relations are analyzed to find devices and, more in general, objects forming the context surrounding a given object, focus of the modeling process. In the sample smart plug case, a single electric load is connected to the plug: a "standing" lamp, with no intelligence on board. As the plug only powers one device, also the metering relation will involve only one instance, i.e., the same lamp. If we assume that our plug is controlled by a remote switch, e.g., located inside the same room of the plug, we finally obatin the result in Figure 9 where the plug model has been omitted to concentrate on objects modeled in this step.

3.4.3. Step 5

In this step the "built" environment in which objects are placed is represented, including all architectural features as well as all "relevant" *UnControllable* elements. In the considered example, the main involved entity is the room containing the plug, the switch and the standing lamp. The istantiation process is almost equal to the one followed in steps 1 to 3 for controllable objects and is omitted here for the sake of clarity. The same simplification is applied to graphical representation where the containing room is represented by omitting related concepts, such as walls, floor and ceiling, etc.

3.4.4. Step 6

The sixth step is the last technology-independent modeling step and provides the complete representation of concepts involved in the described modeling exercise. In such a step, previously isolated models are connected through object properties and corresponding instances are related, thus allowing to perform inferences on the represented information; e.g., to derive that since the lamp is connected to the plug, turning the plug also causes the lamp to be switched off. Figure 10 reports the full model.

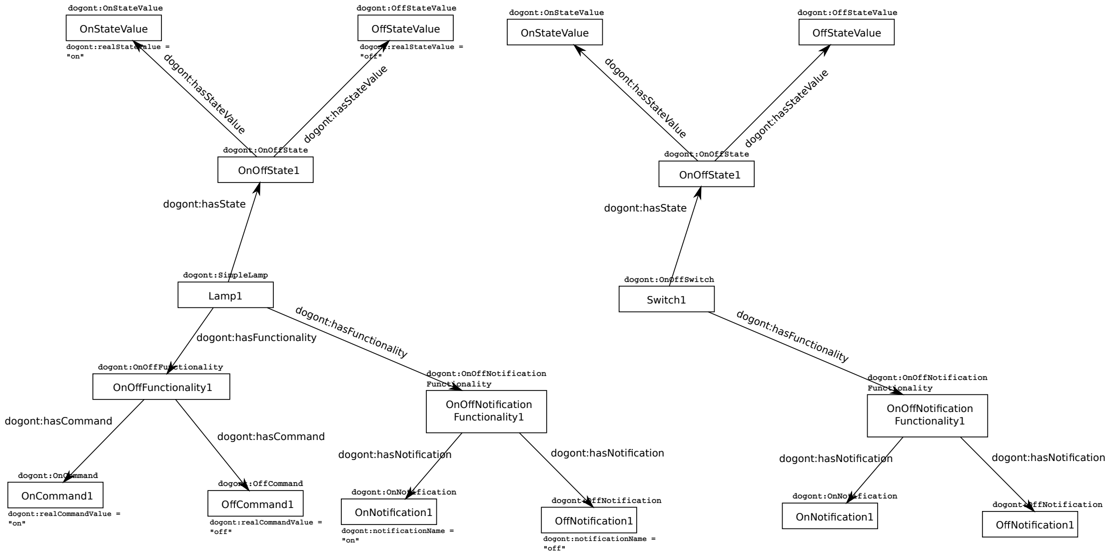

Fig. 9. Modeling approach applied to a smart plug, step 4.

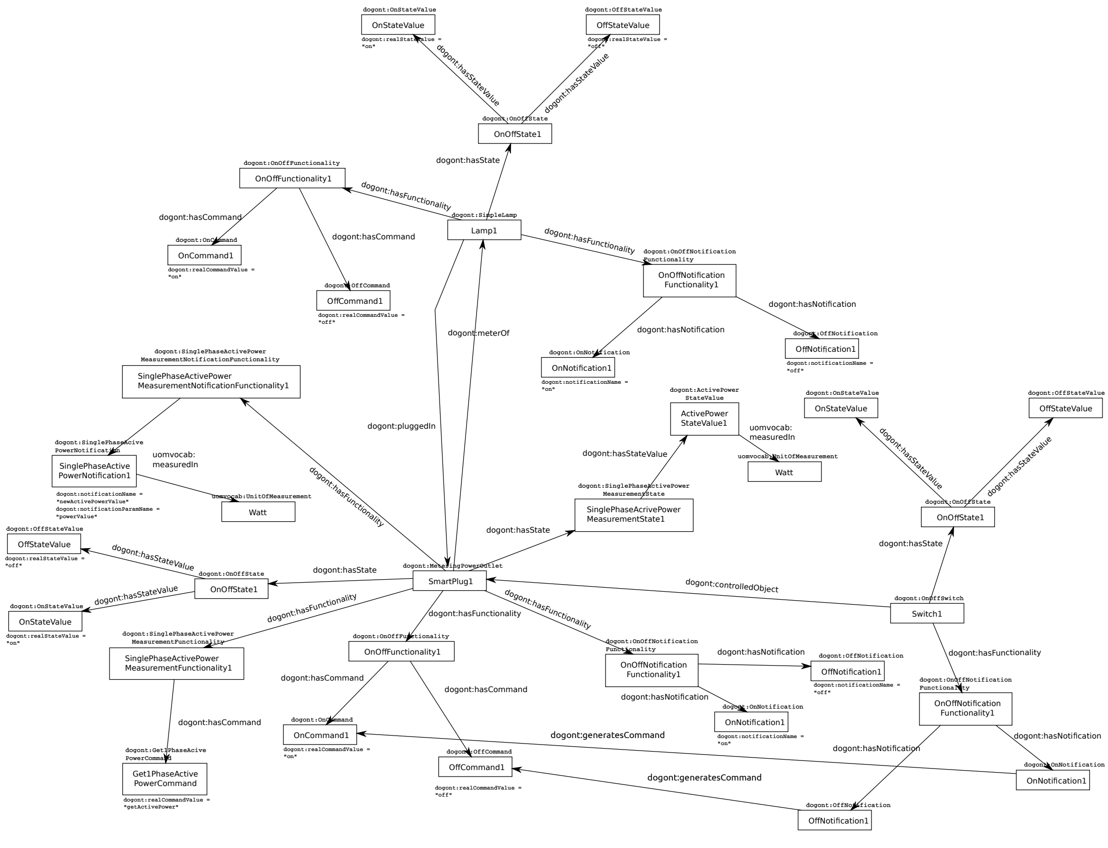

Fig. 10. Modeling approach applied to a smart plug, step 6.

4. Is end-to-end modeling possible?

Given the survey of current energy modeling efforts reported in Section 2, it clearly emerges that energy consumption modeling at district and city level is feasible, and can be achieved on the basis of a solid, standard and shared modeling framework based on ontologies. Among the analyzed efforts, DogOnt proved to be a solid baseline model for high granularity information on device states, which can be easily related to both instantaneous (PowerOnt) and temporal (e.g., as in SAREF) behaviors in terms of energy. Due to existing connections between DogOnt and the SAREF ETSI standard [8], any effort for exploiting DogOnt as a seed for end-to-end modeling of the AEC/FM domain can also be seen as a concrete possibility of defining a "unified modeling framework" for the AEC/FM domain based on standard representations (ETSI) and Linked Open Data approaches.

Clearly, some needed glue layers are still missing. In particular when crossing the modeling domains, from bottom layers (devices) to higher layers (district) modeling gaps and inconsistencies emerge and need to be addressed. In the following subsections, initial mappings between layers are, therefore, discussed and their relations with the DogOnt ontology are highlighted. Clearly no fully applicable, generally viable solution can be identified. Nevertheless, end-to-end modeling of the AEM/FC domain seems feasible and most of the gaps appear to be bridgeable through suitable ontology-mappings, many of those basing on DogOnt. This confirms the potential validity of the authors' initial claim.

4.1. Device to Building mappings

Bridging the device-level representation addressed by DogOnt and the relative energy indicators abstracted at the building level is feasible and could be based on, for example, ThinkHome and EEOnt. Unfortunately, direct mappings between DogOnt and these building level ontologies, in the AEM/FM domain, are not always available. While for EEOnt links already exist, which for example relate the `eeont:BuildingEnvironment` concept to the corresponding `dogont:BuildingEnvironment`, or the `eeont:Controllable` and `eeont:Uncontrollable` classes with the homonym classes in Dogont, ThinkHome directly embeds concepts defined in DogOnt, breaking some of the original hierarchies and redefining some of the core classes. This re-use of single classes and/or model subsets, breaks the linking ontology principles and requires explicit mapping to be defined, possibly solving inconsistencies that might arise due to different approaches in modeling.

Still in this case some mappings can be defined which allow exploiting DogOnt as seed model. For example, single sensor and actuator classes in DogOnt can be directly mapped (through `owl:equivalentClass` relations) to corresponding concepts in ThinkHome (see, e.g., Figure 11), while the building modeling branch rooted at `thinkhome:BuildingEnvironment` is completely equivalent to the one rooted at `dogont:BuildingEnvironment`. Moreover, some cross-fertilization might also be considered, e.g., evolving the DogOnt design to better account the different nature of network-specific components and more general devices and/or appliances, as done in ThinkHome.

Far more challenging would be setting up suitable mappings between same-level ontologies, e.g., between ThinkHome and EEOnt as they potentially follow completely different approaches to model the building-level information. Nevertheless, being both linkable to DogOnt, bridging over a common subset of "shared" classes is certainly feasible.

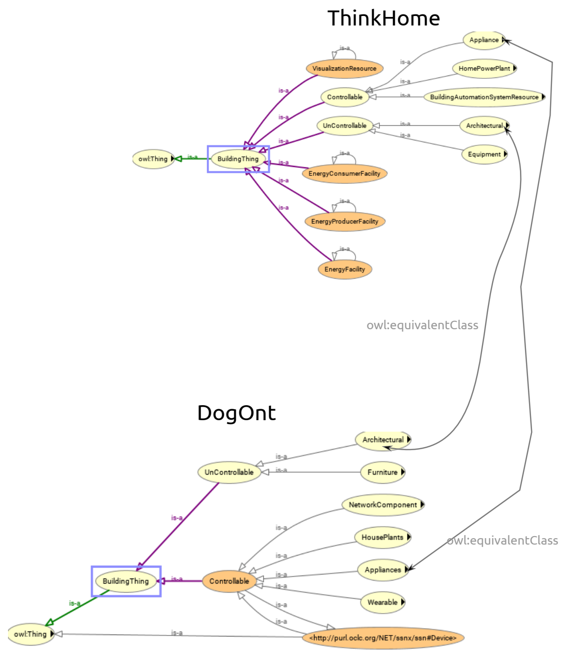

Fig. 11. Oversimplified mappings between DogOnt and ThinkHome.

One of the initially unforeseen commonalities between DogOnt-based modeling and building-level ontologies, emerging from shared adoption of the former, is the rather abstract approach to representation of the built environment, in terms of rooms, walls, etc. While existing efforts in BIM modeling have extensively addressed the building modeling issues, in many applications of the energy domain BIM models are too detailed and include details which are often superfluous. As an example, modeling walls and openings is important for defining the energy indicators of a certain building, however, fine grained details on building materials may often be replaced by much lighter coefficients, thus reducing the computational footprint of the resulting model (e.g., as done in EEOnt). To acknowledge these common needs a dedicated W3C working group[16] is working on the definition of the so-called Building Topology (BOT) ontology, which largely shares the DogOnt modeling approach. In such a sense, defining links between BOT and DogOnt (and this is granted by direct participation of some of the authors to the working group) again enables the adoption of DogOnt a seed model for the AEM/FC domain (see Figure 12).

4.2. Building to city and district mappings

At the district-level, ontologies providing district and city-level views of energy performance indicators and models for energy flows are available. However, a general lack of mappings between systematic representations at district-level and existing building-level characterizations, can be observed. Similarly to the device-to-building case, a possible modeling framework to bridge such a gap can exploit DogOnt as seed, optionally building atop of mappings defined at the building level. Considering the SEMANCO-HEAD ontology [19] as a viable example of district-level model developed in the context of the SEMANCO EU project[17] possible mappings with Dogont can indeed be established. In particular, connections can be defined between the `semanco:Electrical_Appliances` concept and the `dogont:Appliances` class, as well as between the `semanco:Technical_Building_System` and the `dogont:HousePlants`, see Figure 13.

[16]`https://w3c-lbd-cg.github.io/lbdw/`, last visited on April 11, 2017

[17]`http://www.semanco-project.eu`, last visited on April 11, 2017.

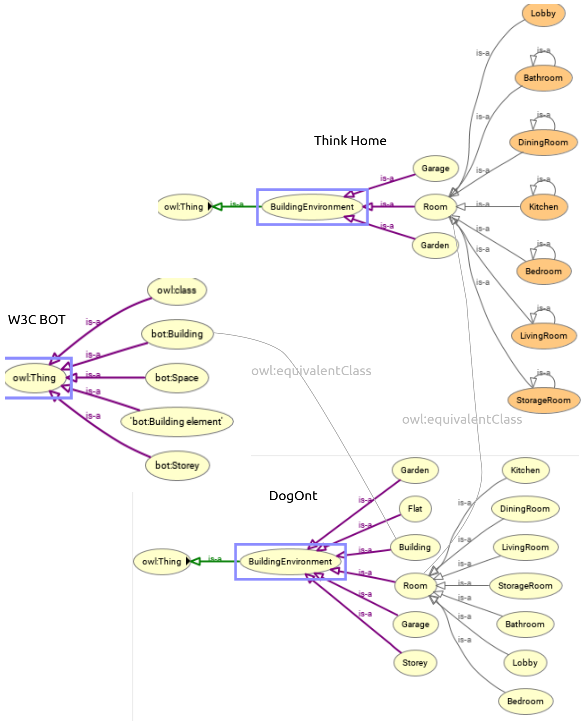

Fig. 12. Relations between DogOnt, W3C BOT and ThinkHome, oversimplified for the sake of clarity.

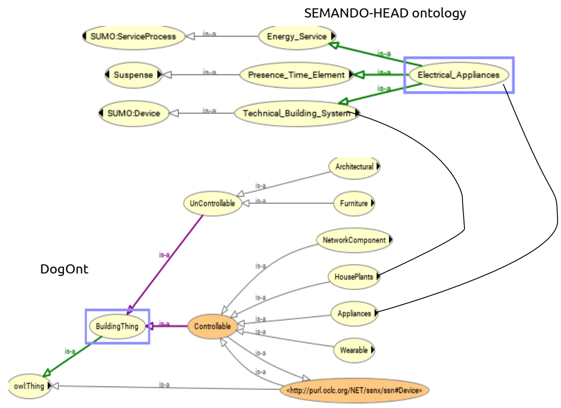

Fig. 13. A very preliminary mapping between SEMANCO-HEAD and DogOnt.

These mappings are not yet published, nor completely checked: for example a possible inconsistency might arise from disjoint axioms in DogOnt that are potentially in conflict with the hierarchy relationship between `semanco:Electrical_Appliances` and `semanco:Technical_Building_System`. Nevertheless it appears clear that, with careful design of ontology interlinks DogOnt can also be exploited to bridge the gap between district and building level modeling. At least on some, almost shared subset of concepts including both the building modeling and the device modeling hierarchies.

5. Conclusions

In this paper we presented the last edition of DogOnt and we discussed its possible role as "emerging" seed for linked, shared modeling of the AEM/FC domain. While monolithic approaches to modeling are clearly not feasible, a linked-open data approach emerging bottom-up from currently adopted models can provide a suitable, shared modeling basis for this challenging domain. According to literature, the DogOnt ontology is starting to emerge as a possible seed to such a bottom-up process and many of the currently available ontologies in the AEM/FC domain can somewhat be referred to such an ontology. While introducing the latest modification to the DogOnt model, the authors highlighted how the emerging role of DogOnt can be sustained by the availability of official mappings between ontologies at the device, building and district levels.

Proposed mappings have various degrees of maturity and are neither exhaustive nor complete. The work presented in the paper, in fact, is more focused on fostering the definition of links between different modeling efforts in the AEC/FM domain rather than in completely specifying ontology mappings and alignments. Several open challenges remain to reach a sufficiently linked set of ontologies for Architecture/Engineering/-Construction (AEC) and Facilities Management (FM).

Additional efforts are needed to explicitly address policy regulations for the energy market, as this aspect is crucial for the successful exploitation of ontology-based energy profiling. In this sense, cities are already moving towards policy-making based on data: an example is represented by the "Global City Indicators" [20], a term created in 2010 to establish a global standard of over 100 city indicators with a standardized definition and methodology. Unfortunately, a formal representation of policies and regulations in the energy domain has yet to appear, and it can be considered as an open challenge that calls for suitable solutions to be designed and developed.

Moreover, applying the full stack of models from district to device can lead to computational issues with inference processes being completed in not acceptable times or even not converging in case of "too" expressive models. Particular attention shall therefore be put on understanding what are the most effective ways of exploiting such emerging, shared models within a feasible computational framework.

References

[1] Michael Compton, Payam Barnaghi, Luis Bermudez, Raúl García-Castro, Oscar Corcho, Simon Cox, John Graybeal, Manfred Hauswirth, Cory Henson, Arthur Herzog, Vincent Huang, Krzysztof Janowicz, W. David Kelsey, Danh Le Phuoc, Laurent Lefort, Myriam Leggieri, Holger Neuhaus, Andriy Nikolov, Kevin Page, Alexandre Passant, Amit Sheth, and Kerry Taylor. The SSN ontology of the W3C semantic sensor network incubator group. *Web Semantics: Science, Services and Agents on the World Wide Web*, 17(0):25 – 32, 2012.

[2] Dario Bonino and Fulvio Corno. *DogOnt: Ontology Modeling for Intelligent Domotic Environments*, pages 790–803. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[3] Dario Bonino, Fulvio Corno, and Luigi De Russis. *PowerOnt: An Ontology-Based Approach for Power Consumption Estimation in Smart Homes*, pages 3–8. Springer International Publishing, Cham, 2015.

[4] ZigBee Alliance Inc. ZigBee Smart Energy Public Profile. Technical report, 2014.

[5] ZigBee Alliance Inc. Zigbee home automation public profile. Technical report, 2010.

[6] European Commission and TNO. Study on Semantic Assets for Smart Appliance Interoperability. Technical report, April 2015.

[7] Laura Daniele, Frank den Hartog, and Jasper Roes. *Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology*, pages 100–112. Springer International Publishing, Cham, 2015.

[8] TS 103 264, SmartM2M, Smart Appliances, Reference Ontology and oneM2M Mapping. Technical report, ETSI, November 2015.

[9] Christian Reinisch, MarioJ Kofler, Félix Iglesias, and Wolfgang Kastner. ThinkHome Energy Efficiency in Future Smart Homes. *EURASIP Journal on Embedded Systems*, 2011(1):104617, 2010.

[10] B. Dong, K. Lam, Y. Huang, and G. Dobbs. A comparative study of the ifc and gbxml informational infrastructures for data exchange in computational design support environments. In *Tenth International IBPSA Conference*, 2007.

[11] J. Verhoosel, D. Rothengatter, F. J. Rumph, and M. Konsman. *An ontology for modeling flexibility in smart grid energy management*, chapter 122, pages 931–938. eWork and eBusiness in Architecture, Engineering and Construction. CRC Press, 2012.

[12] Juan Jose Vinagre, Mark Wilby, Ana Belen Rodriguez Gonzalez Munoz, A.oz, and Jose Garcia. EEOnt: an ontological model for a unified representation of energy efficiency in buildings. *Energy and Buildings*, page In press, 2013.

[13] H. Bell and L. Bjorkhaug. *A buildingSMART ontology*, pages 185–190. eWork and eBusiness in Architecture, Engineering and Construction. Taylor & Francis, 2006.

[14] Christhoph Grimm and Dario Bonino. Towards standardization of M2M communication in Smart Appliances. In Ro-

gelio Segovia and Régis Decorme, editors, *ICT for Sustainable Places, 4th Workshop organised by the EEB Data Models Community*, pages 61– 71. European Commission DG CONNECT H5 Smart Cities & Sustainability, September 2014.

[15] Ian Niles and Adam Pease. Origins of the IEEE standard upper ontology. In *Working notes of the IJCAI-2001 workshop on the IEEE standard upper ontology*, pages 37–42, 2001.

[16] David Harel. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, 8(3):231–274, 1987.

[17] Diego Berrueta and John Phipps. Best Practice Recipes for Publishing RDF Vocabularies. W3c working group note, W3C, August 2008.

[18] Anthony G. Cohn, Brandon Bennett, John Gooday, and NicholasMark Gotts. Qualitative Spatial Representation and Reasoning with the Region Connection Calculus. *GeoInformatica*, 1(3):275–316, 1997.

[19] Vincenzo Corrado, Ilaria Ballarini, Leandro Madrazo, and German Nemirovskij. Data structuring for the ontological modelling of urban energy systems: The experience of the {SEMANCO} project. *Sustainable Cities and Society*, 14:223 – 235, 2015.

[20] Mark S. Fox. A foundation ontology for global city indicators. *University of Toronto, Toronto, Global Cities Institute*, 2013.