Semantic EnOcean: A Tool for Mapping Syntactic Device Descriptions onto an Ontology for the Internet of Things

Janna Herrmann, Jochen Britz and Jan Alexandersson
DFKI GmbH
Saarbrücken, Germany
Email: <f>.<l>@dfki.de

***Abstract*—This paper contributes to the semantification of the Internet of Things by outlining the efforts in including the Digital Concepts Smart EnOcean Gateway into the ISO/IEC 24752 Universal Remote Console framework (URC). Without our work, integrating the syntactic EnOcean device profiles into semantic environments is bound to an manual mapping of each profile to an appropriate ontology. In this paper, we describe a method and tooling for a semi-automated multistage process to map these profiles to an existing ontology. Strictly speaking, EnOcean device descriptions as given by a commercial EnOcean gateway, namely Digital Concepts Smart EnOcean Gateway, are mapped onto a smarthome ontology, namely DogOnt. As a side effect, the ontology is enhanced to fit the requirements of the device descriptions and extended by the actual devices. Finally, with the resulting ontology as a starting point, URC standard-conform, semantically enriched abstract user interface descriptions are generated. Through semantic enrichment of these User Interface Sockets, the functioning and purpose of the corresponding devices becomes machine-understandable and therefore can be easily used in semantic environments like Smart Homes. Without this additional semantic information, an automated generation of comprehensible user interfaces based on User Interface Sockets would not be feasible.**

Keywords–EnOcean; ISO/IEC 14543-3-10; OpenURC; ISO/IEC 24752; IoT; Domotics; Ontology

I. INTRODUCTION

Internet of Things (IoT) emerges rapidly and increasingly covers many technologies that were previously found within the scope of Smart Homes, Ambient Intelligence, etc. The term "Internet of Things" describes the continuous replacement of the classical computer as a stand-alone device by so-called smart objects. Small and embedded computers should no longer be in the focus of attention, but rather support the user with his/her activities [1]. Estimations vary, but there is a broad agreement that there will be many IoT devices: around 20 billion in 2020 [2]. With the growing market, the need for devices that can be easily integrated into the IoT increases.

Devices based on ISO/IEC 14543-3-10 EnOcean Technology [3] are wireless and follow the principle of energy harvesting, which means that the devices are mostly battery-free. The energy required for communication is taken from the environment, e.g., from changes in temperature, solar energy or even kinetic energy of a button press. The EnOcean-Standard is manufacturer-independent and most devices are for domestic environments. The devices can be connected to the Internet through a gateway, for example the recently developed Smart EnOcean Gateway by Digital Concepts [4]. It allows many existing EnOcean devices to connect to the gateway, and the gateway provides a RESTful interface in return.

Every EnOcean device has a profile describing its functionalities in an abstract way. The gateway exposes the profiles based on a syntactic description (JSON REST API) without semantic informations. Hence, the corresponding device and its functionalities are not (easily) machine-understandable, which turns the integration into the IoT to an activity associated with manual work. Purely syntactic descriptions prevent the automated generation of, for instance user interfaces, because necessary characterization of the device, e.g., its type (lamp, switch, etc.) and other information is missing. An enrichment of the profiles with semantic information would solve this problem.

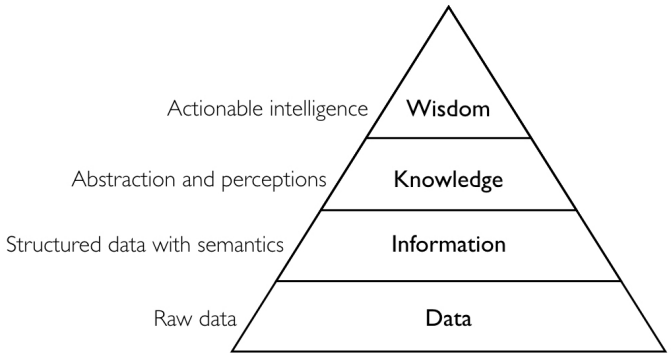

Figure 1. Illustration of the "Knowledge Hierarchy" [5] adapted to the context of the Internet of Things.

The importance of semantics in the IoT in general was pointed out by Barnaghi et al. in their work about semantics for the Internet of Things [6]. They adjusted the well known "Knowledge Hierarchy" [5] to fit into the context of IoT, as depicted in Figure 1. The lowest layer represents the raw data generated by the IoT devices. The next layer is the "Information"-layer. Information can be understood as structured and machine-understandable data with semantic meaning. Based thereon, there is the "Knowledge"-layer, as abstraction and perceptions of the underlying semantic information. On top is the so called "Wisdom"-layer which can be interpreted as actionable intelligence derived from the underlaying abstractions. Clearly, semantic modeling is vital to the development of real knowledge or even wisdom-filled environments.

In this paper, we start in the next section with a brief introduction of the related work, inlcuding the Smart EnOcean Gateway, DogOnt and the URC standard. The sections IIIand IVcontains a description on how to map the EnOcean profiles onto the DogOnt ontology. The generation of the resources for the Universal Control Hub (UCH) – part of the ISO/IEC 24752 Universal Remote Console standard [7] – including User

 ISBN: 978-1-61208-601-9

Interface Sockets, resource sheets, and its semantics will be portrayed in Section V. The work is discussed in Section VIand concluded in Section VII.

II. Related Work

A. Mapping of relational data to an ontology

Ontology mapping is not new and in research there is more than one approach to do so. One is the mapping of data from an relational database to an ontology by Sven Peter [8]. His tool is based on the method by Nussbaumer, Haslhofer and Klas [9] specialized for the used ontology. In order to ensure a high degree of precision an semi-automated approach is used. In the approach, the final decision as well as the constraints for the automated part are in the responsibility of a domain expert.

The work shows that a semi-automated approach can be very effective. In our case, the user is the expert for its own smart home and therefor is in control of the final decision.

B. The Smart EnOcean Gateway

Each EnOcean product comes with a profile that is registered and made public by the EnOcean alliance. A profile is described in the interoperable wireless standard for home and building automation, namely ISO/IEC 14543-3-10 [3]. Recently, there is a commercial gateway available [4] which provides a set – about 150 – of standardized EnOcean profiles. The gateway provides the profiles in a unique but homogeneous way through a JSON API; see Listing 2 for a sample profile of a temperature sensor.

An EnOcean profile starts with a unique identifier, the "EnOcean Equipment Profile" (EEP) followed by a natural language description of the device and a list of function groups. All functions in a group have the same signal direction, either "from" or "to". The sensor in Listing 2 has a function, which has the direction "from". "Function"s have "key"s, which are unique in its groups and may have a natural language description analogous to the profile itself. A function has a list of values. A "value" can be a range with "min" and "max" values, "step" and "unit" but also an enumeration of string values. Consequently, the value of a function can be both a numeric value and also a string-based value. For instance, the numeric value is used if the sensor works under normal conditions and the string value in case of exceptions, as depicted in Listing 2 . This fact is a challenge when working with strongly typed programming languages, where variables typically have exactly one particular type, e.g., JAVA.

C. The DogOnt Ontology

Per definition, an ontology is a "formal specification of a conceptualization" [10]. A special case of an ontology is a Domotic Ontology. "Domotic" stands for DOMus infOrmaTICS, which can be translated to "information technology for homes". Domotic is often used in conjunction with Smart Homes. A prominent example of a domotic ontology is DogOnt [11]. DogOnt is specialized for domotic environments and was developed by Dario Boninound and Fulvio Corno in 2008 [11]. The aim of this ontology is to transfer domotic environments into intelligent domotic environments by enriching them with semantic information on a manufacturer independent level. Other semantic representations with domotic background is, for example, the ASAP-Ontology [12] but this

```
"key" : "temperature",
"description" : "Temperature (linear)",
  "values" : [ {
     "range" : {
        "min" : 0,
        "max" : 40,
        "step" : 0.157,
        "unit" : "°C"
        }
      }, {
     "value" : "overRange",
     "meaning" : "Temperature sensor
                  failure or out of range"
      } ]
```

Figure 2. The function "temperature" from the EnOcean Profile with EEP A5-20-01 contains a numeric and a string-based value.

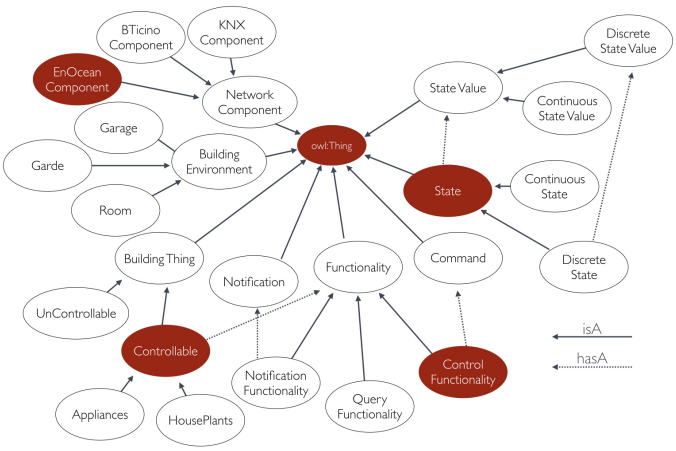

Figure 3. Rough overview over the domotic ontology DogOnt. Important parts are highlighted in red: *Thing*, *EnOcean Component*, *Controllable*, *State* and *Control Functionality*.

ontology is too powerful and models many aspects that are not crucial for our use case. Furthermore, we have selected DogOnt because:

1) It is manufacturer agnostic, so are the EnOcean principles.
2) DogOnt is specialized for domotic environments, which most of the EnOcean sensors and actuators belong to.
3) DogOnt is formalized in Web Ontology Language (OWL) [13], a formal description language for ontologies and therefore easily accessible and usable.

A rough overview of a part of the ontology is shown in Figure 3 . Starting from the standard class *owl:thing* of the OWL description language some of the DogOnt classes and relations are depicted. Especially important for our use case are, among others, the classes *EnOceanComponent*, *Controllable*, *ControlFunctionality* and *State*. More specific information about the roles of these classes for our work will be discussed in Section III.

D. The Universal Remote Console framework

The Universal Remote Console (URC) framework, specified in ISO/IEC 24752 [7], provides an adaptive architecture

that supports the flexible integration and reuse of heterogeneous software and hardware components, communication protocols, and target appliances. URC allows for interfacing arbitrary appliances and services, so called targets and expose them in arbitrary ways through a mechanism referred to as "Pluggable User Interfaces".

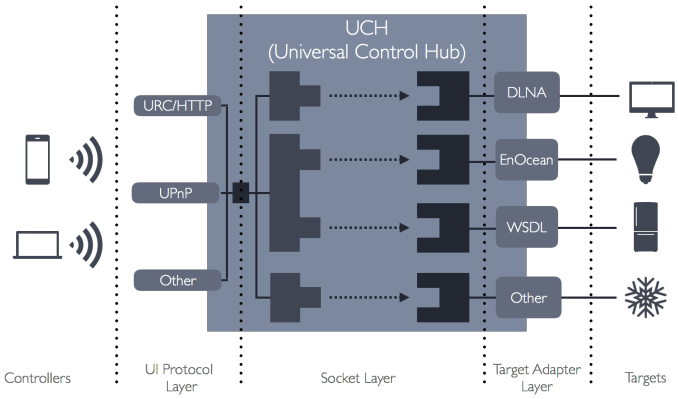

Figure 4. Overview UCH architecture

The advantage of this approach is the possibility to use consistent, secure, personalized and, perhaps most importantly, accessible user interfaces [14]. Users can select a user interface that fits their needs and preferences, using input and output modalities and interaction mechanisms they are familiar with and work well with [15].

The standardized definition of the User Interface Socket (UIS) describes the input and output modalities of a specific target on an abstract level and therefore is the common model of all user interfaces and communication protocols. An UIS contains a flat set of socket elements which are either variables, commands, or user notifications, providing a synchronized communication channel to the controlled device and its current state. The description also specifies how socket elements depend on each other: for example, the commands for confirming or canceling a notification are only relevant while the notification request is active. More advanced dependencies can also be described through the notion of pre- and post-conditions.

The UIS does not provide enough information for generating nice-looking, comprehensible and accessible user interfaces. Concrete instructions on how to build user interfaces are missing: the arrangement and structure of the elements, the language of the labels, additional icons, etc. These can be added as extra resources, the so called "Pluggable User Interfaces". The socket can then be rendered on some controller, giving the abstract UIS a concrete implementation. The concrete user interface connects to one or multiple sockets in two directions: first, getting and representing the values that reflect the current state of the target, and second, requesting changes in the target's state through variable changes and command invocations.

The Universal Control Hub – UCH for short – implements the URC standard in a gateway approach. The architecture consists of three important layers: the UIProtocol Layer, the Socket Layer and the Target Adapter Layer, which connetcts arbitrary Controllers with Targets, as shown in Figure 4. In the Target Adapter Layer, each target is represented by a dedicated target adapter that is responsible for the grounding of abstract socket elements with a specific network protocol or other proprietary communication mechanisms. Target adapters usually consist of a combination of software and hardware, which communicates with the corresponding target and translates its signals to URC. Similarly to the EnOcean alliance, the OpenURC alliance [16] develops and publishes sockets that can be usedby any vendor for making their product/service URC ready.

III. From Syntax to Semantics

Clearly, the sheer amount of EnOcean devices and profiles is a moving target: some device profiles disappear, some emerge. The "EnOcean device profile", "EnOncean profile" or simply "profile" descriptions are already homogeneous and unified but their structure and content is very close to hardware and without semantic informations. In some cases, even humans are not fully capable to deduce the underlying device by means of a profile. Thus, one big challenge addressed in our work is the enrichment of non-semantic structured data with semantic information in automated fashion as far as possible. Since the semantics are represented in OWL, the challenge refines to a semi-automated mapping of non-semantic structured data to a given ontology. Our solution, a multistage mapping approach is described in Section IV . We use the "HermiT" reasoner [17] for consitency check and ontology queries. As laid out in Section VI, HermiT can unfortunately not be used for identifying subsumption relationships. We query the ontology using the "Manchester OWL Syntax" [18]. Another big challenge to conquer is not to loose any information throughout the whole process. Looking at the children's game "Chinese whispers", where some information gets whispered from one child to another, the message steadily alters until it maybe changed completely. This paradigm is also applicable to this work, since at first the EnOcean profiles are mapped to the ontology and at last, semantically enriched User Interface Sockets are generated. Therefore, if some non-important information is evidently lost somewhere in the process, it could lead to larger deviations to the originally profile in the end. Consequently, we needed to find ways to keep all information, no matter how small, in standard- and ontology-conform ways.

Finally, the URC Standard does not specify semantic information for the User Interface Socket, but the URC standards allows for extensions. In connection with the problem above, we had to figure out, how the semantic information can be added in a standard conform manner, which is discussed in Section V.

IV. Mapping EnOcean onto DogOnt

Semantic information is modelled with the DogOnt ontology, see Section II. We have developed a semi-automated multistage approach to map the profiles to DogOnt by combining terminological, structural as well as semantic mapping techniques on both element- as well as structure-level. The logical process of these mapping steps is shown in Figure 5 . Additionally, we involve human decicions in the loop, in the processing step we refer to as "manual mapping". This is necessary in order to select "good" ontological candidates, as discussed below.

 ISBN: 978-1-61208-601-9

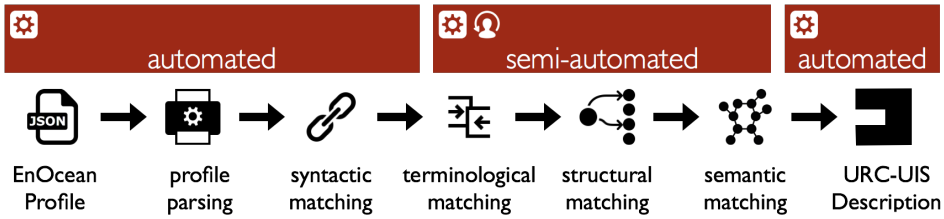

Figure 5. The different steps and matching techniques used to perform the mapping between profiles and user interface sockets.

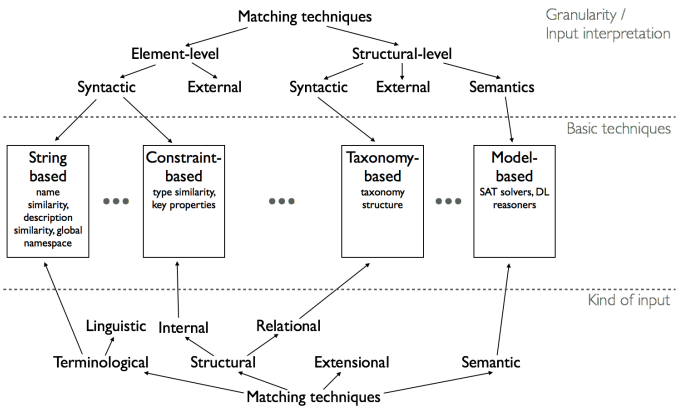

Figure 6. A Classification of ontology matching techniques [19].

Different mapping techniques are used to find best matching partners between concepts in the DogOnt ontology and devices, functions and type restrictions described by the EnOcean device profiles. The classification of these techniques is taken from [19], as shown in Figure 6 . Since DogOnt only comes with a top-level EnOcean concept, each profile gives raise to one new DogOnt entity. Subsequently, for each profile's function and value, new DogOnt entities are created, which are then added to the profile entity.

The first step, see Figure 5 is a fully automated constraint-based mapping technique operating on the profile's complete content, e.g., functions, values, etc. The constraints have been derived on a basis of comprehensive analysis of the profiles, their structures and the available DogOnt concepts. One such constraint states that a new *Controllable* is generated for each profile. Next, the profile's content is translated to ontology concepts. According to their signal direction, EnOcean functions give raise to new sublasses of either *State* or *ControlFunctionality* which are then linked to the *Controllable* with existing DogOnt ObjectProperties using existential restrictions, e.g. *hasFunctionality some...*. Similarily, the profiles' values, namely parameters and return values, are translated to one of four concepts. Values are either a numerical range (possibly with unit and so-called step size) or a fixed string value. A value representing a numerical range corresponds to *ContinuousValue* or a *ParametricCommand* depending on its function's signal direction. Values' properties, e.g., "min", "value", ..., see Figure 2 are translated into DogOnt's "data properties" with appropriate data restrictions and assigned to a new *ContinuousValue* or a new *ParametricCommand*. Fixed string-values translates to *DiscreteValue*s with fixed *realStateValue*s or to *NonParametricCommand*'s value restrictions with fixed value restrictions of the data property *realCommandName*. Recursively, the function specific values are linked to the DogOnt entity representing their corresponding function with an appropriate ObjectProperty, e.g. *hasStateValue* or *hasCommand*. Finally, the *Controllable*'s EEP is assigned to specifically created entities of the *Controllable*.

In the second step, a terminological mapping using the results of the previous step is applied. This mapping applies to the fixed string-values of EnOcean functions, e.g., the value "overRange" in Figure 2 . An appropriate matching partner in *DiscreteValue* or *NonParametricCommand* must have a value restriction of the *realStateValue* or the *realCommandName* respectively, with the exact string value describing the functions fixed value (here: "overRange"). This rule does not apply for *ContinuousValue* or *ParametricCommand*. Here, either the profile's function key is already known or it has to be specified manually. As a fall-back solution, the most general concept is used. If a more specific matching partner is found, the ontology entities created in step one and their object properties are adapted accordingly.

In the third step, possible existing matching partners for the new *State*s and *ControlFunctionalitiess* are filtered using relational mapping. Of course, the possible matching partners have to fulfill the type and cardinality restrictions of each *StateValue* and *Command* e.g., a *ContinuousState* has to have one or more *ContinuousValues*. If the *StateValue*ss and *Command*s are sufficiently specified in the previous step, the result quickly becomes distinct. In case of an ambiguous outcome, again either the most general matching concept from step one is chosen or the most appropriate candidate is selected by user interaction.

In the fourth and final step, a semantic matching between the already build-up ontological device representation and existing devices in the ontology is performed. Here, possible superclasses with less or equal capabilities are searched for. If the outcome is ambiguous, again either the most general matching concept is chosen or user interaction is required. Since only devices with less or equal capabilities are considered and ontologies allow for multiple inheritance, the user can select multiple devices and corresponding Sub-class axioms for these classes. Eventually, a new ontology entity will be created.

A screenshot from the tool is shown in Figure 7 .

V. Generating of Semantically Enriched URC-Sockets

We expose the semantically enriched device profiles through the gateway architecture within the ISO/IEC 24752

 ISBN: 978-1-61208-601-9

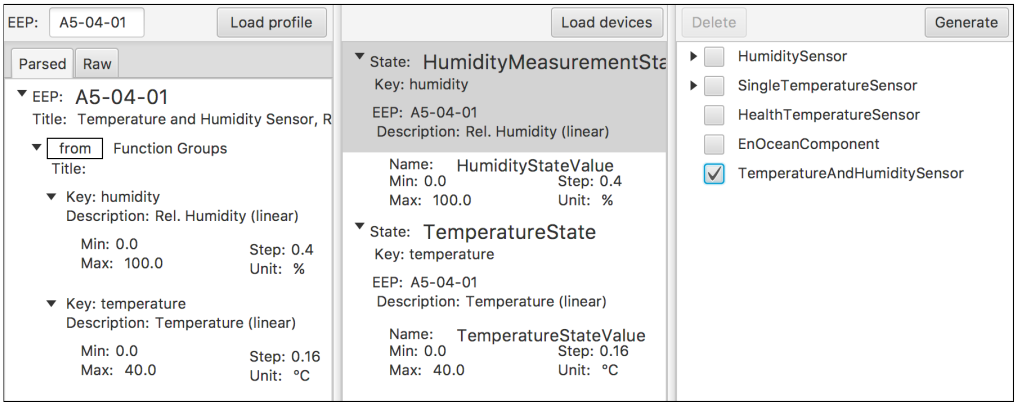

Figure 7. A screenshot of the developed tool. The first column shows the parsed profile, the second shows the current mapping of the functions and values and the third shows the posssible matching partners for the profile in the ontology.

Universal Remote Console standard: The Universal Control Hub (UCH). For every EnOcean profile, a target and User Interface Socket (UIS) description is required. The target description represents the general information of a device, while the UIS description mirrors the exact functionalities. The structure of both descriptions is specified in the URC-Standard [7]. The automated UIS generation is based on entities of the ontology. An entity in the ontology representing a specific profile contains an EEP and other profile specific values and functionalities and will be called profile entity in the following. The semantic enrichment is realised by referencing elements of the UIS and target description with their corresponding entities in the ontology. Thus, the actual version of the ontology is required in order to make use of the semantic information.

As a unique ID for the UIS description the EEP of the corresponding profile is used, since both profile and UIS are an abstract description of the functionalities of a device this is an adequate choice. Furthermore, the EEP is the only unique field of a profile. In the “about”-tag of the UIS description the name of the corresponding profile entity is used.

The generation of the UIS is based on the following rules. “State”s of the profile entity are transformed into variables of the UIS. The type of this variable is a custom type reflecting the corresponding “StateValue”s in the ontology. This custom type is generated as a restricted “xsd:simpleType” and added to the XML schema included in the socket description. These restrictions are either a list of valid strings, the definition of a range or both, in case a function has a numerical and a string based value. To keep all information, the restrictions are also annotated with unit and step size.

Besides the variables, there are commands in the ontology and also in the UIS description. Nevertheless, a direct translation from ontology “Command”s in UIS-commands is not possible, because of the of structure of the EnOcean device profiles, the corresponding behavior of the Smart EnOcean Gateway and the demand to keep all information. The profiles can bundle functions into “functiongroups”. In order to send a command to a device it is necessary to send all commands of one “functiongroup” together. Therefore the generated profile specific “Command”s of the ontology contain information about the “functiongroup” they belong to. In the UIS, one “ControlFunctionality” transforms into one UIS-command and the corresponding DogOnt “Command”s transform into parameters of the UIS-commands. The types of these parameters are handled analogous to the variable types. As type for the UIS-commands “uis:voidCommand” is chosen, because it represents simple commands without timing constrains and return values.

Alongside the UIS description, a “target description” and “resource sheet” are generated. Based on the profile entity, the generated target description uses the EEP followed by the word “Target” as ID, in which the corresponding socket description and the resource sheet are referenced. In order to make the references in the descriptions meaningful, the resource sheet links the corresponding version of the ontology.

VI. Discussion

The main research topic behind our work was to be able to map all profiles to the ontology. The mapping needs to be error free because of the practical context of the tool. Wrong mapped profiles would lead to unusable User Interface Sockets. Our solution was to include the user in this mapping process, which leads to new research problems on how to model the interaction and how to avoid errors made by the user itself. Therefore finding a way of balancing theoretical research and the practical application was one of the main issues. Besides this interaction there are other discussion points regarding the created tool.

As for now, the ontology extension is kept locally. In the future, they should be made available to third parties via some cloud-based service, such as the OpenURC Resource Server. By doing so, new users do not need to start from scratch but could share already gathered background knowledge. However, semantically “sub-optimal” decision could find their way to the cloud and affect future mappings, which has to be prevented by some validation process, such as the OpenURC Technical Committee Procedures [20].

Another point of discussion is the selection of the DogOnt ontology. DogOnt has the advantage of being specialized for domotic environments, the field of use of this work. Furthermore, it is possible to model the required objects and functionalities with some room for improvement and extensions of the work. However, DogOnt limits our work since some necessary classes are not modelled in a generic manufacturer-independent way. As an example, the class *TemperatureSensor* has a Zigbee-specific *GroupFunctionality*. Moreover, some classes

 ISBN: 978-1-61208-601-9

are not specific enough. For example, *Sensor* is a subconcept of *Controllable* but with no additional properties. Therefore, a reasoner will, based on restrictions, not distinguish between the sensor class and its superclass.

During this exercise, we also identified some semantic inconsistencies leading to problems like the "one-to-many" mapping. For example the class *DoubleValuedState* has the restriction minimum of two *StateValues* instead of exactly two. Thus, entities with three *StateValues* would fit the restrictions of being a *TripleValuedState* and also a *DoubleValuedState*.

The solution to overcome these problems is the already mentioned user interaction. Of course this could lead to subjectivity, but regarding the domain of the tool as the smarthome of the user itself, a customized or subjective view on the objects in it and therefore its digital modeling seems justifiable. Another benefit of the user interaction is, that the mapping does not only work for straightforward profiles but also for more specific ones. The learning behavior of our tool also helps to improve the automated process, profile by profile.

VII. Conclusion

This paper presents a method serving two main goals: firstly, a completely information-preserving mapping of syntactically described EnOcean device profiles provided by a commercial EnOcean Gateway in JSON format onto the DogOnt ontology. Secondly, based thereon, an algorithm for the generation of semantically enriched ISO/IEC 24752 URC standard-conform User Interface Socket descriptions. The method has been fully implemented and the tool processes successfully all ~150 EnOcean profiles provided by the used gateway. The tool opens up a whole range of possible extensions. Perhaps most prominent, given a Smart Home installation based on the Smart EnOcean Gateway and the Universal Control Hub middleware, new EnOcean devices will be automatically integrated into the Smart Home environment.

Overall, this work lifts the integration of EnOcean devices from the data and information layer into the knowledge layer as described in [5]. This makes a future proof usage of the EnOcean world possible within the semantic IoT.

The OpenURC universe benefits from the semantic enrichment of the UIS in many aspects. For instance, the UCH respectively its UIS can be interpreted by Smart Services which is essential for the use of the UCH in the scope of the DiDiER project [21]. Semantically enhanced UIS descriptions significantly increase the quality of automatically generated user interaces. In the near future, we will make the results and resources available by publishing it within the scope of the OpenURC Alliance [20] and the method for the semantification of User Interface Sockets developed in the context of this work will be used in the DiDiER project.

To this end, our method has so far been based on structural mapping techniques rather than on semantic ones and the power of the description language OWL itself. In the future, we plan to improve our method accordingly, we are confident that the inclusion of more elaborate devices will support this vision.

Acknowledgment

This research is partly funded by the German Federal Ministry of Education and Research (01FJ15113). The responsibility for this publication lies with the authors.

References

[1] M. Weiser, "The computer for the 21st century," Scientific american, vol. 265, no. 3, 1991, pp. 94–104.

[2] Gartner, "Gartner Says 8.4 Billion Connected "Things" Will Be in Use in 2017, Up 31 Percent From 2016," Feb 2017. [Online]. Available: http://www.gartner.com/newsroom/id/3598917 (Date last accessed 23-April-2017).

[3] "Information technology – Home electronic systems (HES) architecture – Part 3-10: Wireless short-packet (WSP) protocol optimized for energy harvesting – Architecture and lower layer protocols," International Organization for Standardization, Geneva, CH, Standard, March 2012.

[4] DigitalConcepts, "Smart EnOcean Gateway." [Online]. Available: http://enocean-gateway.eu/produkt/ (Date last accessed 13-April-2017).

[5] J. Rowley, "The wisdom hierarchy: representations of the DIKW hierarchy," Journal of information science, vol. 33, no. 2, 2007, pp. 163–180.

[6] P. Barnaghi, W. Wang, C. Henson, and K. Taylor, "Semantics for the Internet of Things: early progress and back to the future," International Journal on Semantic Web and Information Systems (IJSWIS), vol. 8, no. 1, 2012, pp. 1–21.

[7] "Information Technology – User interfaces – Universal Remote Console – Part 1: General framework." International Organization for Standardization, Geneva, CH, Standard, December 2014.

[8] S. Peter, "Abbildung relationaler Daten auf die Ontologie des CIDOC CRM," Master's thesis, Otto-Friedrich-Universität Bamberg, 2015.

[9] P. Nussbaumer, B. Haslhofer, and W. Klas, "Towards model implementation guidelines for the CIDOC conceptual reference model," University of Vienna, Tech. Rep., 2010.

[10] T. R. Gruber, "Toward Principles for the Design of Ontologies used for Knowledge Sharing?" International journal of human-computer studies, vol. 43, no. 5-6, 1995, pp. 907–928.

[11] D. Bonino and F. Corno, "Dogont-ontology Modeling for Intelligent Domotic Environments," The Semantic Web-ISWC 2008, 2008, pp. 790–803.

[12] J. Frey, "ASaP - Integrationsplattform für Smart Services in Intelligenten Umgebungen," Ph.D. dissertation, Universität des Saarlandes, Postfach 151141, 66041 Saarbrücken, 2015. [Online]. Available: http://scidok.sulb.uni-saarland.de/volltexte/2015/6206

[13] S. Bechhofer, "Owl: Web ontology language," in Encyclopedia of Database Systems. Springer, 2009, pp. 2008–2009.

[14] J. Frey, C. Husodo-Schulz, R. Neßelrath, V. Stein, and J. Alexandersson, "Towards pluggable user interfaces for people with cognitive disabilities." in HEALTHINF, 2010, pp. 428–431.

[15] G. Zimmermann and G. Vanderheiden, "A dream… The Universal Remote Console," in ISO Focus+, February 2010, pp. 11–13.

[16] openURC, "OpenURC." [Online]. Available: http://www.openurc.org/index.php/learn-more/urc (Date last accessed 28-April-2017).

[17] R. Shearer, B. Motik, and I. Horrocks, "HermiT: A Highly-Efficient OWL Reasoner." in OWLED, vol. 432, 2008, p. 91.

[18] M. Horridge, N. Drummond, J. Goodwin, A. L. Rector, R. Stevens, and H. Wang, "The Manchester OWL syntax." in OWLed, vol. 216, 2006.

[19] P. Shvaiko and J. Euzenat, "A Survey of Schema-Based Matching Approaches," in Journal on Data Semantics IV, ser. Lecture Notes in Computer Science, S. Spaccapietra, Ed. Springer Berlin Heidelberg, 2005, vol. 3730, pp. 146–171.

[20] "Technical committee procedures (standing document)." [Online]. Available: http://www.openurc.org/SD/tc-procedures-20160915/ (Date last accessed 23-Mai-2017).

[21] P. Elfert, M. Eichelberg, J. Tröger, J. Britz, J. Alexandersson, D. Bieber, J. Bauer, S. Teichmann, L. Kuhn, M. Thielen et al., "Didier-digitized services in dietary counselling for people with increased health risks related to malnutrition and food allergies," in Computers and Communications (ISCC), 2017 IEEE Symposium on. IEEE, 2017, pp. 100–104.

 ISBN: 978-1-61208-601-9