Ontology Based Data Model for Power Control in Smart Homes

Ivaylo Atanasov[1], Anastas Nikolov[1], Evelina Pencheva[1]

Abstract – **One of the key enablers of service capabilities in Machine-to-Machine (M2M) communications which can be reused by different applications are data models. Data models explicitly determine the structure of data exchanged between M2M applications. The paper presents an ontology based model of home appliance power control. The model presents the domain knowledge as a set of concepts and the relationships between concepts. The model description is formalized using Ontology Web Language (OWL).**

Keywords – **Machine-to-Machine Communications, Semantic information, Formal representation**

I. Introduction

In recent years, the concept of creating more smart homes gained popularity. Smart Home has a certain degree of "intelligence" which serves people living in the home. The main objective is to increase the comfort of residents, while minimizing energy consumption and reducing costs. This idea already embedded into another technological concept, namely Machine-to-Machine (M2M), acquires clearly defined and realizable projection in many areas of our daily lives [1], [2], [3], [4].

The potential of M2M system to provide applications with capabilities to discover, interpret and use data from smart objects (M2M data) is essential for the creation of high-level M2M services and for the development of an open M2M data market. To enable data transmission in an M2M system, it is necessary to synthesize semantic information. Through such semantic information, applications can discover data without prior knowledge about their existence [5].

In this paper, we present a structural approach to semantic annotation for power control of appliances in a smart home. Survey on research works in the area shows that adding semantic information to a system can be done in various forms [6], [7], [8], [9], [10]. Our approach is to focus on what is sensed and acted upon and it does not limit the M2M system to considerations of a sensor network. Devices are modeled by a generic approximation with minimal information and the system should be able to integrate them using this information. This semantic information describes the devices as things which may be considered as device shadows in a M2M system.

The paper is structured as follows. First we present the ETSI approach to structuring semantic data of domain-specific knowledge. Next we describe the ontology for the power control in a smart home. Then we model the data following the ETSI approach. Parts of formal model description with the OWL language are presented. The conclusion summarizes our contribution.

[1]The authors are with the Faculty of Telecommunications at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, E-mails: iia@tu-sofia.bg; nikolov.anastas@gmail.com; enp@tu-sofia.bg

II. The ETSI Approach to Modeling M2M Semantic Data

In (ETSI TR 101 584), it is outlined the approach to modelling M2M semantic data. The main guiding principles are separation of the abstract information model from its representation in the ETSI M2M resource structure and separation of domain-specific knowledge from instance-specific data. The synthesis of data abstraction needs to be independent of ETSI M2M resource architecture defined in ETSI TS 102 60 and which may be updated in future releases. With separation of instance and domain knowledge, then the instance-specific data can be understood without any further a-priori knowledge, and different applications can interact on the bases of common domain model.

Fig.1 shows the structure of ETSI domain specific knowledge.

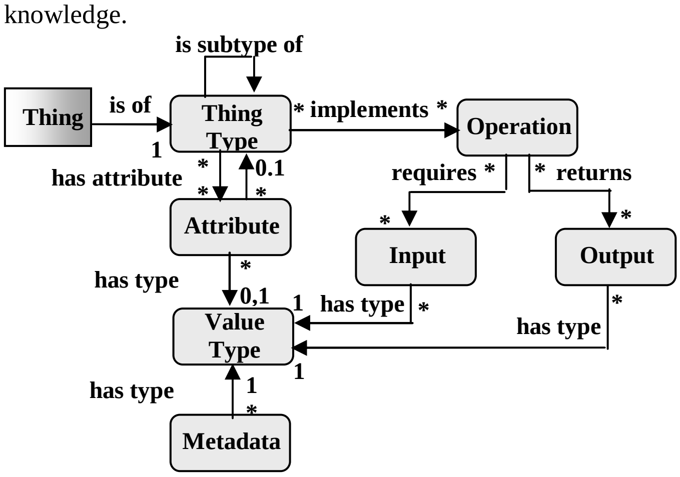

Fig.1 Structure of domain specific knowledge (ETSI TR 584)

Ontology is used to express the information represented by an application domain model. The model identifies Thing types, which have attributes, data about data (metadata) and implements operations which require inputs and return outputs of value types. Attributes, metadata, inputs and outputs have certain value types.

III. Ontology for Power Control in a Smart Home

Let us consider a home energy saving system (HESS) whose general purpose is energy saving through power control on electrical appliances at home. For example, HESS may control the heating, cooling and ventilation home systems and the hot water tanks. For heating and cooling the energy saving is achieved by small changes in the referenced (preferred by the home owner) temperature during predefined time periods when nobody is at home or everybody sleeps [11]. The ontology web that represents the domain knowledge is shown in Fig.2.

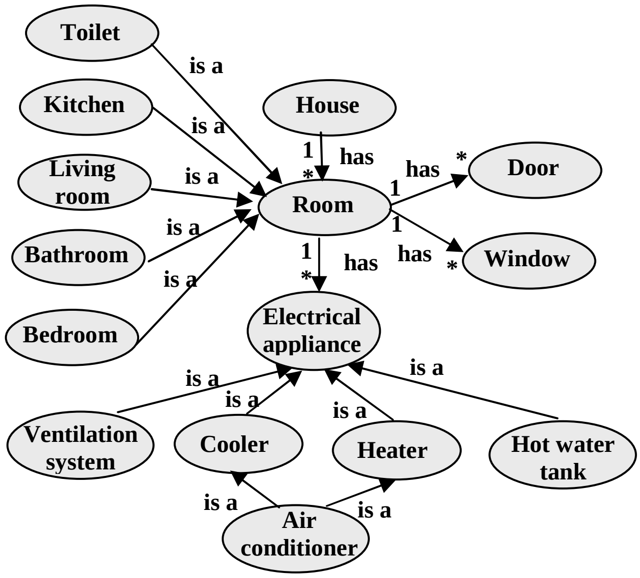

Fig.2 Ontology web for power control in a smart home

The basic concepts in the domain of power control in a smart home are as follows. A house has rooms, a room has windows and doors, a room has electrical appliances that may be remotely controlled. The room may be a kitchen, a living room, a bathroom, a toilet and a bedroom. The home owner may define power shedding schedule for heating/cooling control and a policy stating that if a room door or a room window is open, the heater/cooler has to be switched off.

IV. Domain-Specific Model for Power Control in a Smart Home

The domain specific model defines Thing types, Thing type attributes and operations. Fig.3 shows the domain-specific model of power control of electrical appliances in a room.

The *ElectricalAppliance* type has the following attributes. The *ElectricalOperation* attribute may have on-off or on-off-standby values. The *UsageMode* attribute may have autonomous or manual (user directed) values. The *Usage* attribute may have periodic, permanent or semi-random values. Only electrical appliances that have autonomous usage mode and on-off-standby electrical operation may be controlled remotely. The *ElectricalAppliance* type implements the following operation. The *powerOn* operation is used to power on the appliance, while the *powerOff* operation is used to power off the appliance. A controllable appliance such as a cooler, heater, air conditioner, ventilation system and hot water tank, when is powered on, goes in a standby state.

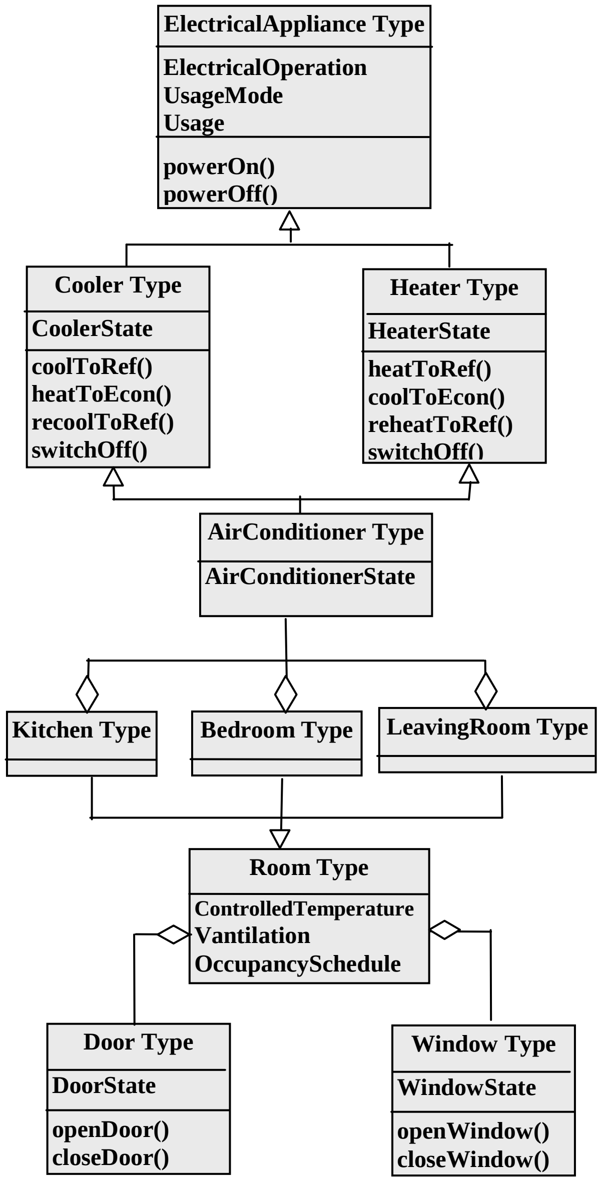

Fig.3 Domain information model of controllable appliances

The *CoolerState* attribute of the *Cooler* type indicates the cooler state (e.g. powered off, standby, etc.).

The *Cooler* type implements the following operations. The *coolToRef* operation instructs the cooler to cool to the referenced temperature T_{ref} (the temperature preferred by the home owner) and it requires T_{ref} as an input. The *heatToEcon* operation instructs the cooler to switch off while the room temperature increases by a small amount (Δ) to realize energy economy and it requires Δ as an input and $T_{ref-\Delta}$ as an output. The *recoolToRef* operation instructs the cooler to re-cool to the referenced temperature and it requires T_{ref} as an input. Similarly, the *Heater* type implements the following operations. The *heatToRef* operation instructs the heater to heat to the referenced temperature T_{ref} and it requires T_{ref} as an input. The *coolToEcon* operation instructs the heater to switch off while the room temperature decreases by a small amount (Δ) to realize energy economy and it requires Δ as an input and $T_{ref-\Delta}$ as an output. The *reheatToRef* operation instructs the heater to reheat to the referenced temperature and it requires T_{ref} as an input. The *HeaterState* attribute of the *Heater* type indicated the heater state. Both *Cooler* type and *Heater* type implement the *swithOff* operation which instructs the appliance to switch off. More information about heating/cooling control may be found in [11].

The *Room* type has the following attributes. The *ControledTemperature* attribute indicates whether the room is heated or not, e.g. the toilet is an unheated room, while the bedroom is a heated one. The *Ventilation* attribute indicates whether the room has a fan that pumps the air or not. The *OccupancySchedule* attribute indicates occupancy schedule of the room. The *Door* type implements *openDoor* operation and *closeDoor* operation and it has a *DoorState* attribute. The *Window* type implements *openWindow* operation and *closeWindow* operation and it has a *WindowState* attribute.

V. FORMAL DESCRIPTION OF THE DOMAIN-SPECIFIC MODEL

This section describes parts of the formal description of the proposed semantic information model of air conditioner device using OWL 2.

The description of the domain-specific model for power control of electrical appliances in a smart home is formalized by the OWL (Ontology Web Language) which is used to represent ontologies in the web. OWL is a language based on computational logic, so that the knowledge expressed in OWL can be analyzed by a computer program in order to verify compliance of this knowledge and turn it from implicit to explicit [12]. OWL ontologies can be published on the World Wide Web and can refer to other ontologies or directed by them.

The software product Protégé is used for the creation and verification of the ontology of the domain-specific model for power control in a smart home. The product fully complies with the standardized specification of the WWW Consortium to create OWL 2 and RDF descriptions [13].

Every "thing" in the OWL world is a member of the class owl:Thing. So, every class is implicitly a subclass of owl:Thing. Class definition has two parts: a name and a list of restrictions. Instances of the class belong to the intersection of these restrictions. Element subClassOf is a down of possible restrictions. Instances are members of classes.

The properties allow describing the general relations for class members and specific facts about instances. There are two types of properties:

- properties of the types of data that describe the relations between instances of classes and RDF literals and XML Schema data types;
- properties of objects that describe the relations between instances of two classes.

When defining the properties, e.g. relation, there are different ways to impose restrictions. As the relations in mathematics, properties are defined with domain (domain) and the range of values (range).

Statements in Figure 4 declare a fragment of named individuals, classes, and object properties.

Figure 5 describes the part of of the relationships between declared individuals, classes and attributes.

Figure 6 shows the data type declarations and definitions.

VI. CONCLUSION

Data models representing semantic information of the domain specific knowledge allow information sharing between different applications. We propose a structural approach to modeling semantic information for power control of electrical appliances in a smart home. A target physical system with distributed monitoring sensors and controlling actuators is represented in the digital world as a set of things which might be controlled or monitored by M2M applications. Things are identified by approximation to a generic model and the M2M system integrates them on the basis of semantic information. Using domain-specific ontology, we define things types, which have attributes and implement operations with input and output parameters. The data model is formally described by OWL and verified by the use of the Protégé tool.

REFERENCES

[1] C. Hung, Y. Bai, P. Chen, J.-M. Hsu, “Remote control of home lighting devices with RFID identification and current detection of outlets”, IEEE 17th International Symposium on Consumer Electronics ISCE '13, Taiwan, Conference Proceedings, pp.7-8, 2013.

[2] K. Hwang,S. Yoon, “A Multifunctional RF Remote Control for Ultralow Standby Power Home Appliances”, *International Journal of Distributed Sensor Networks*, Available at: http://www.hindawi.com/journals/ijdsn/2014/381430/, 2014.

[3] V. Kumar, A. Fensel, G. Lazendic, U. Lehner. „Semantic Policy-Based Data Management for Energy Efficient Smart Buildings”. OTM’2012 Workshops, *Lecture Notes in Computer Science*, vol.7567, pp.272-281, 2012.

[4] G. Wu, X. Yu, “Remote control system for energy efficient home”, IEEE Conference Energytech’2013, Ohio, USA, Conference Proceedings, pp.1-5, 2013.

[5] A. Fensel, S. Tomic, V. Kumar, M. Stefanovic, S. Aleshin, D. Novikov, “SESAME-S: Semantic Smart Home System for Energy Efficiency”, *Informatik Spektrum*, vol.36 issue 1, pp.46-57, 2013.

[6] Z. Qunzhi, S. Natarajan, Y. Simmhan, V.Prasanna, ”Semantic Information Modeling for Emerging Applications in Smart Grid”, International Conference on Information Technology: New Generations ITNG’2012, Las Vegas, USA, Conference proceedings pp. 775-782, 2012.

[7] S.Rohjan, M.Uslar, M. Specht, A. Niesse, M. Troschel, “Towards Semantic Service Integration for Automation in Smart grids” *International Journal of Distributed Energy Sources*, vol.8 number 2, pp.119-146, 2012.

[8] S. Rohjans, K. Piech, W. Mahnke, “Standardized Smart Grid Semantics using OPC UA for Communications”, *IBIS – Interoperability in Business Information Systems*, vol.6, issue 10, pp.21-32, 2012/

[9] Y. Simmhan, Q. Zhou, V. Prasanna, “Semantic Information Integration for Smart Grid Applications,” Chapter 19 in edited book *Green IT: Technologies and Applications*, pp.361-380, 2012.

[10] Y. Tang, I. Ciuciu, “Semantic Decision Support Models for Energy Efficiency in Smart-Metered Homes”, IEEE

International Conference on Trust, Security and Privacy in Computing and Communications TrustCom'2012, Liverpool, UK, Conference Proceedings. pp. 1777-1784, 2012.

[11] I., Atanasov, M. Ivanov, A. Nikolov, R. Dimova, E. Pencheva. "Towards web services for power control", International Journal on Information Technologies & Security, IJITS, 2014, no.4, pp.3-16.

[12] W3C OWL 2 Web Ontology Language Primer, Second edition, W3C Recommendation, 2012.

[13] Protégé, Stanford University, Available at: http://protege.stanford.edu/, Accessed 2015.

```
Declaration ( NamedIndividual ( :TheAirConditioner ) )
Declaration ( Class ( :AirConditioner ) )
Declaration ( Class ( :Appliance ) )
AnnotationAssertion ( rdfs: comment :Appliance "Represents the set of all appliances." )
Declaration ( ObjectProperty ( :hasUsage ) )
Declaration ( ObjectProperty ( :hasElectricalOperation ) )
Declaration ( ObjectProperty ( :hasUsageMode ) )
Declaration ( ObjectProperty ( :hasAirConditionerID ) )
Declaration ( ObjectProperty ( :implementsCooling ) )
Declaration ( ObjectProperty ( :implementsReheating ) )
Declaration ( ObjectProperty ( :implementsHeating ) )
Declaration ( ObjectProperty ( :implementsReCooling ) )
Declaration ( ObjectProperty ( :requiresTref ) )
Declaration ( ObjectProperty ( :requiresDelta ) )
Declaration ( ObjectProperty ( :requiresTrefminus ) )
Declaration ( ObjectProperty ( :requiresTrefplus ) )
Declaration ( ObjectProperty ( :requiresTrefplus ) )
Declaration ( ObjectProperty ( :requiresTrefminus) )
Declaration ( ObjectProperty ( :returnsTrefminus) )
Declaration ( ObjectProperty ( :returnsTrefplus ) )
```

Fig.4. Declarations related to OWL description of air conditioner information model

```
SubClassOf ( :AirConditioner :Appliance)
ClassAssertion ( :AirConditioner :TheAirConditioner)
SubObjectPropertyOf ( :requiresTref  :implementsCooling)
SubObjectPropertyOf ( :requiresDelta  :implementsCooling)
SubObjectPropertyOf ( :returnsTref-  : implementsCooling)
SubObjectPropertyOf ( :requiresTref-  :implementsReheating)
SubObjectPropertyOf ( :requiresDelta  :implementsReheating)
SubObjectPropertyOf ( :returnsTref  : implementsReheating)
```

Fig.5 OWL relationships description in the an air conditioner information model

```
Declaration ( Datatype ( :UsageMetadata )
Declaration ( Datatype ( :ElectricalOperationMetadata )
Declaration ( Datatype ( :UsageModeMetadata )
DatatypeDefinition (
        :UsageMetadata
        DataOneOf ( "1"^^xsd:integer "2"^^xsd:integer "3"^^xsd:integer"
        Annotation (rdfs:comment "1 means permanent, 2 means semi-random, 3 means periodic" )
        )
)
DataPropertyRande  (:requiresDelta xsd:nonNegativeFloat)
DataPropertyRande ( :hasAirConditionerID xsd:nonNegativeInteger)
```

Fig.6. OWL data type declarations and definitions in the air conditioner information model