Interoperable interactions of HVAC components based on a capability ontology

Nicolai Maisch [a], *Maximilian* Both [a], *Ralf* Ulmer [a], Prof. Dr. *Jochen* Müller [a]

[a] Institut für Technische Gebäudeausrüstung, Faculty of Process Engineering, Energy and Mechanical Systems, TH Köln – University of Applied Sciences, Cologne, Germany, nicolai.maisch@th-koeln.de, jochen.mueller@th-koeln.de.

Abstract. In today's building automation networks, automation functions of installed HVAC components are rigidly predefined. Increasingly hybrid HVAC systems combining different generators (e.g. heat pumps with fossil boilers) and different transmission options pose new challenges for the engineering of automation functions. Rigidly predefined automation functions lead to inflexible operating procedures and high engineering efforts when changing the system environment. The reason for this is the lack of availability of standardised digital twins and the lack of mutual informational explorability of their capabilities to enable interactions between assets without rigid automation functions. Yet if digital twins of technical assets are not semantically described in a uniform way, the semantics of their information and capabilities can be referenced to external ontologies. Semantically describing the capabilities is necessary for self-x interactions between assets to be able to take over joint functions. In the field of HVAC, there is no ontology for capabilities that can be referenced by digital twins of HVAC assets, for example, to semantically characterise their functionality within a power generation system. This paper describes the development of such an ontology and the method used to derive the key terms. The ontology also presents a framework to compose the higher-level functionalities from granular asset functions. This is to ensure that references to the respective functions of individual assets can be used to imply the functionality of their overall system. The usage of the presented ontology by digital twins of HVAC components can serve as a basis for flexible interactions between real world assets. This can help to reduce engineering effort and increase energy efficiency.

Keywords. Industrie 4.0, Capability-Based Engineering, Ontology, Semantic Interoperability
DOI: https://doi.org/10.34641/clima.2022.201

1. Introduction

The information model for digital twins of the Industrie 4.0 (I4.0) initiative - the *Asset Administration Shell (AAS)* - represents characteristics of assets in a standardised way. The information of an asset is collected within an AAS, structured according to the meta model and mapped in a machine-readable way in order to lay the foundation for self-x functions, i.e. the ability to independently explore their technical environment and interact with other assets. The central components of the AAS are *submodels*, which represent the properties and functionalities of the assets and their contents. Submodels, for their part, contain *submodel elements* that contain the actual information of the components [1].

Although the metamodel of the AAS is standardised, a semantically homogeneous environment is not guaranteed, as the submodel elements can be described in a manufacturer- or domain-specific way. Figure 1 shows the integration of semantically heterogeneously described assets into an application, which involves a lot of manual engineering because of the different semantic descriptions.

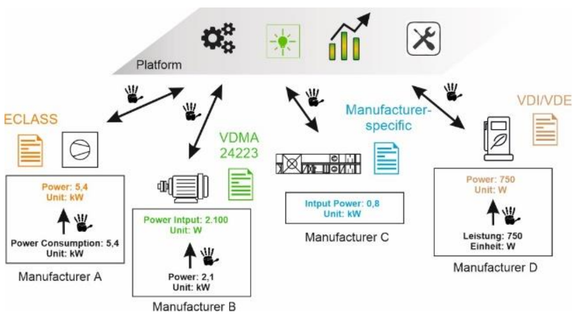

Fig. 1 - Example of manual engineering to integrate the power of semantically heterogeneously described assets

Copyright ©2022 by the authors. This conference paper is published under a CC-BY-4.0 license.

One approach to create a semantically homogeneous space is to provide a uniform standard for describing information that all producers use to describe their assets, e.g. ECLASS. ECLASS is a repository for the classification of products and their characteristics [2]. ECLASS entries can be referenced by a globally unique IRDI within AASs [1]. Another approach references submodel elements to online and generally accessible knowledge bases (*ontologies*). If a submodel element of an AAS is referenced to an item of an ontology, applications or other interacting assets can determine its description and the relationships of the item to other items [1, 3]. Linked Data is used to create links between different manufacturer- and industry-specific ontologies [4], enabling the exchange of information between semantically heterogeneously described AASs (Figure 2).

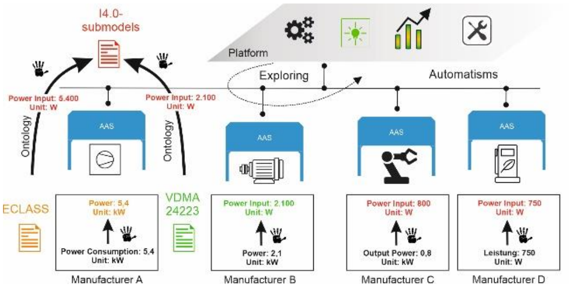

Fig. 2 - Interoperability by mapping semantically heterogeneous standards and feature descriptions to ontologies

The submodel element class *capability* describes a capability of an asset represented by the AAS [1]. Through references to ontologies, the capabilities are semantically distinguished so applications or other AASs can understand and classify them [3]. This forms the basis for the assumption of joint functions by several assets such as the heat supply of a building.

In the field of HVAC, there is a lack of an ontology that maps basic functionalities for the description of supply processes within buildings. In this paper, the development of an HVAC capabilities ontology is explained. The aim is that the ontology enables different HVAC systems within a building to interact in an interoperable and configuration-free way by exchanging their supply functions in order to reduce engineering effort.

2. Functionalities in HVAC Operation

A classification of the primary objectives in the operation of buildings is provided by [5]. The usage phase of buildings is divided into nine fields (Figure 3). In addition to administrative activities such as cost or space management, the activity of supplying and disposing of objects is particularly relevant for the area of HVAC and therefor also for building automation. *Supplying objects* is described as the supply of energy and media in an appropriate form [5].

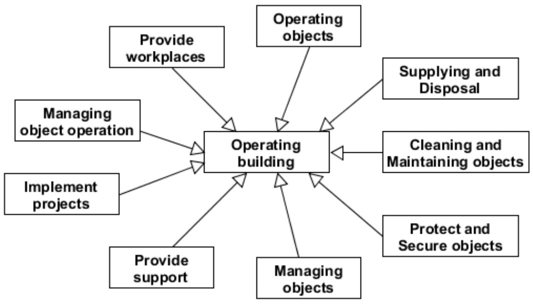

Fig. 3 - Tasks of the life cycle operating according to [5]

Consistent with [5], [6] divides building supply tasks into the energy and media usage (Figure 4). [7] divides the processes for fulfilling energy supply tasks within buildings into their basic components: Generation, distribution and transfer. The supply task with media can be subdivided analogously (Figure 4).

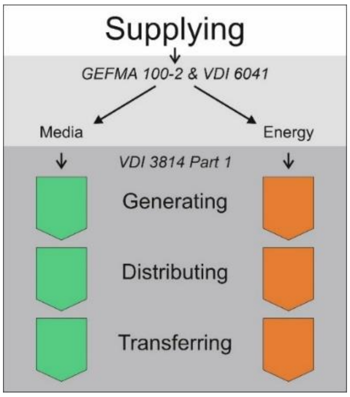

Fig. 4 - Derivation of the basic functions for supplying buildings with energy and media according to [5–7]

In addition to the primary utility functionalities within a building, there are granular functions of individual assets. [8] provides a pool of basic asset functions and describes a three-level classification and coding that defines and differentiates functional properties. Both the tasks to fulfil supply functions within buildings, such as heat generation, and the granular asset functions according to [8] can be described as capabilities in the context of I4.0 and thus assigned to assets in their AASs [1].

3. Capabilities in an AAS

The metamodel of the AAS is fundamental for the exchange of information between assets, as it provides a framework for the representation of submodel elements and thus asset information [1]. For "plug and produce" scenarios and flexible value networks in which assets independently take over common functions without configuration effort, the capabilities of assets must be represented in AASs in addition to descriptive submodel elements (*properties*) [3]. Through communication between AASs, required and existing capabilities can be matched to initiate actions. After a check of the capabilities of an asset (*capability check*), a *feasibility check* is carried out for an interaction process on the basis of the specific conditions and properties [3]. Submodel elements that are semantically linked to the pre-matched capability with the relation attribute *CapabilityRealizedBy* within the AAS serve as the information basis for the feasibility check [9].

The concept is based on the *Process, Product and Resource (PPR)* model, according to which in certain technical areas (e.g. manufacturing industry) production lines are planned and automated in a modular way. This is based on the manufacturing capabilities of individual plants. This allows reacting to changing conditions and easily replacing modules of manufacturing if necessary. [3]. To do so, it is necessary to model the manufacturing operation as an interaction of resource, process, product and capabilities. In this case, a certain manufacturing process, through which a product is created, is fulfilled by a resource with the help of a capability. A sufficient condition for the realisation of the process by the resource is the matching of the functions of the resource with the functions necessary for the process [10–12].

The necessary functions of a process can be formalised and orchestrated in a structured way through ontologies. This also allows resources to be identified for more complicated processes that require more than one capability [13]. The C4I ontology [14] provides a method for assigning capabilities to resources. In addition, it provides the possibility to distinguish between *associatedWithCapability* for a general composition of capabilities and the relationship *hasCapability* for the necessary execution of a process by a single resource.

4. Ontologies as semantic basis of AASs

For different assets to take over common functions, it is necessary that the definitions of the capabilities are semantically aligned. Therefore, the definitions should be external to the AASs, which can be achieved by references to external ontologies [3]. An ontology is a set of axioms that describe relationships between entities explicitly and in a machine-readable way. Through the logic of formalisation languages for ontologies, further relationships and thus additional knowledge can be implied from the explicitly described knowledge [15]. With a reference to entries within an ontology, properties and capabilities of assets can be semantically described. The reference to a specific entry of an ontology can be achieved within an AAS using a semantic ID in the form of a unique URL [1]. Formalised knowledge in the ontology can thus be used to establish relationships between entities defined by semantic triples. With the W3C standards for languages to formalise triples "RDF" [16] and "RDFS" [17], relationships such as "heating by gas boiler *rdfs:subClassOf* heating" can be described. The "Web Ontology Language (OWL)" [18] extends the possibilities for defining axioms with further relationship descriptions. This also makes more complicated statements possible, such as that heating and cooling are always disjoint.

5. HVAC Ontologies

In the HVAC domain, submodel elements can be semantically specified by ontologies. Existing ontologies have different thematic focuses and approaches to the division of HVAC.

The Building Topology Ontology (BOT) is an ontology for the topological description of basic components of a building [19–21]. It divides buildings into *zones*, *elements* (physical components like walls, doors or sensors) and *interfaces* (transitions between zones and elements). The classes are not further subdivided, since for more detailed descriptions of the elements, reference is made to more domain-specific ontologies.

The BRICK ontology [22] can be used to virtually map a building and its HVAC assets as well as their metadata and relationships. In studies, between 96 and 99% of all data points of existing buildings could be covered by entries of the ontology.

The DogOnt ontology [23, 24] allows the mapping of the controls of SmartHome devices with the objective of semantically connecting systems from different manufacturers. For example, on-off signals can be referenced.

The Smart Applications REFerence Ontology (SAREF) [25] can be used to describe assets within buildings. The functions for controlling a device are described, but not the basic functionalities of building services components such as heating.

With ifcOWL [26, 27], *Building Information Models (BIM)* can be represented in the form of semantic triples. A definition of basic functions of the assets within an HVAC system is not part of BIM.

With the ontologies SSN (Semantic Sensor Network Ontology) and SOSA (Sensor, Observation, Sample and Actuator) [28, 29], concepts of sensors and actuators are abstracted. Concrete HVAC capabilities are not mentioned.

Existing ontologies in building services engineering map existing structures and functions within buildings. With none of the mentioned ontologies basic HVAC functionalities can be referenced by AASs in order to take over common tasks interoperably. In the following, the development of an ontology is described that provides a structured description of supply functionalities. This can also be used to infer primary HVAC functionalities from granular asset functions.

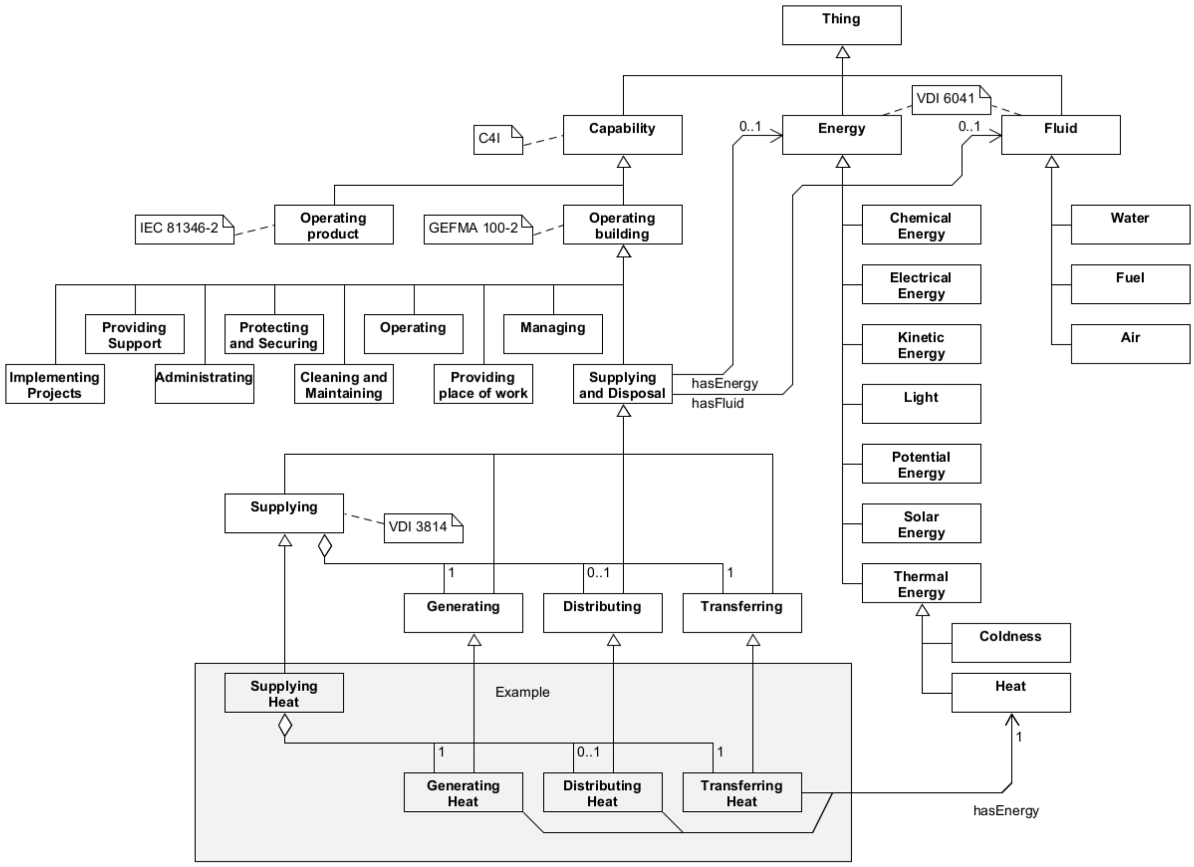

Fig. 5 - Structural design of the ontology for building technology capabilities and exemplary representation of the aggregation of heat supply

6. Derivation of fundamental HVAC functions and structure of the Ontology

For the development of the ontology, all capabilities were derived from the class *Capability* according to [14] in order to create a basis for interoperability of the ontology with other capability ontologies. The basic capability defined is *Operating_Building* and its subclasses according to [5]. Since the scope of the ontology is in building services, specifically in supplying buildings, the main focus of the ontology is on supplying and disposal of objects. Based on [7], the basic capabilities *Generating*, *Distributing* and *Transferring* were derived from this class. However, since a single asset cannot take over a complete supply task with only one of these three basic capabilities, the class *Supplying* was defined as a primary class that can be aggregated from the three fundamental functions (Figure 5). *Distributing* can be omitted in case a supply unit is both generating and transmitting in the sense of the relationship *hasCapability* according to [14], for example a stove.

Based on [6], the three fundamental capabilities were divided with an attribute into their area of activity for the supply type. For example, *Transferring* was divided into *Transferring_Energy* and *Transferring_Fluid*. For this purpose, the object properties *hasEnergy* and *hasFluid* were introduced, which assign one of the different classes of possible energies and fluids to a supplying capability.

By supplying air (*Supplying_Air*), an additional supply of heat or cold can also take place, which is why the supply of heat (*Supplying_Heat*), for example, can additionally be aggregated from the supply of air and the transfer of heat (Figure 6).

The presented ontology structure enables supply tasks within buildings to be composed modularly and according to the type of supply. The background is the possibility for supply systems in buildings to be able to reference capabilities in order to provide supply tasks and thus to be able to react flexibly to a specific need of the building.

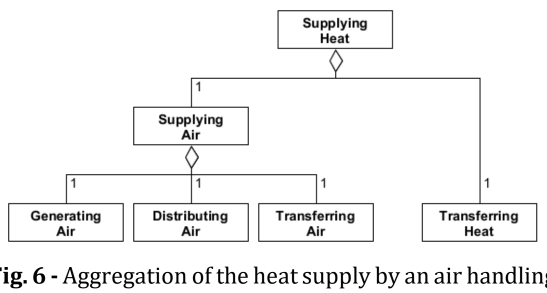

Fig. 6 - Aggregation of the heat supply by an air handling unit

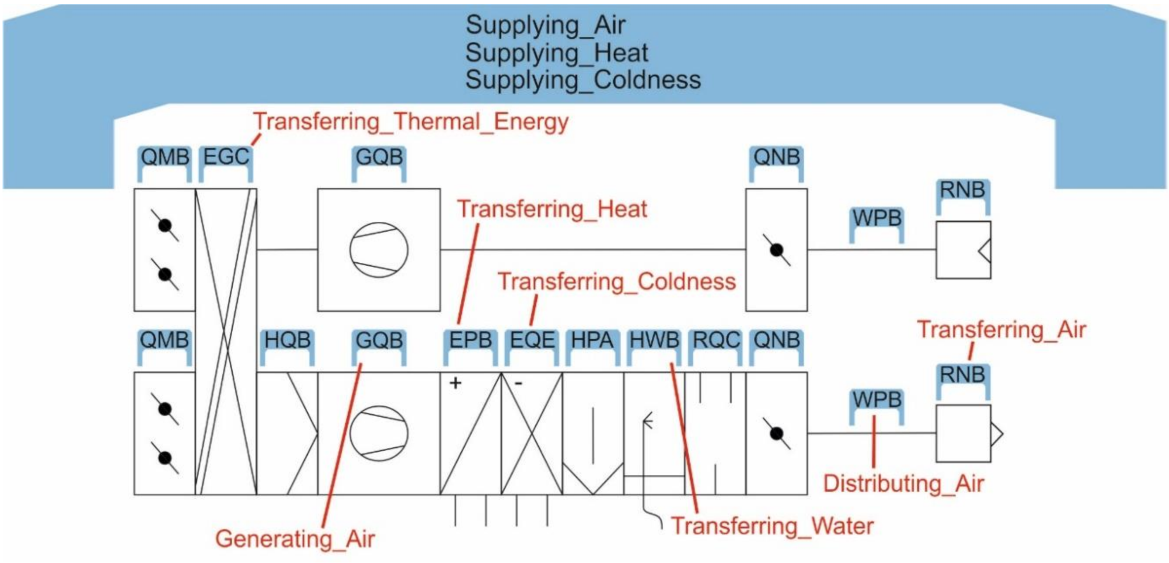

Fig. 7 - Elementary asset functions of the components of an air handling system according to [8] and their equivalent higher-level HVAC capabilities

7. Matching the supply functionalities with basic asset functions

Individual assets (e.g. fans) cannot fulfil entire supply tasks. Therefore, it must be ensured that the defined fundamental supply tasks can be implicitly inferred from various capabilities at the device level. Therefore, in addition to the ontology for building services capabilities, an ontology for asset-level capabilities has been created, which is modelled on [8]. By creating equivalences of subclasses of the basic functions Generating, Distributing and Transferring with capabilities from the ontology according to [8], both ontologies are linked so that the sum of granular asset capabilities can be used to imply higher-level HVAC functionalities. For example, a gas boiler with the capability *EM_Combustion_Heating* in a power generation system can be implied to have the higher-level HVAC functionality *Generating_Heat* without specifying this in the digital twin when developing the asset. Figure 7 shows an example of the components of a full air handling unit with all four thermodynamic air handling functions described individually at the functional level (with three-digit code). For the functions relevant to the fundamental supply functionality, the equivalent functions from the ontology are listed.

If an area within a building can be supplied with heat by more than one system, this can be determined by a capability check. A following feasibility check can be used to determine the most suitable participant, considering all asset-specific characteristics (Figure 8).

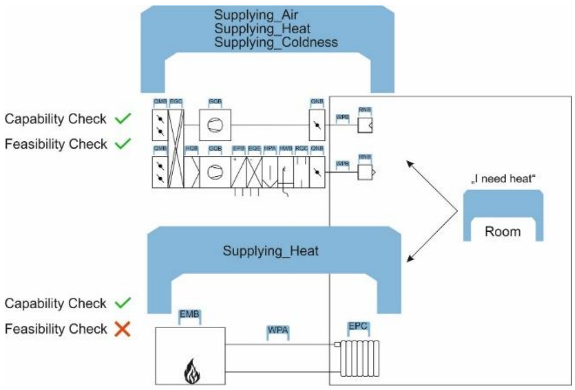

Fig. 8 - Exemplary capability and feasibility check of the heat supply of a room

8. Prototypical implementation of capability checks based on the HVAC capability ontology

In the next step, a script for querying and matching capabilities of AASs has been designed and prototypically implemented in an I4.0 environment. The developed ontologies, which can be accessed online, AASs of exemplary HVAC assets with semantic IDs to the ontologies and programmes for local hosting of the AASs are used as the basis for this.

For this purpose, both ontologies were first formalised and made available online [30]. The online access creates URLs for the respective classes of the ontologies, which can be used as semantic IDs for submodel elements of an AAS, see chapter 4.

Using [31], AASs of pumps, heat pumps and a gas boiler were designed. The AASs receive the corresponding capabilities (e.g. *Heating*) with the respective semantic IDs from the ontology for capabilities at asset level according to [8].

Subsequently, using [32], the AASs were hosted on local HTTP servers to access their data enabling the capability check. In addition, a registry was implemented in which hosted AASs are listed and which thus serves as a directory of available AASs [33].

The capability check is implemented as a script that can be executed inside or outside of AASs. For example, if a heat demand is determined via a trigger, the script is executed. The required ontology class is used as input (Figure 9, step 1). Because of the availability online, the equivalent capabilities defined in the ontology can be determined (step 2). Subsequently, AASs available in the registry are determined (step 3). In the following step capabilities of the AASs get queried and the corresponding semantic IDs are matched with the required ones (step 4). If the classes of one of the required capabilities matches an available capability, information about the corresponding AAS and the submodel elements of an AAS that are linked to the capability by means of the *CapabilityRealizedBy* attribute are retrieved. These form the starting point for a downstream feasibility check. The labelling of the digital twins of HVAC assets with their capabilities through references to ontologies available online thus forms the basis for interactions between assets through a check-up of the existing capabilities.

9. Conclusion and outlook

With the ontology of HVAC capabilities, a foundation has been laid for the assumption of common functions of different assets within buildings and self-x functionalities. By defining the capabilities of individual assets, the ontology can imply which functionality an asset has within a supply system. Buildings can be operated more efficiently and demand-oriented through energy supply systems that can be activated flexibly, because they can be controlled on the basis of current operating and environmental data. Complex engineering of the systems is not necessary. In further research work, the developments presented will be combined with concepts in the area of self-organising interactions between AASs [34]. Capability and feasibility checks are integrated as executable algorithms in active AASs. The checks provide the basis for interactions between AASs.

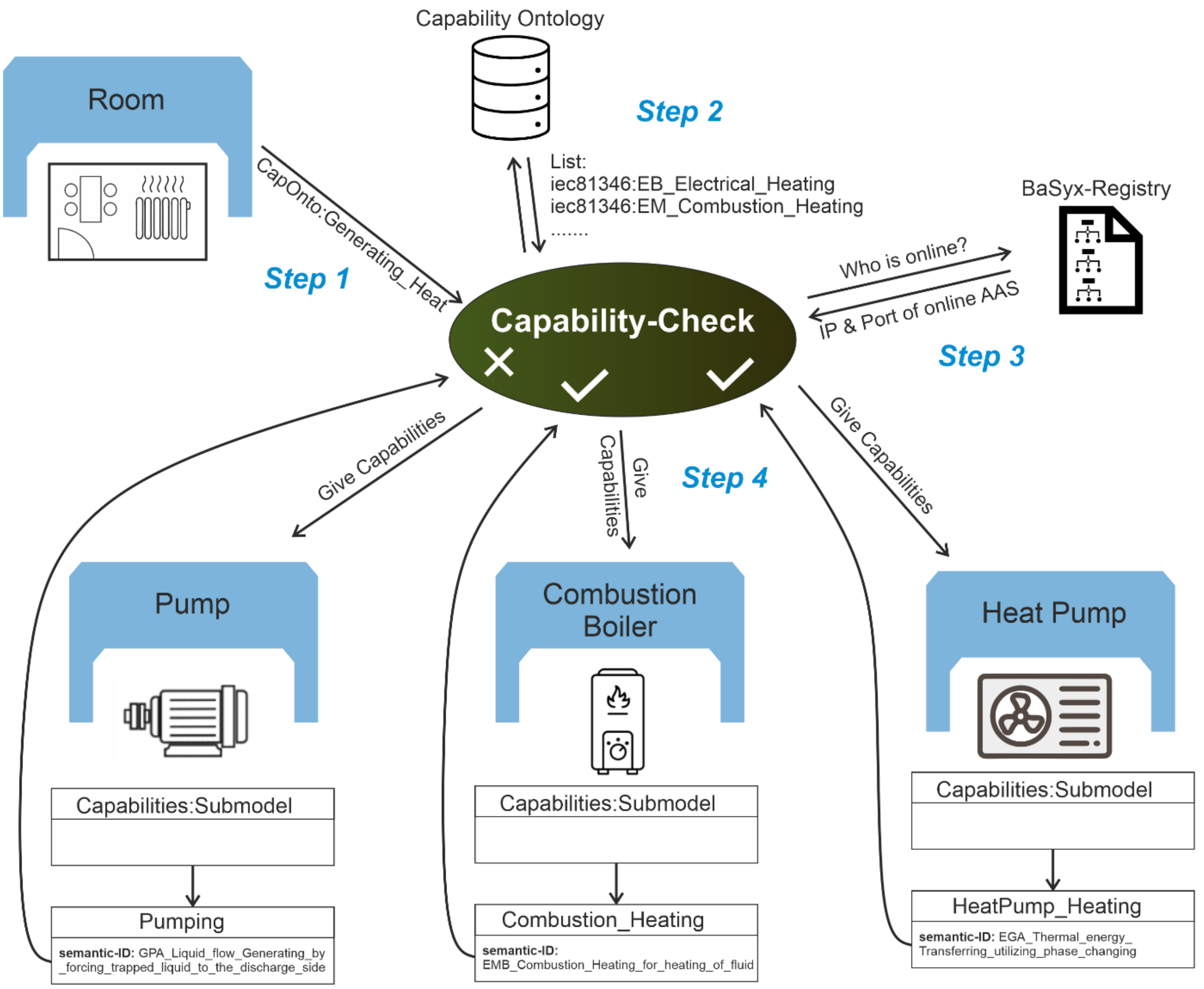

Fig. 9 - Exemplary scheme of implementing the Capability Check

The use of ontologies and the linking of different domain-specific ontologies is one way to achieve interoperability in semantically heterogeneous domains. Nevertheless, this approach involves a lot of manual effort: Ontologies must be built, linked and implemented. Results of investigations in the field of artificial intelligence (AI) show that an automated mapping of different ontologies and manufacturer- and domain-specific knowledge bases to each other is possible by means of natural language processing (NLP) [35].

10. Acknowledgement

The authors gratefully acknowledge financial support from the KSB Foundation in the project Automatic interaction of semantically heterogeneous Industrie 4.0 Asset Administration Shells by means of generic translation mechanisms based on methods of Natural Language Processing (1.1359.2020.1).

11. References

[1] Federal Ministry for Economic Affairs and Energy (BMWi). Details of the Administration Shell: Part 1 - The exchange of information between partners in the value chain of Industrie 4.0; Version 3.0. Berlin; 2020.
[2] ECLASS e.V. ECLASS: ECLASS - Standard für Stammdaten und Semantik für die Digitalisierung [cited 2020 November 17] Available from: URL: https://www.eclass.eu/index.html.
[3] Bayha A, Bock J, Boss B, Diedrich C, Malakuti S. Describing Capabilities of Industrie 4.0 Components: White Paper. Berlin; Nov 2020.
[4] McCrae JP. The Linked Open Data Cloud: Insight SFI Research Centre for Data Analytics. Available from: URL: https://lod-cloud.net/.
[5] GEFMA. GEFMA 100-2: Facility Management: Leistungsspektrum; 2004.
[6] VDI. VDI 6041, Facility-Management: Technical monitoring of buildings and building services; 2017.
[7] VDI. VDI 3814, Building automation and control systems (BACS): Part 1: Fundamentals; 2019.
[8] IEC. IEC 81346-2:2019-06, Industrial systems, installations and equipment and industrial products - Structuring principles and reference designations: Part 2: Classification of objects and codes for classes 2019.
[9] Diedrich C, Miny T, Schmitt S. D-2.11 Capability- und Feasiblity-Interaktion: BaSys 4.2; 2021 Oct 18.
[10] Backhaus J, Reinhart G. Digital description of products, processes and resources for task-oriented programming of assembly systems. J Intell Manuf 2017; 28(8): 1787–800 [https://doi.org/10.1007/s10845-015-1063-3]
[11] Pfrommer J, Stogl D, Aleksandrov K, Schubert V, Hein B. Modelling and orchestration of service-based manufacturing systems via skills. In: Modelling and orchestration of service-based manufacturing systems via skills; 2014 - 2014. IEEE.
[12] Malakuti S, Bock J, Weser M, *et al.* Challenges in Skill-based Engineering of Industrial Automation Systems *. In: Challenges in Skill-based Engineering of Industrial Automation Systems *; 2018 - 2018. IEEE; 67–74.
[13] Perzylo A, Grothoff J, Lucio L, *et al.* Capability-based semantic interoperability of manufacturing resources: A BaSys 4.0 perspective. IFAC-PapersOnLine 2019; 52(13): 1590–6 [https://doi.org/10.1016/j.ifacol.2019.11.427]
[14] Weser M, Bock J, Schmitt S, Perzylo A, Evers K. Ontology-based Metamodell for Capability Descriptions 2020.
[15] Hitzler P, Krötzsch, Markus, Rudolph, Sebastian, Sure Y. Semantic Web. Berlin, Heidelberg: Springer Berlin Heidelberg 2008.
[16] The RDF Concepts Vocabulary (RDF); 1999. Available from: URL: https://www.w3.org/1999/02/22-rdf-syntax-ns#.
[17] The RDF Schema vocabulary (RDFS); 2000. Available from: URL: https://www.w3.org/2000/01/rdf-schema#.
[18] The OWL 2 Schema vocabulary (OWL 2); 2002. Available from: URL: https://www.w3.org/2002/07/owl#.
[19] Rasmussen MH, Pauwels P, Hviid CA, Karlshøj J. Proposing a Central AEC Ontology That Allows for Domain Specific Extensions. LC3 2017: Volume I - Proceedings of th Joint Conference on Computing in Construction 2017: 237–44 [https://doi.org/10.24928/JC3-2017/0153]
[20] Schneider GF. Automated Ontology Matching in the Architecture, Engineering and Construction Domain - A Case Study. The 7th Linked Data in Architecture and Construction Workshop 2019.
[21] Rasmussen MH, Pauwels P, Lefrançois M, Schneider GF. The Building Topology Ontology (BOT); 2018 [cited 2021 April 25] Available from: URL: https://w3c-lbd-cg.github.io/bot/bot.ttl.
[22] Fierro GT. Design of an Effective Ontology and Query Processor Enabling Portable Building Applications. Technical Report No. UCB/EECS 2019; 2019-106.
[23] Bonino D, Russis L de. DogOnt: An Ontology for Intelligent Environments; 2013. Available from: URL: http://iot-ontologies.github.io/dogont/index.html#.
[24] Bonino D, Corno F. DogOnt - Ontology Modeling for Intelligent Domotic Environments. The Semantic Web - ISWC 2008 2008: 790–803.
[25] Daniele L, Hartog F den, Roes J. Created in

Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology; 225: 100–12 [https://doi.org/10.1007/978-3-319-21545-7_9]
[26] Pauwels P, Terkaj W. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. Automation in Construction 2016; 63: 100–33 [https://doi.org/10.1016/j.autcon.2015.12.003]
[27] buildingSMART International. ifcOWL; 2021 [cited 2021 April 25] Available from: URL: https://technical.buildingsmart.org/standards/ifc/ifc-formats/ifcowl/.
[28] Haller A, Janowicz K, Cox SJD, *et al.* The SOSA/SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Actuation, and Sampling. Semantic Web X 2018.
[29] Haller A, Janowicz K, Cox SJD, Le Phuoc D, Taylor K, Lefrançois M. Semantic Sensor Network Ontology; 2020 [cited 2021 April 25] Available from: URL: https://w3c.github.io/sdw/ssn/.
[30] Maisch N. CapOnto: Describing the development and usage of an ontology for I4.0-Capabilities of HVAC-Assets: TH Köln – University of Applied Sciences; 2021 [cited 2022 January 12] Available from: URL: https://github.com/thcologne-gart/CapOnto.
[31] AASX Package Explorer. Festo AG & Co. KG; PHOENIX CONTACT GmbH & Co. KG; Fraunhofer IOSB-INA Lemgo; Schneider Electric Automation GmbH; SICK AG; KEB Automation KG; Lenze SE; 2021. Available from: URL: https://github.com/admin-shell-io/aasx-package-explorer.
[32] Fraunhofer-Institut für Experimentelles Software Engineering IESE. BaSyx [cited 2021 April 25] Available from: URL: https://www.eclipse.org/basyx/.
[33] Kuhn T, Schnicke F, Ziesche C. Gut verpackt: BaSys-Tutorial, Teil 2: Verwaltungsschalen und Teilmodelle mit der BaSyx-SDK definieren. ix 2019; (12): 132–42.
[34] VDI/VDE. VDI/VDE 2193, Language for I4.0 components: Part 1: Structure of messages; 2020.
[35] Both M, Müller J. Deep Learning in Industrie 4.0 Umgebungen als Wegbereiter für automatisierte Abbildungen von Ontologien. Automation 2020 2020: 675–86 [https://doi.org/10.51202/9783181023754-675]