Explain Yourself: A Semantic Annotation Framework to Facilitate Tagging of Semantic Information in Health Smart Homes

Bastian Wollschlaeger[a], Elke Eichenberg and Klaus Kabitzsch
Chair of Technical Information Systems, Technische Universität Dresden, 01062 Dresden, Germany

Keywords: Ambient Assisted Living, Digital Personalized Health, Health Smart Home, Interoperability, Taxonomy, Telehealth, Telemedicine.

Abstract: Health Smart Homes (HSH) are a key concept in future personalized health care. However, with an abundance of heterogeneous assistance components available, the design process of HSHs is becoming increasingly complex and needs to rely on computer-based support. As a key prerequisite, formal models of information semantics are required for component selection and early-on interoperability assessment. This paper proposes a flexible and extensible framework of semantic annotations that copes with the interdisciplinary nature of HSH and can be used to incorporate a formal model of semantics at component interfaces. The framework consists of a taxonomy of semantic annotations (tags) and their relationships and is developed using a well-established taxonomy creation method. The tags address several independent aspects of semantics, increasing the expressiveness of information semantics. With this valuable new level of detail for information, component models can be semantically enriched, in turn enabling computer-based design algorithms to carry out the complex design tasks in the future.

1 INTRODUCTION

Health Smart Homes (HSH) are a key concept for realizing personalized health care in order to cope with future challenges for health care systems (Maeder and Williams, 2017). However, with an abundance of heterogeneous assistance components available, the design process for HSH is becoming increasingly complex and requires computer-based support (Haux et al., 2016; Welge et al., 2015). In order to allow algorithms to identify meaningful design suggestions, formal functional models for HSH components are required to select appropriate and compatible components (Wollschlaeger et al., 2019). However, achieving *interoperability* is a complex problem (Vermesan et al., 2014), resulting in high follow-up costs when neglected in the design phase. As an example, NIST reported annual costs of $15.8 billion for the U.S. building industry (Gallaher et al., 2004). To detect potential interoperability issues as early as possible, it is necessary to include the semantics of information exchanged by components in their functional models, enabling a conceptual interoperability assessment at design time by comparing the resulting semantic data types. While syntactical interoperability can be achieved by middleware approaches and standardizing the data types of programming languages or the payloads of communication technologies, semantic information describing the actual meaning of data is often neglected. This *semantics of information* in the context of *automated design processes* for HSH will be this paper's object of investigation with the aim to provide a *modeling framework* for semantic data types. In this context, semantic data types are classes of equal meaning of information. Subsequently, main requirements for the framework will be discussed.

Despite the importance of semantic data types in achieving semantic interoperability in both the design and operation phase of HSH, the semantics of data is nowadays often not formally specified, so no automated processing of their meaning is possible. Even if some approaches offer a vocabulary of semantic types that enable a basic annotation of semantics (VDI 3813-2, 2011), these vocabularies are usually of low expressiveness. The automated processing of semantic information is limited to a comparison of data type names only. Therefore, it is still an open problem to provide a high model expressiveness, leading to REQUIREMENT I) *explicability*.

During the design process, semantic data types are

[a] https://orcid.org/0000-0002-1101-0680

Wollschlaeger, B., Eichenberg, E. and Kabitzsch, K.
Explain Yourself: A Semantic Annotation Framework to Facilitate Tagging of Semantic Information in Health Smart Homes.
DOI: 10.5220/0008968901330144
In *Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2020) - Volume 5: HEALTHINF*, pages 133-144
ISBN: 978-989-758-398-8; ISSN: 2184-4305

relevant in two different levels of detail: firstly, they are used to model an abstract system specification without usually taking into account aspects of data representation. Secondly, they are used for detailed modeling of the required and provided data for specific components. These different levels of abstraction render it difficult to assess how well a specific component fits to the abstract specification. In addition, the level of abstraction is fixed for existing approaches, i. e. it is not possible to use the same vocabulary for abstract specification and detailed component modeling. This leads to the open problem of supporting different levels of details within the same approach; resulting in REQUIREMENT II) *comprehensiveness*.

As the interdisciplinary domain of HSH is highly diverse and dynamic, a modeling framework for annotating semantics to data types needs to be flexible and extensible. Existing vocabularies of semantic types are usually not extensible, but rather fixed and limited. Therefore, the open problem to be able to account for future changes in the dynamic domain of HSH leads to REQUIREMENT III) *extensibility*.

In order to meet the requirements outlined above, the modeling framework developed by this paper will use the concept of semantic annotations (shortly referred to as *tags*) in multiple dimensions encompassing different aspects of semantics. All in all, this paper proposes the following main contributions:

1. Proposal of an *ontology structure for semantic annotation frameworks* meeting requirements I) to III).
2. Description of a *systematic and reproducible method* that can be used to develop semantic annotation frameworks for different domains.
3. Development of an *exemplary semantic annotation framework for room automation aspects* of HSH that is extensible to further aspects of HSH.

This paper is structured as follows: The next section discusses related work for HSH and semantic annotations of data. Section 3 extends a conceptual core framework of HSH to provide context for semantic annotations in assistance systems. Section 4 describes the development of the proposed semantic annotation framework for room automation aspects of HSH, including a possible extension of the developed framework to cover further aspects of HSH. The framework is validated in Section 5 by discussing its potential use in a variety of application areas. Finally, the paper is summarized by Section 6, which also provides further research areas.

2 STATE OF THE ART

2.1 Engineering of Health Smart Homes

The concept of HSH addresses the living environment of patients and aims at both improving the quality of life and increasing the safety of the occupants. To achieve this, HSH builds upon smart home technology to combine technical equipment with health care applications (Rialle et al., 2002; Maeder and Williams, 2017). Consequently, HSH are the nexus of domains such as smart home, ambient assisted living and room automation technologies (Wollschlaeger and Kabitzsch, 2019).

A conceptual core framework from (Wollschlaeger and Kabitzsch, 2019), which is depicted in Figure 1 in UML-notation, shows the main entities and their relationships for HSH. The left side is constituted by the entities of the *requirements view* and defines the demands (*User Requirements*) of the HSH *Occupant*. On the right side, the entities of the *components view* model the HSH *Assistance System* and its respective *Assistance Components*. Thus, the HSH's capabilities in providing assistance for the occupant are defined. The concept *Assistance Function* mediates the demands and the capabilities by providing a shared vocabulary for both the user requirements' functional intentions and the assistance system's functionality.

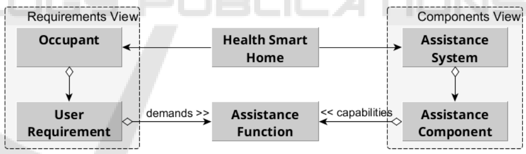

Figure 1: Core concepts of the conceptual framework for HSH, from (Wollschlaeger and Kabitzsch, 2019).

In order to achieve the high level of customization required for assistance systems in HSH (Maeder and Williams, 2017), computer-based support is necessary. (Wollschlaeger and Kabitzsch, 2020) formally defines the HSH engineering task as well as an approach for automated engineering of HSH. This approach is motivated by a prototype of an automated engineering approach in the adjacent domain of room automation (Dibowski et al., 2010). Functional models for components aimed to facilitate the design process have been investigated in different levels of detail (Dibowski and Kabitzsch, 2011), (Wollschlaeger et al., 2019). Key vocabulary for functionality was the German room automation standard VDI 3813 (VDI 3813-2, 2011), which introduces standard functionality and information types available in room automation systems on an abstract and technology-neutral level. As representation for functionality, it defines function blocks and the information required and provided by them. However, the semantic data types used in the prototypes were limited to a fixed set.

2.2 Semantic Data Types and Tags

A previous literature analysis revealed that currently there is no comprehensive standard that can be used as a common knowledge base for functionalities and data types in the whole HSH domain (Wollschlaeger and Kabitzsch, 2019). However, when only considering the room automation aspects of HSH, the *functional standard* VDI 3813 can be regarded as such a common understanding of functionality and information types. It defines a set of 46 semantic types, which each are semantically labeled with one mnemonic name consisting of two components hinting at the *type of information* represented and its *meaning*(e. g. A_SUN representing the angle of the sun's position). Thus, a coarse structure for attributing semantics to the information is available; yet, the two components are often not clearly delimitated and further details are only mentioned as unstructured explanation text. The validity of interoperability assessment based solely on the data type names is therefore limited.

As a different approach, *system information models* provide means to formalize specific technical system configurations. There are plenty of approaches for building automation systems (Butzin et al., 2017) and smart homes (Grassi et al., 2013), such as DomoML (Sommaruga et al., 2011), the SSN-Ontology (Compton et al., 2012), DogOnt (Bonino and Corno, 2008), ThinkHome Ontology (Kofler et al., 2012), AAL-Onto (Welge et al., 2015), or SAREF (Daniele et al., 2015). They focus on the detailed modeling of environment and context, but mostly not on health-related aspects. Though these approaches provide generic high-level views of HSH, they do not model sufficient information to attribute semantics to information for design purposes.

Similarly, abstract *reference models* such as the UniversAAL platform (Tazari et al., 2012) provide a conceptual framework for systems in general, but remain too coarse on the aspect of data type semantics. The ISO/IEEE 11073 standards family (ISO/IEEE 11073-10101a, 2015) offers a detailed nomenclature for health information, yet remains largely focused on communication in a hospital setting. *Middleware platforms* aim at translating data between different entities with possibly heterogeneous data models. The ESKAPE platform (Pomp et al., 2017) lifts this translation the level of data to the level of information by offering means for semantic data integration. It enables the integration of custom semantic models into a dynamic semantic knowledge graph, thus enabling querying of information agnostic of the specific data formats. However, the platform provides no guidance on which data should be semantically modeled and how the creation of the custom semantic models should be conducted, resulting in potentially incomplete and incompatible semantic models.

Another class of approaches for modeling the semantics of data are *semantic tagging approaches*. The most prominent example from the field of building automation is Project Haystack[1], which aims at modeling the semantics of building automation components. To this end, more than 200 semantic annotations (tags) are available as a list or as an ontology (Charpenay et al., 2015). This enables to create a system model that is organized along the hierarchies of site, equipment and (data) point. The flexibility and community-based approach of Haystack has inspired further work in the domain of IoT, e. g. the extended data model *BACnet XD* by the BACnet group (Butzin et al., 2017), the *KNX Information Model* by the KNX Association (Pelesic et al., 2016), or for integration of IoT and building systems (Schachinger et al., 2017).

In order to improve the applicability of tagged system models, BRICK (Balaji et al., 2016) introduced a *metadata schema* for buildings aimed at enabling applications such as model-predictive control or fault detection and diagnosis. BRICK offers an ontology of entities (Point, Equipment, Location, Measurement) and their relationships in order to model building automation systems and components.

These approaches aim at facilitating building analytics by adding semantic information to models of building systems. While capturing the systems' structure and metadata, they are focused on the operation phase and do not intend to use the semantic information in the design phase. In contrast to the available coarse granularity of data semantics, design tasks require a higher level of expressiveness such as shown by (Wollschlaeger et al., 2019) for interoperability assessments.

2.3 Taxonomies, Ontologies and Taxonomy Creation Methods

A *taxonomy* is an hierarchical categorization system consisting of dimensions, which each can have different values (characteristics). Borrowing from(Nickerson et al., 2013), a taxonomy T can be defined as

[1] http://project-haystack.org/

$$T = \{D_i, i = 1, \ldots, n \mid D_i = \{C_{ij},\ j = 1, \ldots, k_i;\ k_i \geq 2\}\} \tag{1}$$

with D_i being the dimensions and C_{ij} their characteristics. *Ontologies* extend the concept of taxonomies from an hierarchical categorization towards a network of information and logical relationships. Due to the relationships between the concepts, the strict hierarchical structure of a taxonomy is not sufficient for representing it. Instead, a knowledge graph is required.

Taxonomies and ontologies are often used for structuring concepts in a specific domain of interest. A reproducible method for developing these structures promotes re-use of the categorizations. In the field of Information Systems Research, however, several analyses pointed out that most taxonomies are developed in an ad hoc and thus non-reproducible manner (Nickerson et al., 2013; Lösser et al., 2019). In order to increase the rigor in taxonomy development and to provide guidance for researchers during the taxonomy design process, (Nickerson et al., 2013) proposed a generic high-level method for taxonomy development. Their proposed method uses meta-characteristics as a conceptual kernel and allows for an iterative taxonomy design process offering both a bottom-up (inductive) and a top-down (deductive) approach (cf. Figure 2).

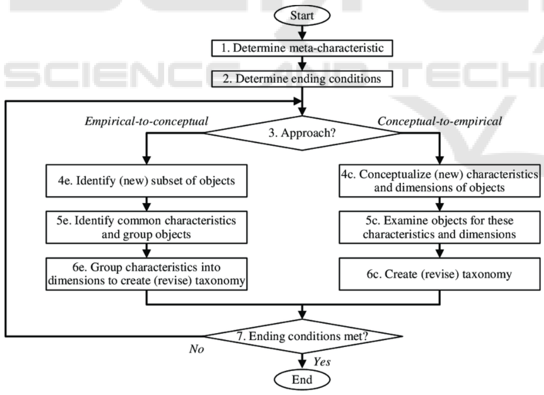

Figure 2: Method for Taxonomy Creation according to (Nickerson et al., 2013).

With the flexibility of choosing both inductive as well as deductive approaches, this taxonomy development method fits for creating the semantic annotation framework. This is due to the fact that on the one hand, some semantic data types have been defined in the domain (favoring the inductive approach) and on the other hand, a vast conceptual knowledge is available (facilitating the deductive approach). The iterative nature of the method allows for a combination of both approaches to reach the best trade-off between structuring and adhering to the existing types.

3 SEMANTIC DATA MODELS IN HSH CONTEXT

In order to put semantic data types into context, a conceptual framework of the assistance system aspects of HSH will be used. Based upon the conceptual core framework from (Wollschlaeger and Kabitzsch, 2019) (cf. Figure 1) and the UniversAAL reference model (Tazari et al., 2012), the concepts and relationships of the assistance systems are depicted in Figure 3 in UML notation.

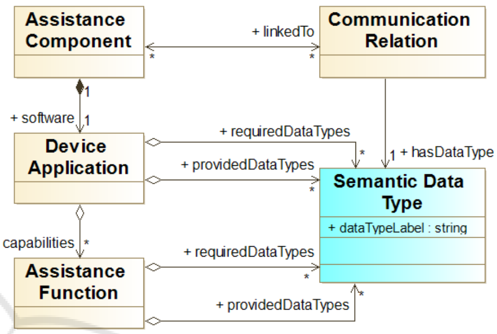

Figure 3: Conceptual Model of Assistance Systems in HSH context.

The classes on the left model `Assistance Component` as structures for equipment, which contains a software application, modeled by `Device Application`. The software application provides a certain functionality modeled by `Assistance Function`. The classes on the right side – `Communication Relation` and `Semantic Data Type` – are used to model the interconnections of assistance components in the assistance system. More specifically, the assistance components are communicating with each other in order to fulfill their functionality. A communication relation models one of the information exchanges for a set of "linked" assistance components. Each communication relation is used to exchange a specific type of information between the assistance components' software applications, modeled by its data type. On the other hand, these software applications need to provide information about their software interface. More specifically, they need to model *required data types* on the input side as well as *provided data types* on the output side.

The key concept of *interoperability* can now be investigated: the software applications exchanging information in the context of a specific communication relation need to adhere to its semantic data type. More specifically, the data types of the provided and the required side need to be *compatible* with each other.

The types are said to be compatible if the provided types are equal to or a subclass of the required types.

4 SEMANTIC ANNOTATION FRAMEWORK

The following section shows the development of the semantic annotation framework in more detail. First, the overall structure of the framework is discussed. Then, the taxonomy creation method of (Nickerson et al., 2013) is applied and descriptions of the different iterations and the decisions made in the taxonomy creation process are provided. When describing the taxonomy creation, we will use the numbers of the different *steps* according to Figure 2 for reference.

4.1 Structuring Semantic Data Annotations

The main idea for structuring the semantic annotation framework is based on describing semantics along several different information dimensions. This is motivated by different approaches of semantic system modeling for building analytics, such as the Semantic Sensor Network Ontology, the semantic tagging approach *Project Haystack*, and the BRICK metadata schema (cf. Section 2.2). Those approaches provide detailed system models by taking entities and their relationships into account. We aim to translate such expressiveness to the semantic information modeling by providing different aspects of semantics. Such a composite semantic model allows for expressing information and relations in much more detail than semantic models consisting of only one semantic label.

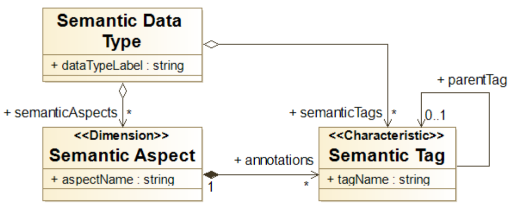

Figure 4: Model of the Semantic Annotation Framework.

Following the structuring approach outlined above, Figure 4 depicts a model of the proposed framework. A `Semantic Data Type` is composed of several different `Semantic Aspects`, which each are annotated with `Semantic Tags`. When considering the semantic annotation framework as a taxonomy, the aspects are taxonomy dimensions with the different annotations making up the characteristics of these dimensions. Since taxonomies are structurally less complex than ontological knowledge graphs, we start with a development process geared for taxonomy creation (Nickerson et al., 2013) and evaluate the taxonomy's expressiveness at each iteration in order to decide whether an extension (e. g. by using the *parentTag* relation in Figure 4) towards an ontology structure is beneficial.

4.2 Meta-characteristics and Ending Condition

At the beginning of the taxonomy creation process, the Nickerson method requires to define meta-characteristics and ending conditions for this process. Meta-characteristics are used as the basis for choosing the dimensions in the taxonomy. As the process itself is iterative, the ending conditions are required to determine, at which point the taxonomy creation process may terminate, i. e. the resulting taxonomy fulfills the intended goal (Nickerson et al., 2013).

For *step 1*, we choose meta-characteristics that refer to the orthogonal aspects of i) which entity is described, ii) in which context, and iii) using which realization. The last meta-characteristic is closely related to syntactic aspects of how information is encoded; however, since it models semantic aspects, it will only entail realization aspects on an abstract level. We define the meta-characteristics as follows (cf. Table 1):

Table 1: The meta-characteristics (MC) chosen in *step 1* of the taxonomy creation process.

ID	NAME	EXPLANATION
MC1	Entity	What is the information referring to?
MC2	Context	In which context is the information generated and used?
MC3	Realization	How is the information encoded?

Several possible pre-defined ending conditions for *step 2* are defined by Nickerson, divided into objective and subjective ending conditions (Nickerson et al., 2013). The conditions depicted in Table 2 were selected for the development process.

4.3 First Iteration

The first iteration starts with the decision, if the top-down (conceptual-to-empirical) or bottom-up (empirical-to-conceptual) approach should be used (*step 3*). Since various domain knowledge is available

Table 2: The ending conditions (ECO / ECS) chosen in *step 2* of the taxonomy creation process.

ID	NAME	EXPLANATION
		OBJECTIVE ENDING CONDITIONS
ECO1	Comprehensive Modeling	Semantic data types of target domain are comprehensively modeled (all objects have been examined)
ECO2	Objects stable	No object was merged or split in the last iteration
ECO3	Important concepts identified	No new dimensions or characteristics were added in the last iteration
ECO4	Concepts stable	No dimensions or characteristics were merged or split in the last iteration
ECO5	Dimension Uniqueness	Every dimension is unique and not repeated (i. e. there is no duplicate)
ECO6	Characteristic Uniqueness	Every characteristic is unique within its dimension
		SUBJECTIVE ENDING CONDITIONS
ECS1	Conciseness	The number of dimensions limited to at most 7 ± 2
ECS2	Robustness	Sufficient differentiation amongst the objects is feasible
ECS3	Comprehensiveness	REQUIREMENT II) All objects of interest can be classified
ECS4	Extensibility	REQUIREMENT III) New dimensions / characteristics can be added easily
ECS5	Explicability	REQUIREMENT I) The dimensions / characteristics are able to explain the objects

that may serve as a conceptual guideline for structuring semantic models, we chose the *conceptual-to-empirical* approach with the data types of the VDI 3813 as the initial set of objects. At this point, we have to restrict the objects under investigation to the room automation aspects of HSH, as further aspects such as assistance technology still lack a common understanding of semantics (Wollschlaeger and Kabitzsch, 2019). We envisage to broaden the perspective towards further aspects of HSH once the conceptual structure has been consolidated. Section 4.6 demonstrates the framework's extensibility by discussing health-related HSH examples.

Since this is the first iteration, *step 4c* conceptualizes new dimensions and characteristics based on the conceptual kernel of meta-characteristics. In order to achieve an easily extensible taxonomy, we follow the approach of the SSN-Ontology (Compton et al., 2012) and deduct the two new dimensions FEATURE OF INTEREST and PHYSICAL QUALITY from MC1 *Entity*. Their characteristics can be determined by investigating potential entities and properties in a room. Entities involve equipment (*climatisation*, *contact*, *damper*, *drive*, *fan*, *lamella*, *lamp*, *sunblind*, *valve*, *vav*[2]), and physical entities (*air*, *light*, *occupant*, *sun*, *water*), distinguishing if the entity can be directly controlled by technical means (equipment) or not (physical entity). On the other hand, each entity possesses different properties, constituting a list of PHYSICAL QUALITIES.

[2]Variable Air Volume, a type of ventilation system

The second meta-characteristic MC2 *Context* is used to infer two further dimensions LOCATION and VARIABLE TYPE to model the information origin and usage, respectively. The LOCATION dimension contains characteristics describing the location type of the information's origin, namely *indoor*, *outdoor*, *bms* (building management system), and *equipment* (if the information is used in the technical systems only).

The VARIABLE TYPE dimension models the intended usage of the contained information. Since the processes in room automation may be modeled as control loops, the VARIABLE TYPE dimension uses characteristics classifying data based on its usage in a standard control loop. Figure 5 depicts the different characteristics that were identified using the international electrotechnical vocabulary for control theory (IEC 60050-351, 2013). The types of variables can be differentiated into operating values (*command* and *setpoint*), *computed* values (i. e. controller output) and measured values (*state* and *value*).

Finally, meta-characteristic MC3 *Realization* allows to infer abstract realization characteristics. The dimension TRIGGER TYPE models that information can be created by a *manual* human action (e. g. choosing temperature set point or the push of a button) or by algorithms in an *automatic* way (e. g. the blind position following the movement of the sun). This distinction often carries further semantic information with respect to prioritizing the commands, as most often the command issued by humans can override automatically generated commands.

In addition, the dimension REFERENCE TYPE models if a value is to be interpreted as an *absolute* or a *relative* value. In case of relative values, a base value is required in order to compute the actual value to work with. This distinction is important for e. g. setting the temperature set point, as most room control units only provide means to adjust the set point within $\pm 5K$, while others allow a specific temperature as set point.

Finally, the dimension REPRESENTATION allows to specify the general type of information encoding in terms of the scale of measurement. It distinguishes amongst *binary*, *quantified*, *continuous*, *nominal*, and *ordinal* values. Nominal and ordinal values represent enumeration types with ordinal values additionally featuring a partial order.

Having identified the initial set of dimensions and characteristics, *step 5c* examines existing semantic data types as empirical objects for these dimensions and characteristics. The 46 types of the VDI 3813 were annotated semantically. If multiple semantic annotations for each dimension were possible, they were assigned simultaneously. However, this strongly indicates that the semantic type in question is *ambiguous*, i. e. defined too vaguely as it can have multiple meanings, preventing this type to be used in an interoperability assessment.

With the results of the first iteration, the resulting vocabulary (*step 6c*) consists of the seven dimensions and 53 characteristics. Different from Nickersons original method, we put major emphasis on REQUIREMENT III) *extensibility* (e. g. new equipment, new considered physical entities, new forms of representation, . . .), so we do not eliminate dimensions and characteristics that are not yet used, as long as they are potentially useful. As an example, the dimensions TRIGGER TYPE, REFERENCE TYPE and REPRESENTATION are usually not specified at the abstract level of the VDI 3813, but they can be used to further model the semantics of specific component interfaces (cf. (Wollschlaeger et al., 2019)). For this reason, we did not eliminate these dimensions, but keep them as extension points for more detailed modeling. By allowing information dimensions to be unspecified, subclass-relationships amongst the semantic types can be inferred quite easily. This enables to model semantic types with different levels of abstraction, e. g. more abstract types for system specification and specific types for functional component modeling (cf. REQUIREMENT II) *comprehensiveness*).

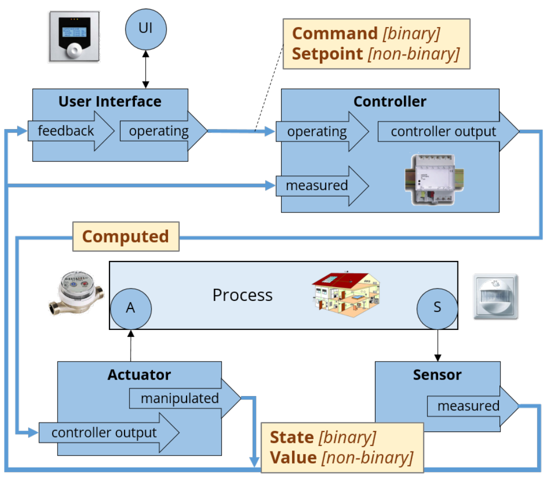

Figure 5: The different Variable Types (bold in boxes) in a closed-loop control system. The functionalities are displayed as function blocks according to (VDI 3813-2, 2011).

After the first iteration, *step 7* tests if the ending conditions have been met. From the objective ending conditions, ECO3 is not met since new dimensions and characteristics were added. As not all ending conditions have been met, at least one more iteration is needed.

4.4 Second Iteration

At the beginning of the second iteration in *step 3*, a decision for the top-down or bottom-up approach needs to be made. Since we identified several ambiguous data types in iteration 1, we use the bottom-up approach in this iteration to investigate new and more specific data types.

In *step 4e*, we identified missing data types that will be added to the set of empirical objects. In total, 29 new data types have been identified, including three new data types that were missing until now (frost state, rain alarm, wind alarm). The further 26 data types resulted from introducing new, more specific data types for the ambiguities of D_ACT (containing both date and time), L-types (describing lamp activation or level), S-types (addressing sunblind or lamella), as well as V_SET and V_STA (modeling either drive position, fan rotation, or air flow). The former ambiguous data types were kept as “abstract data types” and for compatibility with the standard.

In order to semantically annotate the new data types, we adapted the vocabulary in *step 5e*. More specifically, we added new characteristics for entities (*calendar*, *controller*, *hvac*, *room*, *shading*), qualities (*frost*, *humidity*, *intrusion*, *pressure*), and representation (*complex*). In addition, we renamed entity *vav* to *ventilation* and restructured some of the qualities (*azimuth* to *angle.azimuth*, *elevation* to *angle.elevation*, *maintenance* to *maintenance.position*, *protection* to *protection.position*, *mode_ventilation* to *mode*) in order to better grasp the real-world relationships. There was no need to adapt the dimensions themselves.

At the end of iteration 2 (*step 6e*), the semantic annotation framework consists of 63 annotations in seven categories. It was successfully used to model 75 semantic data types.

After the second iteration, *step 7* tests if the ending conditions have been met. In order to test the robustness (ECS2) of the taxonomy, we modeled the dimensions, characteristics and also data types in an OWL ontology in order to use the reasoner for consistency checks and subsumption analysis. We modeled the data types as *defined classes* requiring annotations according to their respective characteristics. As an example, we can define the room brightness H_ROOM as a data type that contains information about the FEATURE OF INTEREST *light* and its PHYSICAL QUALITY *illuminance*, referring to an information about an *indoor* (LOCATION) *value* (VARIABLE TYPE). Usually, this value is represented as an *absolute* value (REFERENCE TYPE). In $\mathcal{EL}^{++}$ notation (Baader et al., 2005), this yields:

$$\begin{aligned} \mathit{H\_ROOM} \equiv & \mathit{DataType} \\ & \sqcap (\exists\ \mathit{hasFeatureOfInterest.Light}) \\ & \sqcap (\exists\ \mathit{hasQuality.Illuminance}) \\ & \sqcap (\exists\ \mathit{hasLocation.Indoor}) \\ & \sqcap (\exists\ \mathit{hasVariableType.Value}) \\ & \sqcap (\exists\ \mathit{hasReferenceType.Absolute}). \end{aligned} \tag{2}$$

The reasoner confirmed the overall consistency of the ontology and inferred a subsumption hierarchy of types. Most of the hierarchical relations were correct, but the following issues could be identified: First, the shading data types have no connection to each other; however, e. g. S_AUTO should be a super type of S_AUTO_SBL (sunblind) and S_AUTO_LML (lamella). Secondly, T_SUPPLY is inferred to be a subclass of V_STA, which may contain positions, flows, or rotations, but not temperatures. This is due to the too lax definition of V_STA.

From the objective ending conditions, ECO3 is not met, since new dimensions and characteristics were added. ECO2 is also not met since some data types were split. As not all ending conditions have been met, at least one more iteration is needed.

4.5 Third Iteration

Again, the type of approach has to be defined in *step 3* for this third iteration. In the previous iteration, two issues were identified by using reasoning: On the one hand, some relationships were missing (e. g. that the data types for shading are related) and on the other hand, some definitions were too broad, encompassing non-related types (e. g. V_STA wrongly subsuming T_SUPPLY). Since these main issues refer to the semantic definition of existing data types, we choose the top-down approach to adjust the characteristics.

In *step 4c*, we chose to adapt the generic room model, which was used to specify the dimensions FEATURE OF INTEREST and PHYSICAL QUALITY in the first iteration, in order to add the missing relationships. Therefore, we specified that both characteristics *sunblind* and *lamella* are sub-classes of the *shading* system and that the *climatisation* and *ventilation* characteristics are sub-classes of the *hvac* system.

Since the characteristics in one dimension of the taxonomy are only lists, we included these hierarchical relations in the semantic annotation framework ontology, but did not change the taxonomic structure. At this point, the transition from taxonomy towards ontology seems beneficial.

In *step 5c*, we modified the annotations for those types that were defined too broadly. As an example, the type V_STA had neither annotations from the FEATURE OF INTEREST and PHYSICAL QUALITY dimensions in order to include its possible information types (drive position, fan rotation, air flow). We defined multiple annotations in previously empty dimensions to include all sub-types while excluding other non-related types. This was done for the types V_STA and V_SET, D_ACT (to include the two possible properties *date* and *time*) and the L-Types (include two possible properties *activation* and *level*).

As we did not change the dimensions or characteristics in this iteration, we do not need to revise the taxonomy in *step 6c*. We added relationships between the characteristics so that a reasoner can automatically infer additional relations between the data types. By doing so, we go beyond a taxonomic structure, which is not able to capture relationships amongst the characteristics. Instead, this renders the semantic framework to an ontology that is largely based on the developed taxonomic structure.

At the end of the procedure, *step 7* again checks the ending conditions. Since we made no changes in the semantic annotation vocabulary and only added relationships in the ontology, the objective ending conditions are met this time. By using the additional relationships, the annotation framework could be increased in robustness. All other subjective ending conditions are also met, so that the taxonomy creation process can be concluded.

The final and valid taxonomy for semantic data types of the room automation aspects contains the dimensions and characteristics shown in Table 3.

Table 3: The final semantic annotation framework for the room automation aspects of HSH.

Dimensions	Characteristics					
FEATURE OF INTEREST	air	calendar	climatisation	contact	controller	damper
	drive	fan	hvac	lamella	lamp	light
	occupant	room	shading	sun	sunblind	valve
	ventilation	weather				
PHYSICAL QUALITY	activation	angle.azimuth	angle.elevation	date	dewpoint	flow
	frost	illuminance	intrusion	level	maint.pos	mode
	position	precipitation	presence	pressure	protect.pos	quality
	rotation	scene	temperature	time	velocity	
LOCATION	bms	equipment	indoor	outdoor		
VARIABLE TYPE	command	computed	setpoint	state	value	
TRIGGER TYPE	automatic	manual				
REFERENCE TYPE	absolute	relative				
REPRESENTATION	binary	complex	continuous	nominal	ordinal	quantified

4.6 Extension Capability towards Remaining HSH Aspects

As there are no semantic standards for further aspects of HSH yet, we cannot create a holistic semantic framework for HSH at this moment. However, we can hint at how such a framework would look like due to the extensible nature of the developed framework for room automation.

To include aspects such as health care and assistance systems into the semantic annotation framework, further iterations of the Nickerson method may extend the characteristics. Once a common understanding of further HSH functionality has been established, each of these functions can be investigated with respect to its required and provided information. These types of information would constitute a new set of semantic data types that can be annotated with the framework proposed in this paper.

With health being a major part of the missing HSH aspects, further characteristics of the occupant need to be added. This may include the body's *height* and *weight* as well as vital parameters such as *pulse*, *blood pressure*, *body temperature*, *breathing rate*. To incorporate e. g. chronic diseases, closely related characteristics like *blood sugar level* or *HbA1c* value for diabetes type 2 might be added, too.

For dementia patients with wandering behavior, a "tracking system" might be a useful functionality for a HSH to prevent the patients getting lost. In this case, the functionality would involve exchanging information about the *occupant*'s *position* (for which annotations already exist), which could also be specialized to include the GPS position components *latitude* and *longitude*. In case of the functionality "dietary assistance", information related to *food* are important, more specifically its *type* and the *amount* consumed. Similarly, the functionality "medication reminder" is processing *medication* information regarding its *type* and *amount*. This is combined with some information regarding the day time or specified as a *rate*. As a final example, some assistance systems are able to identify states of emergency based on consumption data from energy and water meters. This can be incorporated by the FEATURES OF INTEREST *energy* and *water* as well as the PHYSICAL QUALITY annotations *consumption.current* and *consumption.total*.

By discussing several examples from the assistance- and health-related aspects of HSH, it could be shown that the proposed framework is extensible towards new semantic data types from the HSH domain. New characteristics will be necessary to account for specific aspects of HSH. However, it can be expected that the structure of the framework's dimensions is expressive and robust, so most likely no changes will be required here.

5 APPLICATION AREAS AND VALIDATION

In order to validate that the proposed method for creating a semantic annotation framework has produced a valid result, this section will discuss the applicability of the resulting framework based on several application areas. For each application area, we will provide an example that illustrates how the annotation framework facilitates such applications.

Firstly, the annotation framework enables a *conceptual consistency check* of established semantic types due to its expressive semantic model. As has been discussed in Section 4.3, we were able to detect ambiguities in the established standard VDI 3813.

The most important ambiguities were the data types for lamp control – missing a distinction between switching (activation) and dimming (level) lamps, the types for shading control – not specifying whether sunblind or lamella or both are controlled – and V_SET and V_STA, which each could represent three completely different meanings.

Secondly, the structure of the framework allows for a *seamless semantic modeling* from abstract specification to detailed component models. In context of automated design approaches for HSH this means that by applying the proposed framework a consistent approach in semantic data modeling can be used. Depending on the number of dimensions used for semantic specification, the information type can be specified with varying level of abstraction. While information types in system specifications on an abstract level might only contain *core* aspects, independent of realization details, a detailed functional model of an assistance component may feature all available dimensions to model the semantic of its exchanged data as precisely as possible. Since both cases use the same structure, a relation between more abstract and very specific information types can be easily determined.

As an example, a system specification might indicate that a functionality *temperature controller* requires a temperature setpoint on the one hand and the current temperature on the other hand. Thus, both data types would be modeled as *air* (FEATURE OF INTEREST) *temperature* (PHYSICAL QUALITY) *indoor* (LOCATION). Additionally, the temperature setpoint would be tagged with the *setpoint* annotation in the VARIABLE TYPE dimension, while the current temperature would be tagged with the *value* annotation.

A device implementing such a temperature controller functionality might now additionally provide semantic annotations for its interface. If the temperature setpoint is expected to be an offset value, it could add the *relative* annotation to the REFERENCE TYPE dimension. On the other hand, if it expects a specific temperature value as setpoint, the annotation *absolute* would be more appropriate. This additional level of detail is vital to be able to detect a mismatch between the sender of the setpoint and the expected semantic of the data. If the sender is a room control unit with a rotary switch for determining the setpoint offset, it would not be compatible with a temperature controller expecting an absolute setpoint value.

Thirdly, as was hinted at by the example above, the proposed framework allows for an *improved interoperability assessment* by lifting the assessment from a mere comparison of names to the level of expressive semantics and subsumption hierarchies. For the temperature controller example above, instead of comparing the semantic types T_SETPOINT and T_SETPOINT, the more detailed versions (*air, temperature, indoor, setpoint, relative*) and (*air, temperature, indoor, setpoint, absolute*) can be compared. Only this detailed comparison enables to spot interoperability issues – when comparing the previous semantic names of the data types, issues would not have been predicted and only noticed after installation, incurring high revision costs.

The downside of using a subsumption hierarchy instead of a name-based interoperability comparison is the more difficult interoperability assessment compared to string comparisons. The effect of using a logical reasoner for interoperability assessment still needs to be investigated more closely; however, a component-wise interoperability assessment that investigates each dimension independently for interoperability could be implemented as a trade-off. As this would involve several comparisons of strings, its performance is expected to lie between single string comparison and logical reasoning. Since it would work on a taxonomic structure only, a loss of expressiveness would be traded for a better performance. In the future, the structure of the used semantic annotation framework could be investigated to determine the most efficient interoperability assessment algorithm: if the annotations form a taxonomy, the component-wise algorithm would be preferred, whereas logical reasoning is required if the framework's structure can only be represented as an ontology.

Lastly, the detailed semantic specification of the proposed framework allows for *more sophisticated analyses of semantic data types*. This new depth of understanding the semantics of a data type may be used in future to make the component model specification process more efficient and robust. This process consists of component manufacturers specifying a functional model of their components. Until now, they need to specify both the component's functionality and the semantics of its interface independently, resulting in a labor-intensive process with a high error probability. Furthermore, they are restricted to a fixed list of semantic data types. If no exactly fitting data type is available, they need to resort to the best fit, thereby increasing the modeling error. By using the proposed semantic framework, the vocabulary of available data types can be expanded strongly. If none of the pre-specified data types is fitting, new data types may be easily created by specifying the different dimensions as it seems appropriate. Unlike assigning the semantics to data types based on their names only, newly created data types are instantly understandable as the annotation framework provides a high degree of explicability.

On the other hand, the detailed semantic understanding of the data types can be used for suggestion mechanisms. Instead of choosing a data type amongst a list or manually specifying each dimension separately, the knowledge base of functionalities and their semantic interface can be used to automatically infer and suggest possible data types based on the already specified functionalities. This suggestion mechanism can also be applied in the other direction, suggesting possible functionalities based on the already semantically annotated component interface. Such suggestion mechanisms thus connect the previously independent aspects in the modeling process.

6 CONCLUSIONS

In this paper, we proposed an extensible semantic annotation framework that is able to add a significant level of semantic expressiveness to information types exchanged in information systems such as health smart homes. We applied a well-adopted method for taxonomy creation to end up with a basic structure, featuring seven dimensions and 63 characteristics for semantic annotations. Afterwards, it was extended to an ontology by introducing relationships between the annotations, which allows the use of automated reasoning to infer relations between the modeled data types. Since there are currently no common vocabularies of functionality or semantic data types for HSH as a whole, all its aspects could not yet be included in the semantic framework. While it is restricted to room automation aspects of HSH, 75 data types of the VDI 3813 could be successfully modeled. Furthermore, the feasibility of extending the semantic framework to encompass missing aspects could be demonstrated in Section 4.6 by adding new annotations while preserving the general framework structure. This fulfills REQUIREMENT III) *extensibility*).

With its aspect-oriented approach and the definition of different dimensions of semantic annotations, the approach is able to specify semantics in detail and to meet REQUIREMENT I) *explicability*, surpassing the expressiveness capabilities of using mere data type names to encode semantics. Since multiple dimensions can be taken into account, the framework also allows for a seamless semantic data modeling from abstract specification to detailed component modeling, starting from a few dimensions to fully encompass all aspects of semantic data as demanded by REQUIREMENT II) *comprehensiveness*).

We demonstrated how different applications are facilitated by the proposed semantic annotation framework – namely, it enables a consistency check on established semantic types, it improves the quality of interoperability assessments, and it lays the foundation for streamlining the process of specifying semantic models for automation components by enabling auxiliary suggestion functionalities.

As next steps, we intend to put the semantic annotation framework into action by further investigating the mentioned application areas. Improved interoperability assessment and support functions for model specification seem to be the most promising applications as of now. Furthermore, once a consolidation of functionality and data types is achieved for the whole domain of HSH, we plan to extend the framework so that is encompasses HSH as a whole. In order to integrate the semantic annotation framework with existing tagging approaches, ontology matching and merging approaches might be a viable tool to consolidation of interdisciplinary semantic vocabularies (e. g. the nomenclature of health informatics provided by the IEEE 11073).

We consider the increased semantic expressiveness offered by the annotation framework as vital for increasing the degree of automation of the design process of HSH and as an enabler for an increased proliferation of assistance technology applications in the future.

ACKNOWLEDGEMENTS

The work for this paper was funded by the European Social Fund and the Free State of Saxony (Grant no. 100310385). The authors would also like to offer special thanks to *Lena Otto* and the other members of the junior research group *Care4Saxony* for their valuable feedback on the manuscript.

REFERENCES

Baader, F., Brandt, S., and Lutz, C. (2005). Pushing the EL envelope. In *IJCAI - Int. Joint Conf. on AI*, pages 364–369.

Balaji, B., Bhattacharya, A., Fierro, G., Gao, J., Gluck, J., Hong, D., Johansen, A., Koh, J., Ploennigs, J., Agarwal, Y., Berges, M., Culler, D., Gupta, R., Kjærgaard, M. B., Srivastava, M., and Whitehouse, K. (2016). Brick: Towards a Unified Metadata Schema For Buildings. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, BuildSys '16, pages 41–50, New York, NY, USA. ACM.

Bonino, D. and Corno, F. (2008). DogOnt - Ontology Modeling for Intelligent Domotic Environments. In Sheth, A., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T., and Thirunarayan, K., editors, *The Semantic*

Web - ISWC 2008, pages 790–803, Berlin, Heidelberg. Springer Berlin Heidelberg.

Butzin, B., Golatowski, F., and Timmermann, D. (2017). A survey on information modeling and ontologies in building automation. In *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pages 8615–8621.

Charpenay, V., Käbisch, S., Anicic, D., and Kosch, H. (2015). An ontology design pattern for IoT device tagging systems. In *2015 5th International Conference on the Internet of Things (IOT)*, pages 138–145.

Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W. D., Phuoc, D. L., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., and Taylor, K. (2012). The SSN ontology of the W3C semantic sensor network incubator group. *Journal of Web Semantics*, 17:25 – 32.

Daniele, L., den Hartog, F., and Roes, J. (2015). Study on semantic assets for smart appliances interoperability: D-S4: Final report. *European Commission, Brussels*.

Dibowski, H. and Kabitzsch, K. (2011). Ontology-based device descriptions and device repository for building automation devices. *EURASIP Journal on Embedded Systems*, 2011:3:1–3:17.

Dibowski, H., Ploennigs, J., and Kabitzsch, K. (2010). Automated design of building automation systems. *IEEE Transactions on Industrial Electronics*, 57(11):3606–3613.

Gallaher, M., O'Connor, A., Dettbarn, J., and Gilday, L. (2004). Cost Analysis of Inadequate Interoperability in the U.S. Capital Facilities Industry. techreport, National Institute of Standards and Technology.

Grassi, M., Nucci, M., and Piazza, F. (2013). Ontologies for smart homes and energy management: An implementation-driven survey. In *2013 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)*, pages 1–3.

Haux, R., Koch, S., Lovell, N. H., Marschollek, M., Nakashima, N., and Wolf, K.-H. (2016). Health-enabling and ambient assistive technologies: Past, present, future. *Yearb Med Inform*, Suppl. 1(Suppl 1):76–91. 27362588[pmid].

IEC 60050-351 (2013). International electrotechnical vocabulary - Part 351: Control technology.

ISO/IEEE 11073-10101a (2015). Health informatics – Point-of-care medical device communication – Part 10101: Nomenclature Amendment 1: Additional Definitions.

Kofler, M. J., Reinisch, C., and Kastner, W. (2012). A semantic representation of energy-related information in future smart homes. *Energy and Buildings*, 47:169 – 179.

Lösser, B., Oberländer, A. M., and Rau, D. (2019). Taxonomy Research in Information Systems: A Systematic Assessment. In *Proceedings of the 27th European Conference on Information Systems (ECIS)*, Stockholm and Uppsala, Sweden.

Maeder, A. J. and Williams, P. A. H. (2017). Health smart homes: New challenges. *Studies in health technology and informatics*, 245:166–169.

Nickerson, R. C., Varshney, U., and Muntermann, J. (2013). A method for taxonomy development and its application in information systems. *European Journal of Information Systems*, 22(3):336–359.

Pelesic, I., Fernbach, A., Granzer, W., and Kastner, W. (2016). Semantic Integration in Building Automation A Case Study for KNX , oBIX and Semantic Web Applications. In *Proceedings of the KNX Scientific Conference*.

Pomp, A., Paulus., A., Jeschke., S., and Meisen., T. (2017). Eskape: Information platform for enabling semantic data processing. In *Proceedings of the 19th International Conference on Enterprise Information Systems - Volume 2: ICEIS,*, pages 644–655. INSTICC, SciTePress.

Rialle, V., Duchene, F., Noury, N., Bajolle, L., and Demongeot, J. (2002). Health "Smart" Home: Information Technology for Patients at Home. *Telemedicine Journal and e-Health*, 8(4):395–409. PMID: 12626109.

Schachinger, D., Fernbach, A., and Kastner, W. (2017). Modeling framework for IoT integration of building automation systems. *at-Automatisierungstechnik*, 65(9):630–640.

Sommaruga, L., Formilli, T., and Rizzo, N. (2011). DomoML: An Integrating Devices Framework for Ambient Intelligence Solutions. In *Proceedings of the 6th International Workshop on Enhanced Web Service Technologies*, WEWST '11, pages 9–15, New York, NY, USA. ACM.

Tazari, M.-R., Furfari, F., Fides-Valero, Á., Hanke, S., Höftberger, O., Kehagias, D. D., Mosmondor, M., Wichert, R., and Wolf, P. (2012). The universAAL Reference Model for AAL. In *Handbook of Ambient Assisted Living*.

VDI 3813-2 (2011). Building automation and control systems (BACS). Room control functions (RA functions).

Vermesan, O., Friess, P., et al. (2014). *Internet of things-from research and innovation to market deployment*, volume 29. River publishers Aalborg.

Welge, R., Busch, B.-H., Kabitzsch, K., Laurila-Dürsch, J., Heusinger, S., Lipprandt, M., Eichelberg, M., Eichenberg, E., Engelien, H., Gök, M., Moritz, G., Hein, A., and Dutz, T. (2015). AAL-Onto: A Formal Representation of RAALI Integration Profiles. In Wichert, R. and Klausing, H., editors, *Ambient Assisted Living: 7. AAL-Kongress 2014 Berlin, Germany, January 21-22, 2014*, pages 89–102. Springer International Publishing, Cham.

Wollschlaeger, B., Eichenberg, E., and Kabitzsch, K. (2019). Switching to a holistic perspective on semantic component models in building automation – tapping the full potential of automated design approaches. *IOP Conference Series: Earth and Environmental Science*, 323:012047.

Wollschlaeger, B. and Kabitzsch, K. (2019). Navigating the Jungle of Assistance Systems: A Comparison of Standards for Assistance Functionality. In *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 5: HEALTHINF*, pages 359–366. INSTICC, SciTePress.

Wollschlaeger, B. and Kabitzsch, K. (2020). Automated Engineering for Health Smart Homes: Find a Way in the Jungle of Assistance Systems (accepted). *Studies in health technology and informatics*.