Automation in Construction 119 (2020) 103402

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Design of knowledge-based systems for automated deployment of building management services

Georg F. Schneider[a,b,*], Georgios D. Kontes[a,c], Haonan Qiu[a,d], Filipe J. Silva[e], Mircea Bucur[f], Jakub Malanik[g], Zdenek Schindler[g], Panos Andriopolous[h], Pablo de Agustin-Camacho[i], Ander Romero-Amorrortu[i], Gunnar Grün[a,b]

[a] *Technische Hochschule Nürnberg, Fürther Straße 250, 90429 Nürnberg, Germany*
[b] *Fraunhofer Institute for Building Physics IBP, Fürther Straße 250, 90429 Nürnberg, Germany*
[c] *Fraunhofer Institute for Integrated Circuits IIS, Nordostpark 84, 90411 Nürnberg, Germany*
[d] *Ulm University, Institute of Artificial Intelligence, 89069 Ulm, Germany*
[e] *Instituto de Soldadura e Qualidade, 33, 2740-120 Porto Salvo, Portugal*
[f] *Kiwi Power, 3 London Wall, London EC2M 5PD, United Kingdom*
[g] *Honeywell Prague Laboratory, Pod vodárenskou věží 4, 182 08 Prague 8, Czech Republic*
[h] *Hypertech, Athens, Greece and Que Technologies, Athens, Greece*
[i] *TECNALIA, Basque Research and Technology Alliance (BRTA), Parque Científico y Tecnológico de Bizkaia, Edificio 700, 48160 Derio, Spain*

ARTICLE INFO

Keywords:
Building management services
Knowledge-based systems
Energy efficiency
Knowledge engineering
Ontology

ABSTRACT

Despite its high potential, the building's sector lags behind in reducing its energy demand. Tremendous savings can be achieved by deploying building management services during operation, however, the manual deployment of these services needs to be undertaken by experts and it is a tedious, time and cost consuming task. It requires detailed expert knowledge to match the diverse requirements of services with the present constellation of envelope, equipment and automation system in a target building. To enable the widespread deployment of these services, this knowledge-intensive task needs to be automated. Knowledge-based methods solve this task, however, their widespread adoption is hampered and solutions proposed in the past do not stick to basic principles of state of the art knowledge engineering methods. To fill this gap we present a novel methodological approach for the design of knowledge-based systems for the automated deployment of building management services. The approach covers the essential steps and best practices: (1) representation of terminological knowledge of a building and its systems based on well-established knowledge engineering methods; (2) representation and capturing of assertional knowledge on a real building portfolio based on open standards; and (3) use of the acquired knowledge for the automated deployment of building management services to increase the energy efficiency of buildings during operation. We validate the methodological approach by deploying it in a real-world large-scale European pilot on a diverse portfolio of buildings and a novel set of building management services. In addition, a novel ontology, which reuses and extends existing ontologies is presented.

1. Introduction

The reduction of energy-related emissions of greenhouse gases is an ultimate goal for societies in industrialised and emerging countries in order to mitigate their negative effects on global climate [1]. The buildings sector comprises a large share of the primary energy demand in industrialised and emerging countries (e.g. 43% in Germany [2]) and energy efficiency measures applied in this domain can significantly contribute to reduce the total energy demand. Buildings with highest rating in terms of energy efficiency will be mandatory in future in many regions of the world, e.g. all new buildings in the European Union need to consume nearly zero energy by 2021 [3].

1.1. Impact of building management services on energy performance of buildings

In buildings a reduction of 14–50% of thermal energy and 9–13% of electrical energy, dependent on the building type, can be achieved by deploying a sophisticated building automation system [4]. However, the whole building needs to be properly serviced as up to 20% of final

[*] Corresponding author at: Technische Hochschule Nürnberg, Fürther Straße 250, 90429 Nürnberg, Germany.
E-mail address: georg.schneider@ibp.fraunhofer.de (G.F. Schneider).

https://doi.org/10.1016/j.autcon.2020.103402
Received 7 August 2019; Received in revised form 24 August 2020; Accepted 27 August 2020
0926-5805/ © 2020 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license
(http://creativecommons.org/licenses/BY/4.0/).

energy in a building is lost in the heating system due to poor control configuration and system faults [5]. In particular, a plethora of well-studied Building Management Services (BMServ) exist such as Fault Detection and Diagnosis (FDD) [6,7] methods to improve building operation. Throughout this work we distinguish a certain method of BMServ, such as the APAR rule set [8] and the instance of such a method, which actually is deployed in some real building. Although it is proven that these methods help to significantly improve the energy efficiency of a building, still, the sector fails to achieve the anticipated goals [9].

1.2. Problems and challenges in the setup of building management services

The process of setting up BMServ in a building typically requires trained experts, which are knowledgeable in domains such as building physics, Heating Ventilation and Air-Conditioning (HVAC) engineering and building automation. These experts gather detailed knowledge for each targeted building including, among others, a description of the building envelope, the topology of the building including the decomposition in its storeys and spaces. In addition, detailed knowledge on installed technical equipment (boilers, chillers), knowledge on the distribution network of pipes, ducts and wires guiding material and energy flows through the building. Additionally, the relationship between spaces served by some technical equipment needs to be known. In addition, the expert needs to know what quantities and units are measured by points in the Building Automation Systems (BAS) and how they are affiliated to spaces or a piece of technical equipment. Moreover, the actual control algorithms are relevant [10] as these significantly influence the overall building performance. Finally, the operational history of a building including time series readings of sensor and actuators as recorded in the Building Management System (BMS) need to be available. Typically this knowledge needs to be collected from disparate sources and extracted from heterogeneous formats such as textual descriptions, spreadsheets, databases, human experts and drawings [9,11].

The process of manually collecting the pieces of knowledge needed, manual selection and manual setup of a respective BMServ method in a building is a cumbersome process, which requires a noticeable amount of human experts. The knowledge needs to be collected from disparate sources and needs to be decoded from heterogeneous formats [11,12]. In addition, each of the million existing buildings [13] slightly differs in its constellation of designed envelope, technical equipment and control system. These constellations of technical installations need to be matched to the diverse needs of a plethora of existing BMServ methods [14]. As building experts are facing this combinatorial explosion the low deployment rate of BMServ is no surprise [15]. Consequently, a finding in the review by Shi & O'Brien [16] on Automated Fault Detection and Diagnosis (AFDD) methods, which can be seen as a subdomain of BMServ, is that 'the main cost barrier for [a] AFDD system is the human resource required to setup and maintain a proper AFDD system'.

1.3. Need for formal Knowledge Base and limitations of existing approaches

To address the problems and challenges mentioned above there is a strong need to automate the knowledge intensive tasks related to the setup of BMServ. For this purpose the respective knowledge needs to be collected from disparate silos and transformed from heterogeneous formats. The flexible modelling capabilities of Semantic Web Technologies (SWT) have proven to be appropriate to cope with these challenging tasks and allow to successfully integrate respective building knowledge [11,17,18] into a building knowledge base. In addition, their formal basis in description logic [19] allows for semantic search and reasoning. Moreover, only a rigid formal description of both the present constellation of envelope, technical equipment and automation system in a building can be matched with the diverse Knowledge Requirements (KR) of BMServ [20,21]. Here, KR refer to the various knowledge needed to decide whether a certain BMServ method can be configured, how many instances of the method can be deployed and if the required inputs are correctly available. For instance, subtle differences, such as storing the floating point number of a temperature reading in degree Celsius instead of degree Fahrenheit, can cause the failure of a BMServ method or prevent its use. In traditional data base technologies this often can only be discovered by a human expert reading the data base schema and, hence, this is not suitable for the intended automated deployment of BMServ using knowledge-based systems. A consolidated building knowledge base allows the design of a knowledge-based system, which automates the manual process described above [14,20].

A number of existing approaches address some of these problems by following a knowledge-based approach (see review in Section 2). This involves the design of a knowledge-based system based on ontology to solve the mentioned problems. To design these systems it is necessary to develop a formal description of relevant domain knowledge, acquire assertional knowledge of a specific site and, based on the acquired knowledge, design a service to automatically deploy BMServ [15,20–22]. This involves: (i) Matching KR of the selected BMServ method with the formal specification of the building constellation, (ii) if a match is found configure an archetype of the matched BMServ method for the respective building and deploy it in the building, e.g. on a daily basis (dependent on the BMServ method). In this work we call systems capable of performing all of these task as Automatically enabled Building Management Services (ABMS).

Existing contributions describing ABMS are analysed as presented in Section 2. Most contributions on ABMS use ontology-based knowledge representation. A key motivation in this regard is the ability to successfully cope with heterogeneous data formats and separated knowledge silos prevalent in the building domain [9,11,23]. Some limitations in the existing contributions can be identified. Frequently, the presented approaches lack the use of a qualified Knowledge Engineering Method (KEM). For example, from 151 papers reviewed by Zhou et al. [24] in their review on ontology-related works in the construction industry only 9 mention the use of a qualified KEM. Here, we define a qualified KEM as a KEM, which covers the ontology development life cycle defined by Pinto & Martins [25] (specification, conceptualisation, formalisation, implementation, maintenance, knowledge acquisition, evaluation and documentation) and includes reuse of existing ontologies. One drawback of designing ontologies without a qualified KEM is that it is difficult for externals to understand why some design decisions have been chosen. In addition, a pitfall in this regard is the absence of reuse of existing ontologies as stipulated by state of the art KEMs [25,26], e.g. see reviewed works in Section 2 and criticism in Rasmussen et al. [27]. Rather, researchers (re-)design ontologies from scratch potentially recreating interoperability issues [27] among the different approaches just on a semantically higher level [28]. Furthermore, the acquisition of assertional knowledge on sites and buildings of interest to populate the project knowledge base with instances is often undertaken in an ad hoc manner, without using existing standards in the domain. This makes it difficult to reuse or port the developed methods and tools to different sites and buildings and prevent the wide spread adoption of these methods and, hence, improve the energy performance of the domain. Finally, existing ontology-related studies on BMServ often do not exploit the semantically well-defined collection of knowledge on a site or building to automatically deploy BMServ. This contradicts demands made, for instance, to automatically deploy AFDD [16]. Instead the approaches keep humans in the loop and visualise obtained insights to building managers.

To overcome the described limitations of existing approaches a consistent methodological approach is needed for the design of ABMS. In this regard we define the following requirements for a novel design methodology:

R1. Use of a qualified KEM, which covers the life-cycle stages in ontology engineering and reuse of existing ontologies as defined by Pinto & Martins [25]. This ensures that relevant terminological domain knowledge is captured and the developed solutions are portable as existing ontologies are reused

R2. Use of a structured (automated) knowledge acquisition method to capture assertional knowledge on a respective building portfolio based on existing, open standards;

R3. Use of collected knowledge to automatically deploy instances of BMServ methods.

1.4. Main contributions of the article and outline

The analysis of existing work presented in Section 2 reveals that none of the existing approaches fulfils the defined requirements. The identified gap is the absence of a comprehensive methodological approach for the design of knowledge-based systems for the automated deployment of BMServ. Hence, this work aims at filling this gap by presenting a novel methodological approach, the ABMS methodology, for the design of such systems. The main purpose of the methodological approach is to guide through the different steps of the design process and their iterations, provide best practices and methods needed to accomplish each step according to the current state-of-the-art. By this the methods supports the development of this kind of systems and provides the basis for a wide spread adoption of the technology in the domain. Moreover, stipulation of reuse by the approach enables the portability of developed solutions between different buildings.

The methodological approach includes three steps: In the first step terminological knowledge of a domain is captured using a qualified KEM. The second step comprises the acquisition of assertional knowledge of an existing building portfolio from open standards. Finally, in the third step, the acquired, semantically well-defined knowledge is used for the automated deployment of BMServ.

We validate the approach by testing its capabilities in full filling the defined requirements by designing a knowledge-based system, which automates the deployment of a novel set of simulation-assisted BMServ developed in the Modelling Optimization of Energy Efficiency in Buildings for Urban Sustainability (MOEEBIUS) project[1] [29]. One outcome is a novel ontology, the MOEEBIUS ontology, which reuses and extends the BRICK ontology [30] and Ontology of Units and Measures (OM) [31] for representing building knowledge. To extract assertional knowledge we follow an approach using the Construction Operations Building information exchange (COBie) format [32] and show how it can be mapped to an existing ontology. Finally, we present results from deploying the designed knowledge-based system for the automated deployment of novel BMServ constituting the MOEEBIUS holistic energy performance framework. In addition, we report lessons learned from deploying the method on a diverse building portfolio available from a large-scale European pilot distributed across three countries composed of in total 47 buildings.

In the remainder of this work we analyse in Section 2 existing works related to knowledge-based systems for the automated deployment of BMServ. Then we present in Section 3 a novel methodological approach for the design of these systems. Finally, we validate the approach in Section 4 by deploying it in a real-world large-scale use case related to the MOEEBIUS project and present results from a designed knowledge-based system, which automatically deploys a diverse set of BMServ constituting the MOEEBIUS framework.

2. Related work

Within this section we review existing contributions, which use knowledge-based methods based on ontology to increase the energy efficiency in buildings w.r.t the initially defined requirements. An overview of the results of the analysis is presented in Table 1. We select relevant contributions on ontology-related research based on the recent reviews by Zhou et al. [24] and Zhong et al. [33]. A large amount of research is published related to AFDD, which can be understood as a special domain of BMServ. Here we consider only ontology-related references from the two recent reviews in this domain published by Kim et al. [34] and Shi & O'Brien [16].

The review of KEMs is out of scope of this work (see reviews in [25,26]). As mentioned in Section 1, we define a qualified KEM for the scope of this work as a KEM, which covers the ontology development life cycle originally defined by Pinto & Martins [25]. A general introduction to description logic and ontological modelling using SWT can be found in Petnga & Austin [19] aðnd an introduction to semantic web technologies with specific emphasis on the built environment is provided by Pauwels et al. [35].

2.1. Ontology-based energy management systems

D'Elia et al. [36] describe an approach for context-aware applications, which support the maintenance of large buildings. The developed ontologies allow to describe faults, failures and fault detection as well as sensor-provided monitoring data and the emphasis of the approach lies in the semantic integration of multi-vendor and multi-device scenarios. The developed ontologies reuse through extension the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) upper ontology [45] and describe the automated deployment of fault detection algorithms to improve building performance. However, no use of KEMs is reported and the 'automated' refers to the periodic execution of manually configured algorithms.

Wicaksono et al. [37] present SERUM-iB, an IT framework for intelligent energy management in buildings based on ontology and BAS. The authors report on the development of a building domain ontology manually created by experts without considering a qualified KEM. Site specific knowledge is acquired semi-automatically from two-dimensional drawings [46] and from data mining algorithms. A module for energy analysis allows the detection of energy waste situations and reports them to the user through a presentation module. Rules identifying the energy waste situations are partly generic and can, thus, be instantiated for every building in the knowledge-base as well as added semi-automatically for specific building instances.

Bat-MP [38] is an ontology-based energy management platform, which supports semantic integration of heterogeneous BAS protocols to enable a scaleable deployment of agents and applications to enhance energy consumption of buildings. The ontologies are developed from scratch without dedicated methods and no structured acquisition of site knowledge is reported. Services to improve building operation are automatically deployed from knowledge stored in a knowledge base.

The case for the ontology-based performance assessment of buildings to close the perceived gap between predicted and actual energy demand is described by Corry et al. [9]. Based on the acquisition of site and domain knowledge from heterogeneous knowledge silos, the developed energy performance ontology and framework allows to assess the energy performance of buildings. Site knowledge automatically acquired by linking the Web Ontology Language (OWL) serialisation of the Industry Foundation Classes (IFC) model [47] to the ontology. The use of a qualified KEM during the development process of the energy performance ontology is not reported.

Tomašević et al. [11] present an ontology-based facility management framework to enable ISO 50001 [48] compliant energy management. The authors report the use of the Ontology 101 method [49] during development of ontologies needed for the integration of

[1] https://cordis.europa.eu/project/id/680517, Last accessed: 24 August 2020

Table 1
Overview of reviewed existing contributions with respect to the initially defined requirements.

Citation	Brief description	Use of qualified KEM	Structured knowledge acquisition	Use of ABMS
D'Elia et al. [36]	Approach for context-aware applications, which support the maintenance of large buildings.	–	°	–
Wicaksono et al. [37]	Describe SERUM-iB, an IT framework for intelligent energy management in buildings based on ontology and BAS.	–	°	°
Caffarel et al. [38]	Present an ontology-based energy management platform Bat-MP, which supports semantic integration of heterogeneous BAS protocols.	–	–	✓
Corry et al. [9]	Approach for ontology-based performance assessment of buildings to close the perceived gap between predicted and actual energy demand.	–	✓	°
Tomašević et al. [11]	ISO 50001 compliant ontology-based facility management framework.	✓	°	–
Han et al. [39]	Ontology-based energy management system.	–	–	°
Esnaola-Gonzalez et al. [23]	Describe Energy Efficiency Prediction Semantic Assistant process.	–	–	–
Kučera & Pitner [40]	Semantic Building Management System	–	–	°
Dibowski et al. [14,21,41–43]	Set of ABMS methods for virtual sensors, fault propagation, etc.	–	✓	✓
Schneider et al. [10,20,22]	Example of ABMS for FDD	✓	°	✓
Balaji et al. [30,44]	BRICK ontology and ABMS on, e.g., model predictive control, demand response and occupancy modelling.	–	°	✓
This work	Use qualified KEM; reuse and extend existing ontologies [30,31]; acquire knowledge based on COBie format; automated setup of diverse BMServ.	✓	✓	✓

BAS: Building Automation Systems. ✓: Requirement fulfilled, °: requirement partly fulfilled, –: Requirement not fulfilled.

heterogeneous data sources in energy management. The developed ontology is based on existing upper ontologies. As multiple data sources are needed for populating their knowledge-base a number of structured and unstructured inputs are used such as interviews with facility personnel or technical data sheets. The use of BMServ such as FDD [6,7] is reported, however, their automated deployment is not described.

Another report of an ontology-based energy management system is presented by Han et al. [39]. Despite the high quality documentation of the developed ontologies, no use of a KEM is reported and the intelligent energy management system based on semantic rules is instantiated semi-automatically by a human expert. The acquisition of site knowledge is carried out semi-automatically.

Esnaola-Gonzalez et al. [23] present the Energy Efficiency Prediction Semantic Assistant (EEPSA) process, which supports balancing the trade off between occupant comfort and energy demand in tertiary buildings. The approach supports human building operators in discovering knowledge from different data bases. The developed ontology is based on a set of domain ontologies and the use of a qualified KEM is not reported. For the investigated use case the respective knowledge is instantiated manually. The presented approach supports end users with enriched knowledge on their building but does use the collected knowledge to automatically setup a ABMS.

The Semantic Building Management System (SBMS) described by Kučera & Pitner [40] integrates BMS and Computer Aided Facility Management (CAFM) knowledge using ontology. It constitutes a semantic middleware layer to enable diverse applications to improve building performance and efficiency. The ontology is developed based on 'long term experience' [40] of the authors and the (semi-)automatic population of the knowledge repository is kept for future research. The presented middleware allows the retrieval of static building and operational performance knowledge to enable ABMS but except a walk through describing the approach an ABMS is not described.

2.2. Ontology-based automated building management services

In recent years a number of researchers have focused on ontology-based approaches to realise ABMS.

Dibowski et al. describe a number of applications including automated setup of FDD algorithms [14,21], automated setup of virtual sensors [41], probabilistic model-based fault detection [42] and automated fault propagation in BAS [43]. The variants of the approaches acquire structured site knowledge from accepted formats in the domain and reuse an existing ontology, BASont [50], to represent respective domain and site knowledge. The presented applications represent excellent examples of ABMS, however, in the documentation of BASont no qualified KEM is reported.

Schneider et al. present precursors of this work on ontology-based ABMS for automated deployment of rule-based fault detection in BMS [20,22] and related to verifying designed control logic in BAS [10]. The described methods for ABMS use the Ontology 101 method [49] for the structured design of the utilised ontologies. However, the structured acquisition of site knowledge from domain formats is undertaken manually or semi-automatically in the reported approaches.

To enable the effortless porting of applications implementing technical BMServ between different buildings Balaji et al. present the BRICK [30,44] ontology. The ontology is engineered following a 'ground truth' [44] approach, where the vocabulary and concepts are mined from real world'BMS deployments and smart building applications' [44]. Site knowledge is acquired from disparate sources such as human readable BAS point names and multiple successful examples of ABMS are reported. This includes model predictive control, demand response and occupancy modelling, which are tested on a portfolio of six buildings.

2.3. Summary

From the review presented above, none of the existing approaches

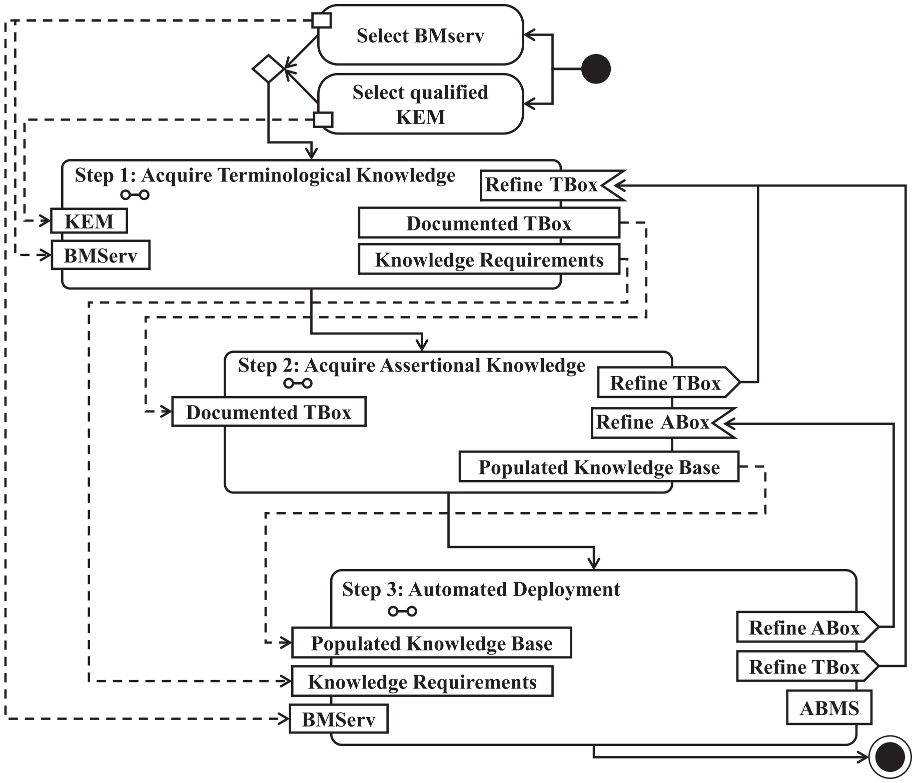


Fig. 1. UML activity diagram providing an overview of the activities of the novel methodological approach for the design of knowledge-based systems for the automated deployment of Building Management Services (BMServ). Initially a BMServ method and a qualified Knowledge Engineering Method (KEM) need to be selected. The three main steps (Step 1–3) are subsequently executed, while iterations among them are possible if refinements of TBox Terminological Knowledge (TBox) and Assertional Knowledge (ABox) are necessary. The compound activities depicting Step 1 to 3 are detailed in Figs. 2 to 4, respectively. For the sake of clarity, object flow between pins (rectangles) is depicted as dashed arrow in this figure.

fulfils the above defined requirements (see Table 1).

One main shortcoming is the lack of a consistent use of a qualified KEM. Only four contributions are found, which make some use of a qualified KEM [10,11,20,22]. This finding is congruent with the finding presented in the review by Zhou et al. [24], where from 151 papers only 9 mention the use of a qualified KEM. The contributions by Dibowski et al. [14,21,41–43] constitute excellent examples on how to leverage on the possibilities offered by ontology-based ABMS, e.g. formal semantics and reasoning to improve building performance. However, all contributions are based on BASont [50], which has not been designed using a qualified KEM. Furthermore, the structured acquisition of assertional knowledge on sites and buildings of interest to populate the project knowledge base is often undertaken in an ad hoc manner often without the use of existing standards in the domain [23,36–40] This makes it difficult to reuse the developed ABMS methods and port them to different sites and buildings. Another drawback of existing contributions is that the approaches do not exploit the context-rich and semantically well-defined site and building knowledge to automatically deploy BMServ [9,11,23,36,37,39,40]. The reviewed approaches mainly use semantic technologies to integrate and aggregate heterogeneous building knowledge but, rather keep the human in the loop, e.g. to inform or visualise insights to building operators and owners.

The reviewed contributions provide evidence, that ontology-based knowledge representation can successfully integrate heterogeneous knowledge from various sources in the building domain and that automating BMServ is a possibility to support building operation to reduce, e.g., energy consumption. However, exploiting the full potential of the technology, i.e. ‘automate the automated’ by using the formally defined knowledge in a building knowledge base, match this with the KR of BMServ to automatically deploy a variety of the available BMServ is rather limited. In the absence of a comprehensive method guiding the design, existing contributions constitute isolated solutions, which are designed from scratch, tailored to specific sites and buildings and cannot be easily ported. This hampers the widespread adoption of the technology. Hence, to fill this gap we introduce in the next section the ABMS method, a methodological approach for the design of knowledge-based systems for the automated deployment of BMServ.

3. Methodological approach

To overcome the identified problems and shortcomings of existing approaches as analysed in Section 2, we present in this section a novel methodological approach for the design of knowledge-based systems for the automated deployment of BMServ, the ABMS methodology. In Section 3.1, we provide an overview of the approach and how iterations between the main steps happen. We describe the preparatory activities of selecting a BMServ method intended to be automated and the selection of a qualified KEM in Section 3.1. In addition, we provide insights on how the novel methodological approach can be executed in parallel, if multiple methods of BMServ are to be automated. The three main steps are described in detail including examples in Sections 3.2 to 3.4. The different steps and activities of the approach are depicted as Unified Modelling Language (UML) [51] activity diagrams and the utilised graphical nomenclature is presented in Fig. A.1 in the appendix. We include a description of the involved human roles of each step. Beyond the descriptions and examples provided in this section the deployment and validation of the approach in a real-world large-scale use case is treated in depth in Section 4 including results from designing a knowledge-based system for the automated deployment of a diverse set of BMServ.

3.1. Overview

The main purpose of the methodological approach is to guide through the different steps of the design process and provide best practices and methods needed to accomplish each step according to the current state-of-the-art. The two initial activities of the approach, the three main activities and potential iteration loops among the main activities are depicted in the UML [51] activity diagram presented in Fig. 1.

A necessary input to the presented methodological approach is the selection of a BMServ method, which is intended to be automated. If

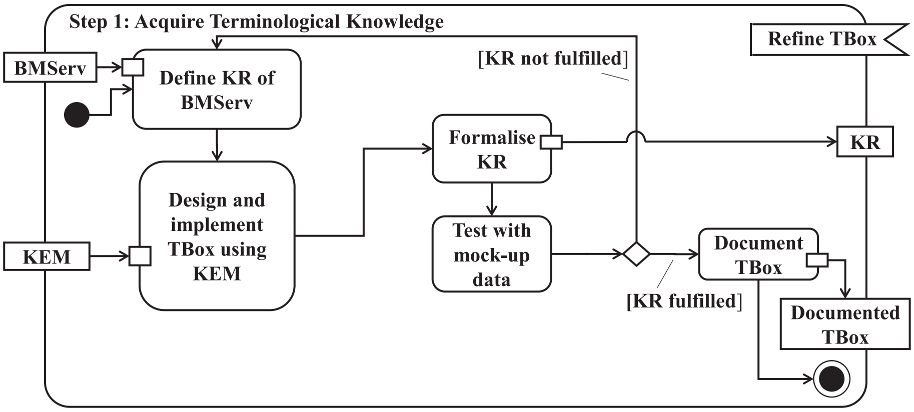


Fig. 2. UML activity diagram depicting the activities related to the acquisition of terminological knowledge in Step 1. In an iterative manner, the Knowledge Requirements (KR) of the pre-selected Building Management Services (BMServ) method are defined and the respective Terminological Knowledge (TBox) is designed and implemented using the selected qualified Knowledge Engineering Method (KEM). Dependent on the selected implementation language the KR are formalised and tested against mock-up data. If the KR are fulfilled the TBox model is documented as final activity.

required, it is possible to select in this activity multiple BMServ methods. BMServ methods with similar KR can be handled at once, and only a minor overhead can be expected. If the KR of the BMServ methods of interest are diverse we do not recommend to consider them at once, as, for instance, several diverse knowledge domains need to be integrated into one Terminological Knowledge (TBox) model. A TBox model, which covers multiple domains can be cumbersome to maintain and a can cause a significant overhead, when developing the intended knowledge-based system. Here, again the importance of reusing existing domain ontologies (cf. R1), e.g. BRICK [30], Semantic Sensor Network Ontology (SOSA) [52] or Building Topology Ontology (BOT) [53], comes into play, as this ensures that obtained separated solutions still remain compatible apart from domain specific extensions. In the following, explanations are provided in singular but apply to multiple methods as well.

Another prerequisite is the selection of a qualified KEM (cf. definition in Section 1.3). As analysed in Section 2 a number of approaches exist, which realise ABMS. However, in absence of a qualified KEM these approaches frequently violate the basic principle of ontology reuse in ontology engineering. This results, among others, in the redefinition of terms, which then prevent the reuse of the developed solutions or the combination of solutions with other solutions. Hence, the selection of a qualified KEM is an essential prerequisite and a preparatory activity in the approach presented in this paper.

The development of KEMs for the design of ontologies has been researched in the field of computer science for decades. A plethora of methods exist, such as TOVE [54], METHONTOLOGY [55], Ontology 101 [49], On-to-knowledge methodology [56], NeOn [57], UPON Lite [58], with partly overlapping capabilities. A complete review of all existing methods is out of the scope of this work. Instead we refer to the comprehensive reviews of Corcho et al. [26] and Pinto & Martins [25], which define minimal requirements for KEMs, describe needed activities and steps and review, compare and assess different methods available in literature. A suitable, qualified KEM can be selected based on the defined requirements [25,26].

After selecting a BMServ method and a qualified KEM the workflow can continue with the three main activities. The activities are depicted sequentially but iterations among them can occur. In Step 1 terminological knowledge is acquired for the selected BMServ method and using the selected KEM. As outputs a documented TBox model and formally defined KR of the selected BMServ method are obtained. The documented TBox model serves as an input to Step 2, which is dedicated to the acquisition of assertional knowledge distributed across various knowledge sources of a given portfolio of buildings. A result of the second step is a populated knowledge base, which holds both the terminological and assertional knowledge needed for the automated deployment. Finally, in the third step the populated knowledge base, the defined knowledge requirements and the selected BMServ method serve as inputs to create the intended ABMS. At the end of each step the obtained results are checked and reiterations of previous steps are triggered if required (e.g. trigger'Refine Tbox' in Step 2). The method terminates when the ABMS finally is successfully created. The three main steps are described in detail in the subsequent sections.

Over the lifetime of a building potentially new devices or technical equipment are installed in the building. If these can be described with the existing TBox model, the Assertional Knowledge (ABox) model needs to be updated (Step 2, see Section 3.3), else, the TBox model needs to be revised according to the new requirements (Step 1, see Section 3.2). In case of a new BMServ method the KR of the method need to defined and checked whether they are congruent with the existing ones. If they match no changes are necessary and a service can be implemented (Step 3). If KR differ a reiteration of the method needs to be triggered to adjust the TBox model (Step 1) until the KR can be fulfilled.

3.2. Step 1: Acquire terminological knowledge

The *first step* covers the activities needed to acquire and formally represent terminological knowledge of the domain of interest. The activities associated to this step are depicted in Fig. 2.

For the selected BMServ method a collection of KR is defined. These KR define, which knowledge is needed as an input to create an instance of the selected BMServ method, determine if it is applicable in the given constellation of building envelope, technical equipment and BAS (see Section 1.2) and deploy it in the target building. The KR can be formulated semi-formally as Competency Question (CQ)s [54] and more formally if an implementation language is chosen. Examples for formal definitions are specifications using SPARQL Protocol and RDF Query Language (SPARQL) queries ([20], Section 4.5) or specifications in OWL [21].

The initially selected qualified KEM is used to design a TBox model. As mentioned above many candidate methods exist, which can be selected based on the presented requirements in the reviews presented by Corcho et al. [26] and Pinto & Martins [25]. Here, we describe the key activities associated with this task (sequentially) and include building domain specific examples [25]:

- **Specification:** The main goal of this activity is to clarify the purpose of the intended ontology. Main questions to be answered are, which users will be actively using the ontology, which domains are in its scope and which not, etc. The output of this phase can be a human readable document, which contains, for instance, a glossary of terms [55], which contains basic terms and verbs used in the respective domain, e.g. Sensor isLocatedIn Room;
- **Conceptualisation:** In this activity contextual relationships among the found concepts are defined and can be documented, for instance, using concept dictionaries. The *reuse* of existing conceptualisations is an integral part of KEMs [25] and can be considered as a sub-

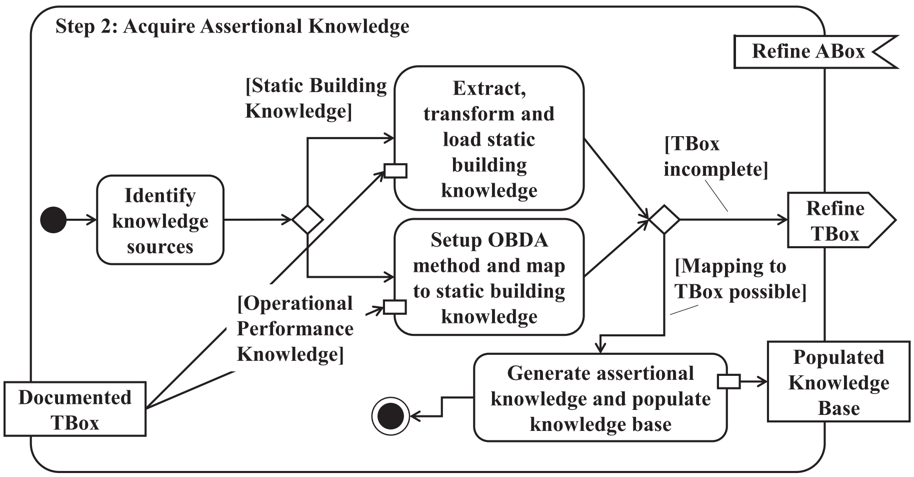

Step 2: Acquire Assertional Knowledge
Refine ABox
[Static Building Knowledge]
Extract, transform and load static building knowledge
Identify knowledge sources
[TBox incomplete]
Refine TBox
Setup OBDA method and map to static building knowledge
[Mapping to TBox possible]
[Operational Performance Knowledge]
Documented TBox
Generate assertional knowledge and populate knowledge base
Populated Knowledge Base

Fig. 3. UML activity diagram depicting the activities related to Step 2. The goal of this step is to populate the Terminological Knowledge (TBox) model with Assertional Knowledge (ABox). Static building knowledge such as the building topology is materialised completely and loaded to the knowledge base. Operational performance knowledge such as sensor readings is materialised on demand through Ontology-Based Data Access (OBDA) [61] and mapped to the static building knowledge. A refinement of the TBox can be triggered if the TBox is found to be incomplete. A populated knowledge base is created as an output of the activity.

activity of conceptualising. An exemplary conceptualisation frequently occurring in the building domain (cf. [53]) is the definition of the concepts Site, Building and Storey and defining the relationships hasBuilding and hasStorey between Site and Building and Building and Storey, respectively.;

- **Formalisation:** In this intermediate activity the informal/ semi-formal model obtained up to now is further formally specified by adding axioms such as inheritance relationships, etc. An example for this can be to specialise the concepts Temperature_Sensor and Humidity_Sensor from the general concept Sensor;
- **Implementation:** Now the formalised model is implemented in the knowledge representation language of choice. A number of knowledge representation languages exist and should be chosen dependent on the required expressivity [26]. It should be noted that current approaches in the building domain mainly focus on OWL [59] for implementation. For instance, the triple ex:Sensor rdf:type owl:Class formally defines the concept Sensor as a OWL class.;
- **Maintenance:** Knowledge engineering is an iterative process as KR may change over the course of a project or new knowledge is intended to be modelled. Hence, the developed terminological knowledge model needs to be revised and updated. In this activity, this is ensured by evaluating the developed model according to the defined KR prior to ending this step.

In the development process some activities are executed in parallel with the activities above [25]:

- **Knowledge Acquisition:** At schema level different knowledge sources can be tapped to acquire respective domain knowledge, e.g. through expert interviews with building professionals, literature review e.g. ([35]) or brainstorming.;
- **Evaluation:** The developed ontology needs to be constantly evaluated as its sole purpose is to fulfil the initially defined KR;
- **Documentation:** During each of the above described activities the respective development status of the model needs to be documented. This is particularly important when iterating and during revisions to clarify on earlier chosen modelling decisions, e.g. using the Widoco tool mentioned below.

After designing and implementing the TBox model the defined KR are formalised according to the chosen implementation language, e.g. as SPARQL queries (cf. Section 4.5 and [20]). The formalised KR are handed over to other activities later in the approach as well as used to test the resulting TBox model with mock-up data whether the initially defined KR can be fulfilled. A reiteration of Step 1 is triggered if the KR could not be fulfilled. If the KR are fulfilled the TBox model is documented. Again, if OWL is chosen as implementation language, the documentation can be undertaken with existing tools such as Widoco [60]. As a result and terminating the step a formally defined TBox model with a human readable documentation is created, which allows to track the design decisions and evolution of the model.

In the activities associated to Step 1 one or more knowledge engineers are involved, which are proficient in knowledge engineering and implementing formal domain models. In addition, one or more domain experts from the BMServ domain support this activity, which are guided by the knowledge engineer through the knowledge engineering process and are one important knowledge source in the elicitation of domain knowledge. In particular, domain experts are an important knowledge source needed to define the KR of the selected BMServ method, i.e., for example, knowing which types of technical building equipment the BMServ method can be applied to or which sensor readings are needed as an input. The actual total number of persons involved certainly depends on the size of the respective project.

3.3. Step 2: Acquire Assertional knowledge

In this step the developed schema level (TBox) model is populated with assertional knowledge of the respective project or site. The activities related to this step are depicted in Fig. 3.

With the help of domain experts, knowledge sources from the building domain are identified. These are typically disparate sources [17] distributed across heterogeneous domains [62] and examples include:

- Data from BMS stored in structured databases holding knowledge on the operational performance of a building;
- Structured models from applying the Building Information Modelling (BIM) method [63], which provide knowledge on the topology of a building, the utilised materials, the installed technical equipment, the location of equipment and sensors and actuators of a BAS, etc.;
- CAFM systems used to operate buildings, schedule maintenance activities, etc.;
- Human experts, which hold implicit knowledge, e.g. how to operate special pieces of technical equipment, etc. This tacit knowledge needs to be externalised through, e.g., expert interviews to be also considered in a knowledge base;
- Legacy drawings and spreadsheets being the state-of-the-art for information exchange in the past and often still exist in modern buildings today.

Independent from the physical source two different categories of knowledge relevant for ABMS need to be distinguished: *Static Building Knowledge* and *Operational Performance Knowledge*.

We define *Static Building Knowledge* as the knowledge domains covering among others knowledge on the topological structure of the building, the installed elements and technical equipment of a building, such as HVAC system components or building automation hardware and utilised construction material. In addition, it includes knowledge on building occupants, on sequences of operation for the automated control of the building, etc. [62]. More and more static building knowledge becomes available and easily accessible from adopting model-based information exchange in the built environment through the BIM method. An example, which describes the topology of a building using the BOT [53] is presented later in this work (see Code 4). It should be noted that typically this knowledge is extracted once for each building as it does not change frequently or increases significantly in volume over time, e.g. the location of windows in a building is typically fixed for years.

Contrary to static building knowledge we define *Operational Performance Knowledge*. Operational Performance Knowledge of buildings includes, timely readings logged through a BMS. Examples are time series of control actions, temperature readings in HVAC zones, measurements by energy meters or sensor readings associated to a certain piece of equipment. These readings consist of a time stamp and a value, their associated specifications such as unit or associated sensor typically do not frequently change over time. For readings, their amount grows over time and eventually requires Big Data technologies to analyse. As an example Code 1 shows a reading of a temperature sensor formalised using the SOSA [52] and OM ontologies [18]. The only changes appearing in this formalisation for each observation of this reading is the numerical value and the time stamp. All other statements are redundant. In terms of computational efficiency we do not recommend to completely materialise a formal representation of this kind of data and store it in a knowledge base.

Code 1: Example RDF triples in turtle syntax [64] formalising a reading of a temperature measurement [18].

```
osh:Bathroom-temp-Sensor-obs0 a owl:NamedIndividual,
      sosa:Observation ;
sosa:hasFeatureOfInterest osh:Bathroom ;
sosa:hasResult [ a om:Measure ;
       om:hasNumericalValue 1.921e+01 ;
       om:hasUnit om:degreeCelsius ] ;
sosa:madeBySensor osh:Bathroom-temp-Sensor ;
sosa:observedProperty osh:Bathroom-temp ;
sosa:resultTime "2017-03-09T00:58:47+01:00"^^xsd:dateTime.
```

Instead the Ontology-Based Data Access (OBDA) [61] paradigm should be followed. Here a mapping is defined between the source database and triples of the readings are only materialised on demand. Consequently the benefits of both approaches can be exploited: the formal representation of knowledge in a knowledge base and the performant storage of readings in a dedicated database. Finally, the operational performance knowledge needs to be linked to the static building knowledge, e.g. by ensuring that the generated identifiers of observations match identifiers of corresponding instances obtained from static building knowledge.

For the extraction of knowledge from structured sources a large number of ready-to-use tools as well as programming interfaces for the most popular programming languages exist. A versatile, open-source tool to link various kinds of input data formats (CSV, eXtended Markup Language (XML), web Application Programming Interface (API)s, etc.) to RDF is the KARMA tool [65], which also learns from its user and suggest mappings from the original format to the target ontology. To extract knowledge from a BIM model and convert it to RDF triples the IFC-to-LBD[2] and IFC-to-RDF[3] converters exist. These converters parse ifc-SPF files and convert them compliant to a set of ontologies created by the World Wide Web Consortium (W3C) Linked Building Data (LBD) Community Group (CG) [53,66] or ifcOWL [47], respectively. Libraries to implement custom adapters include among others RDFlib[4] and owlready2 [67] for Python and Apache Jena[5] and the OWL API [68] for Java programming language. In case not all data from source systems is supposed to be materialised as triples many tools support OBDA [61], where a mapping is defined between the source system and the knowledge base and triples are materialised on demand. An open-source variant of this is the Ontop [69] tool. Mappings are defined using the R2RML language [70]. In addition, tools for the semi-automatic extraction of knowledge from sources such as printed drawings [46] exist, which allow for instance the extraction of topological relations from a floor plan.

For both categories the corresponding knowledge needs to be extracted from the sources either permanently or on demand and mapped to the developed TBox model. If the TBox model is found to be incomplete, e.g. a unit of a reading is missing, a reiteration (Trigger" Refine TBox") is triggered. If a mapping is possible, the respective knowledge is added and as a final output of the step a populated knowledge base is created, which stores terminological and assertional knowledge on a portfolio of buildings.

To implement the connections to the diverse knowledge sources one or more data engineers are needed, which setup extraction pipelines to legacy data systems. One or more knowledge engineers provide assistance such that the tapped sources can be mapped to the target TBox model. Domain experts are needed to guide data and knowledge engineers to identify the correct knowledge sources and define how to

[2] https://github.com/jyrkioraskari/IFCtoLBD, Last accessed: 24 August 2020

[3] https://github.com/pipauwel/IFCtoRDF, Last accessed: 24 August 2020

[4] https://github.com/RDFLib/rdflib, Last accessed: 24 August 2020

[5] https://jena.apache.org/, Last accessed: 24 August 2020

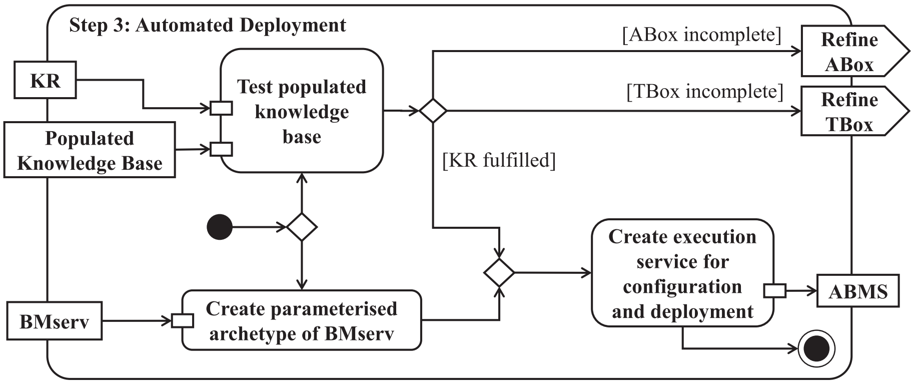

Step 3: Automated Deployment
KR
Populated Knowledge Base
Test populated knowledge base
[ABox incomplete]
Refine ABox
[TBox incomplete]
Refine TBox
[KR fulfilled]
Create execution service for configuration and deployment
ABMS
BMserv
Create parameterised archetype of BMserv

Fig. 4. UML activity diagram depicting the activities of Step 3, which includes activities to setup and create a service for the automated deployment of instances of the selected Building Management Services (BMServ) method. For this purpose the populated knowledge-base is tested against the defined Knowledge Requirements (KR). Refinements of both the TBox model and ABox model can be triggered, if necessary. A parameterised archetype of BMServ method is created. Based on this an execution service is created and the final intended output, an instance of an Automatically enabled Building Management Services (ABMS), is obtained.

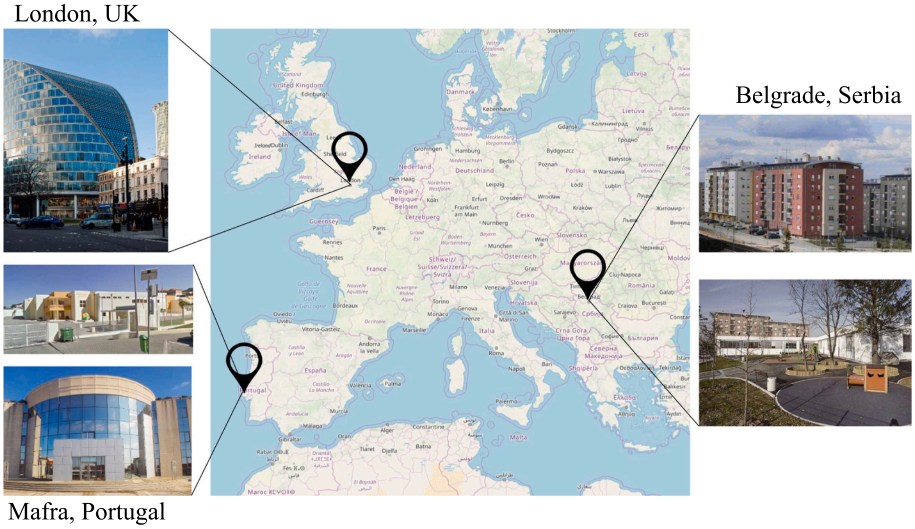


Fig. 5. Location of demo sites in London/ UK, Belgrade/ Serbia and Mafra/ Portugal on a map of Europe including pictures of some of the building stock. Map: © OpenStreetMap contributors used under CC BY-SA 2.0.

perform the transformations necessary to extract, transform and load instances from these sources.

3.4. Step 3: Automated deployment

Finally, the *third step* involves all activities associated to design and implement an application, which performs the actual task on automated deployment of instances of the selected BMServ method. Fig. 4 illustrates the associated activities.

A parameterised archetype is created for the selected BMServ method. The archetype needs to be designed such that instances can be automatically configured and deployed to the target building according to the knowledge retrieved from the populated knowledge base. A final test is performed to ensure that the populated knowledge base fulfils the KR. If necessary, refinements of TBox or abox are triggered. An execution service is created, which performs the task of matching the KR against the populated knowledge base, retrieve static building knowledge and operational performance knowledge, configure instances of a BMServ method from the defined archetype, handle the execution, deploy and store the results. The execution can be triggered timely (e.g. every night) or by a signal (e.g. Alarm signal) from the BMS. As a final result the intended ABMS is obtained.

An example for an execution service has been implemented for the APAR rule set [8], a rule-based FDD method for air handling units. An archetype has been implemented in Java programming language [71] and an implementation of an execution service and a platform capable of handling these kind of services is described in Schneider et al. [22].

Implementing a service, which performs the automated deployment, i.e.: execution of the sequence of activities (match KR with populated knowledge base, configure and instantiate an instance of the archetype based on knowledge retrieved, execute the instance and store results) makes every designed solution an ABMS.

For the implementation and design of the solution architecture and data pipeline(s) one or more platform architect and data engineers are needed. Software developers are needed to design front-end end user application(s). The requirements for the solution are again defined by the domain expert. Knowledge engineers consult if problems related the TBox model appear in the process.

4. Validation in a real-world case study

Within this section, we validate the presented novel ABMS methodology in its ability to fulfil the initially defined requirements. We present results from its validation in a real-world, large-scale pilot setup available from the EU H2020 MOEEBIUS project [29]. In the first part of this section, we provide a detailed description of the studied building portfolio utilised in this real-world case study. Then we present practical examples and experiences obtained from designing a knowledge-based system for the automated deployment of a diverse set of novel BMServ, the modules constituting the MOEEBIUS holistic energy performance optimisation framework [29], developed within the MOEEBIUS project.

Table 2
Overview of building portfolio studied in the described case study including the number of points of each building included in the knowledge base (total number of points per building differs), type of the building and outline of HVAC systems installed. Total gross floor size in square meter (sqm) for all countries in parentheses.

Building	# Points	Type	HVAC systems
Portugal (8.1 k sqm):			
Mafra City Hall	98	Office	Fan coil unit per zone for heating and cooling, ventilation, light dimming
Kindergarten	33	Educational, Office	Split units for cooling
School	71	Educational, Sports, Office	Gas boiler, ventilation
Serbia (434 k sqm):			
Building 1D	94	Residential	District heating
Building 2A	133	Residential	District heating
Building 3G	662	Residential	District heating
Building 6D	768	Residential	District heating
Building 6 L	64	Residential	District heating
UK (22.5 k sqm):			
Moor House	31	Hotel, Retail	Compression chiller, heating
Gifford House	18	Residential	Gas boiler
Jennings House	18	Residential	Gas boiler
Aylmer House	30	Residential	Gas boiler
Total:	2020		

4.1. Description of the pilot setup and challenges

The energy performance of a building is a major concern for all stakeholders involved in the design, construction and operation of a building. Despite the careful design and later operation, often the predicted and actual energy demand of buildings significantly differs. This perceived gap [72] is recognised in the building domain. Within the MOEEBIUS project a holistic energy performance optimisation framework is developed [29], which supports the reduction of the gap. The framework consists of a number of modules each contributing to the gap reduction. The in-depth description of the functionalities of the developed framework and modules is not in the scope of this paper. A detailed description is provided in the deliverables of the project [73–76].

The consortium had to face some challenges, when shifting from the development phase of the different modules to their deployment in the entitled pilot sites. Each module needs specific knowledge to be instantiated at a respective site. This depends, for instance, on the installed technical building equipment, building type or available readings obtained from a BMS, etc. This knowledge was distributed across a diverse set of data bases, spreadsheets and drawings, often written in different languages. Moreover, the portfolio of buildings has been selected on purpose to represent a large variety of technical installations, building types and BMS to highlight the robustness of the developed architecture and modules. In addition, the three selected demonstration sites are distributed across Europe including buildings in the United Kingdom (UK), in Portugal and in Serbia (see map in Fig. 5). An overview of the buildings and their systems is provided in Table 2 and the solution architecture [22] developed and implemented in the project is illustrated in Fig. 8. The different sites comprise a diverse portfolio of building types including office, educational, sports, residential, hotel and retail building types. Each of the buildings is equipped with various HVAC systems including, among others, decentralised Variable Refrigerant Flow (VRF) units and split units, gas boilers and ventilation systems, district heating and compression chillers. Being spread over different European countries, a variety of climatic zones is present in the considered building stock.

The manual collection of the described knowledge, the manual matching of KR of the modules with the collected knowledge and the manual deployment of modules is a cumbersome task, which requires a substantial amount of human resources (cf. Section 1.2). Rather than following this manual path, the consortium decided to design a knowledge-based system to automate the above tasks and in the absence of a dedicated methodology the approach presented in this work has been developed.

The approach presented in Section 3 is used to develop a knowledge-based system, which allows the automated deployment of the described modules of the MOEEBIUS framework. The system automates the manual process above by matching the KR of the BMServ methods with a populated knowledge base, which stores the necessary knowledge about the diverse building portfolio. If the KR of a service can be fulfilled, an instance of it is automatically deployed in the building. Automatically deploying BMServ makes this approach an ABMS.

4.2. Initial activities

One of the first activities is the selection of a qualified KEM (see definition in Section 1.3). As mentioned above, a number of methodologies dedicated for this purpose exist [49,54–58] and most of them have been carefully reviewed by Corcho et al. [26] and Pinto & Martins [25]. The main ‘*stages*' [25] typically occurring in the life cycle of an ontology include ‘specification, conceptualization, formalisation, implementation and maintenance’ [25]. In addition, the need for ontology reuse is emphasised by Pinto & Martins [25].

From the above mentioned KEM we select METHONTOLOGY [55] as the methodology covers the above mentioned stages and emphasises ontology reuse. Here, we do not agree with the assessment of Pinto & Martins stating ‘Existing ontology building methods, such as […] METHONTOLOGY, do not explicitly address reuse, […]’ [25] as ontology reuse is explicitly mentioned and treated in the paper describing the methodology (e.g.’So, you should reuse existing ontologies.’, [55, p. 34]). Moreover, METHONTOLOGY has a proven track of successful deployments with good results [25]. In addition, the intermediate representations documenting the different stages in the methodology are reported to be beneficial by junior ontology developers as these guide the development and allow to execute concrete tasks when developing. This has been found to be beneficial in the setup of the MOEEBIUS project as knowledge representation is relatively new technology in the field and the number of available experts is still low. Additionally, the method can be conducted with the involvement of multiple persons and even offline as the documented intermediate representations can be shared or circulated among involved stakeholders. Ontology development essentially involves multiple experts to ensure its maturity after a couple of iterations and ‘shared’ [77] understanding is established. This activity is in particular fulfilment of R1 (compare Table 1).

The BMServ methods selected for the approach constitute the modules of the MOEEBIUS framework. The KR of the different modules cover an overlapping domain and, hence, the design activities are carried out for all modules at once.

4.3. Step 1: Acquire terminological knowledge

Within this step the terminological knowledge needed for the described validation study is modelled.

Table 3
Excerpt of the ontology requirements specification document of the MOEEBIUS ontology [76].

Name: MOEEBIUS Ontology

Purpose: Within the EU H2020 MOEEBIUS project a set of advanced analysis services for the optimization of energy efficiency in buildings for urban sustainability are developed. For the configuration and deployment of these services dynamic and static data, information and knowledge from buildings and its technical systems is required. The dynamic data […]. The MOEEBIUS ontology is supposed to support the information needs of the advanced analysis services designed to improve the energy efficiency of buildings. It constitutes the nexus between static building data, information and knowledge and the dynamic data.

Scope: The general scope of the ontology is energy-management and decision support in buildings. This includes descriptions of the site and building topology, technical equipment in the buildings and sensors and actuators installed in the building.

Not scope: The ontology does not cover three-dimensional data representing the outer appearance buildings, parts of buildings or technical equipment within the building. The ontology does not provide information on the material of structural elements of the buildings or of the technical equipment.

Implementation language: The ontology will be implemented using the Web Ontology Language 2 (OWL 2). This includes the RDF data model and Resource Description Framework Schema (RDFS) knowledge representation language.

Intended End-Users: The intended end users of the ontology are knowledge engineers and application developers. The actual interface to the ontology will be a RESTful web service, which is translated into SPARQL queries run against the populated knowledge base.

Intended Use: The ontology is intended to be used as the central knowledge representation facility holding all facts related to the use cases of the MOEEBIUS demo sites. Moreover the newly developed analysis services of the MOEEBIUS partners define the information needs to be fulfilled by the ontology.

Ontology Requirements:
1. Non-Functional Requirements
a. The reuse of existing ontologies in the domain should be preferred whenever possible. Potential candidates are BRICK, OM, SOSA, SEAS, CTRLont, SAREF and SAREF4Bldg, BOT.
2. Functional Requirements/ Competency questions:
a. What are the temperature sensors in this storey?
b. Which sensors are in this room?
c. Which final spaces are served by Boiler B?
d. What is the ID of the temperature sensor in this room?
e. Which VRF supplies heat to this room?
f. Which is the total energy meter of this building?
g. What is ID of the active power sensor of this building, storey, room?
h. What is the ID of the simulated value of the temperature in Room 52?
i. Which pieces of HVAC equipment are fed by VRF5/VRF9?
j. Rooms served/ fed by VRF5/VRF9
k. Which pieces of HVAC equipment supply the HVAC zone of Room < name > ?
l. What is the cassandra ID/ BEPS ID of point < name > ?
m. What HVAC equipment is installed in building < name > ?
n. What is the quantity and unit of point < name > ?
[…]

Sources of Knowledge: Schemes of the MOEEBIUS demo sites; General domain knowledge on building management systems and building energy performance simulation, MOEEBIUS Common Information Model defined in T3.2.

Table 4
Excerpt of the glossary of terms defined for the ontology [76].

Concepts:	Air flow, refrigerant flow, water flow, lighting system, […] point, energy meter, electrical energy meter, thermal energy meter, […] Unit, Celsius, kilowatt, kilowatt hours, parts per million, […]
Verbs:	contains, hasLocation, hasPart, isPartOf, feeds, isFedBy, […]

4.3.1. Specification

As first step of the METHONTOLOGY method we create an Ontology Requirements Specification Document (ORSD). This document is the first conceptual document of the ontology development and allows the description of the purpose of the ontology. Additionally, it contains a definition of its scope and what should be not in the scope. Moreover, a clarification of which implementation language to choose and which users will work with the ontology is included. The document includes the definition of (semi-)formal KR, which are later used to evaluate the developed ontology. This is undertaken by defining a set of CQs [54]. For the definition of the CQ a series of expert interviews has been conducted and the list of semi-formal questions has been curated in an iterative manner. Table 3 presents an excerpt of the specified ORSD for the MOEEBIUS ontology (The full document can be found in the projects documentation [76]).

4.3.2. Conceptualisation

The description collected in the ORSD is used to collect relevant terms in a glossary of terms. These have been collected in initial brain storming sessions with involved building experts and have undergone some refinements over iterations. The glossary contains concepts, instances, verbs and properties of the domain of interest, e.g. terms used to describe technical equipment in the building, sensor types, etc. An excerpt of the full glossary [76] is presented in Table 4.

4.3.3. Formalisation

The previously defined semi-formal conceptualisations are to be formalised in this step. The up to this point created documents are the starting point for this task. Ontology-based knowledge representation allows the definition of concepts, their hierarchy, instances and verbs and again, the METHONTOLOGY method provides tools for documenting this effort in a set of documents.

An important activity at this stage is to extensively consider ontology reuse, which is also emphasised in ontology engineering literature [25,26]. A number of tools exist, which support the ontology engineer to search for concepts in existing ontologies, such as the Linked Open Vocabularies (LOV)[6] [78]. In addition to searching for terms, literature review of published ontologies is an important task to identify potential candidate ontologies. We review existing ontologies in the building domain [9,11,23,30,35,79] and conclude that for the described use case and application BRICK ontology [30] fits most of the defined KR. Hence, instead of developing an own ontology from scratch we reuse BRICK in its latest stable version (v1.0.3) as of writing and extend from it where needed. An excerpt of the mapped and extended concepts and verbs of the BRICK ontology to terms listed in the glossary of terms is provided in Table 5 and the full table is available in Kontes et al. [76]. In addition, some terms to describe units and quantities as described in the OM [31] and relationships from Dublin Core ontology [80] are utilised.

4.3.4. Implementation

Based on the mapping tables we implement the ontology using the Protégé tool [81] and OWL. The resulting ontology structure is depicted in Fig. 6. We do a full import of BRICK ontology (owl:imports) and reuse only some axioms (partly import) from OM. The resulting MOEEBIUS ontology is stored in a web repository and can be accessed via its Uniform Resource Identifier (URI)·[7]

4.3.5. Evaluation and maintenance

Before leaving the first step in the methodological approach an evaluation of the developed ontology is necessary. This is inline with the final step of the METHONTOLOGY method. The evaluation is undertaken testing the capability of the ontology to fulfil the initially defined KR. The KR are denoted as CQs within the ORSD (see Table 3).

To perform the test we setup an empty triple store,[8] load the developed MOEEBIUS ontology to the triple store including all imports, that is BRICKFrame and BRICK ontologies (both v1.0.3) [30]. The reasoning

[6] https://lov.linkeddata.es/dataset/lov, Last accessed: 24 August 2020

[7] https://w3id.org/moeebius/MOEEBIUSOntology#, Last accessed: 24 August 2020

[8] GraphDB, Sirma AI, https://www.ontotext.com/products/graphdb/, Last accessed: 24 August 2020

Table 5
Excerpt of defined mappings and extensions between concepts and verbs of the BRICK ontology [30], OM [31] and Dublin Core ontology [80] to the glossary of terms.

Term	Mapped Entity	Specialised from
Building	Brick:Building	–
Room	Brick:Room	–
Radiator	Moeebius:Radiator	brick:Space_Heater
Proximity sensor	Moeebius:NOD_Proximity_Sensor	brick:PIR_Sensor
Temperature	om2:Temperature	–
Cold	Moeebius:Cold	om2:Energy
feeds	Brick:feeds	–
isLocatedIn	Brick:hasLocation	–
Has unit	Moeebius:hasUnit	–
Cassandra DB Identifier	Moeebius:cassandraID	dcterms:identifier

mechanism of the triple store supporting the OWL 2 RL profile [82] is activated to materialise implicit knowledge and allow the retrieval of it via a query language. In addition, we load triples to the triple store populating the knowledge base with instances. The instances have been semi-automatically created for the KUBIK test building [83], a lab facility used for testing in the project provided by the project partner Tecnalia. The triples for the lab building are available for reference in the MOEEBIUS web repository[9] and an excerpt is presented in Code 4.

The in the ORSD semi-formally defined CQs (cf. Table 3) are formalised using SPARQL [84]. All implemented CQs are tested against the populated knowledge base and are successfully answered. SPARQL implementations for all queries are available online.[10]

From all defined CQs, we present in the following two example queries and their SPARQL transcription. We present results obtained from the queries and highlight why the results can only be obtained with the support of a reasoner in the knowledge base. The remaining queries can be examined in the web repository as mentioned above. The first question.

CQ1*'What are the temperature sensors in this storey?'*

is a question motivated by some BMServ, which need to know all temperature sensors in a building storey. The question is formalised in Code 2 with a binding for the test building. In addition, results of the query listed in Code 2 for the test building are given in Table 6.

Code 2: SPARQL query implementing CQ1.

```
PREFIX [...]
SELECT ?Sensors
WHERE {
  BIND( Abox:SI01.BL00.ST00 as ?Location )
  ?Location rdf:type brick:Floor .
  ?Location bf:hasPart ?Room .
  ?Room bf:isLocationOf ?Sensors .
  ?Sensors rdf:type brick:Temperature_Sensor .
}
```

The query exploits implicitly stated knowledge and only returns correct results with a reasoner enabled. In particular, the definition of bf:hasPart and bf:isLocationOf as inverse properties of bf:isPartOf and bf:hasLocation, respectively, allows to determine the demanded sensors as only the inverse is used in the instance ontology of the KUBIK building. In real world settings different modelling may happen (sensor has a location or location has a sensor). With the reasoner activated the query retrieves results in both cases. In addition, the definition of the concept brick:Temperature_Sensor as generalisation of more specialised temperature sensors in BRICK allows to retrieve the sensors. In other words, Abox:SI01.BL00.P00.M1.RT rdf:type brick:Return_Water_Temperature_Sensor implies Abox:SI01.BL00.P00.M1.RT rdf:type brick:Temperature_Sensor and, hence, the sensor is returned.

The second question

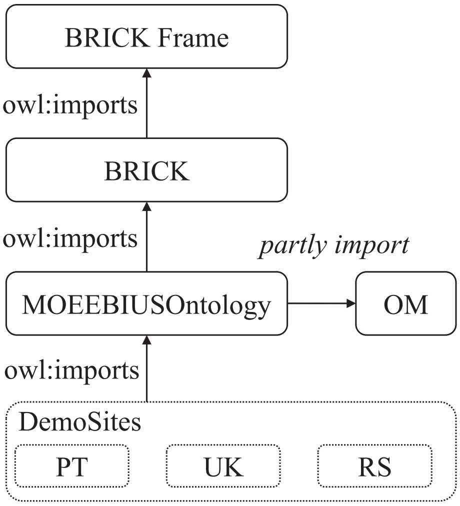


Fig. 6. Overview of developed ontology structure.

CQ2*'What HVAC equipment is installed in building X?'*

discussed here matches needs of BMServ and types of HVAC equipment available in a building. The CQ is formalised as a SPARQL query as presented in Code 3 and example results for the test building are given in Table 7.

[9] https://raw.githubusercontent.com/MOEEBIUS/MOEEBIUS_Ontology/master/MOEEBIUSOntology/KUBIKSite-0.0.3.ttl, Last accessed: 24 August 2020

[10] https://github.com/MOEEBIUS/MOEEBIUS_Ontology/blob/master/CompetencyQuestions.md, Last accessed: 24 August 2020

Table 6
All URIs of temperature sensors in the test building [83], which are on the same building storey. Results for query presented in Code 2.

?Sensors
Abox:SI01.BL00.P00.M1.RT
Abox:SI01.BL00.P00.M1.ST
Abox:SI01.BL00.P00.M1.T
Abox:SI01.BL00.P00.M1.T2
Abox:SI01.BL00.P00.M1.TI
Abox:SI01.BL00.P00.M2.RT
Abox:SI01.BL00.P00.M2.ST
Abox:SI01.BL00.P00.M3.RT
Abox:SI01.BL00.P00.M3.ST

Table 7
All URIs of results for query presented in Code 3.

?Equipment
Abox:SI01.BL00.CL00.CH01
Abox:SI01.BL00.CL00.CH02
Abox:SI01.BL00.CL00.CHP
Abox:SI01.BL00.CL00.SP00.AHU00
Abox:SI01.BL00.CL00.SP00.BLR00
Abox:SI01.BL00.P00.M1.FC
Abox:SI01.BL00.P00.M2.FC
Abox:SI01.BL00.P00.M3.FC

Code 3: SPARQL query implementing CQ2. Binding for test building KUBIK [83].

```
PREFIX [...]
SELECT ?Equipment
WHERE {
BIND( Abox:SI01.BL00 as ?building )
?building rdf:type brick:Building .
?building bf:hasPart ?location .
?location bf:isLocationOf ?Equipment .
?Equipment rdf:type brick:Equipment .
}
```

The query can only be answered if reasoning is activated. In particular, the transitive object property bf:hasPart allows to retrieve all explicit and implicit locations in the respective building. In other words, the building Abox:SI01.BL00 does not have an explicit relationship brick:hasPart with the room Abox:SI01.BL00.ST00.SP03 in the test triples. This is beneficial as building topologies may differ in their segmentation from the general building to a room. Next, the inverse property of bf:hasLocation, bf:isLocationOf, is used to be able to determine all equipment located in the building.

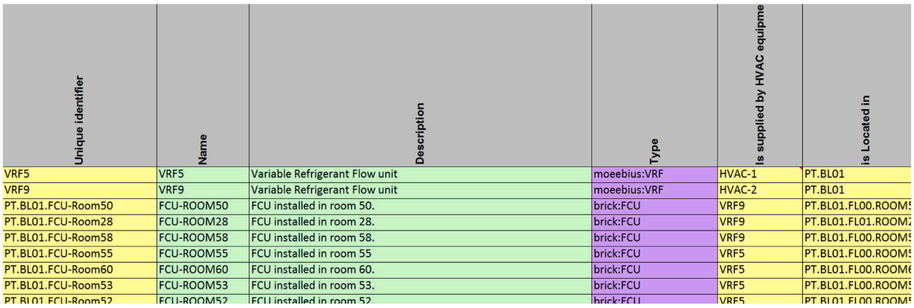
A continuous monitoring of the KR in terms of maintaining the developed schema is necessary over the life cycle of the ontology as KR might change or evolve. The result of this step is a tested TBox model, the ontology, and its documentation generated while executing the METHONTOLOGY method. Also KR as formulated in the ORSD are a result. The ontology files are versioned and are kept in named graphs, when loaded to the knowledge base to allow the complete removal and update over different iterations.

In the described case study it took about three weeks for one knowledge engineer to obtain the TBox model, including receiving inputs by domain experts, e.g. on competency questions.

4.4. Step 2: Acquire Assertional knowledge

In this section we present in detail how respective knowledge has been acquired from various sources to populate the knowledge base of the MOEEBIUS project on the building portfolio considered in the project (see overview in Section 4.1 and Table 2). As defined in Section 3.3 we differentiate knowledge sources into two main categories:

1: *Static Building Knowledge* of the building portfolio of interest including, for example, the topology of the building, its technical equipment, the location of a sensor, technical specification of the sensor including, for instance, its data type or unit, etc.;



Unique identifier	Name	Description	Type	Is supplied by HVAC equipme	is Located in
VRF5	VRF5	Variable Refrigerant Flow unit	moeebius:VRF	HVAC-1	PT.BL01
VRF9	VRF9	Variable Refrigerant Flow unit	moeebius:VRF	HVAC-2	PT.BL01
PT.BL01.FCU-Room50	FCU-ROOM50	FCU installed in room 50.	brick:FCU	VRF9	PT.BL01.FL00.ROOM5
PT.BL01.FCU-Room28	FCU-ROOM28	FCU installed in room 28.	brick:FCU	VRF9	PT.BL01.FL01.ROOM2
PT.BL01.FCU-Room58	FCU-ROOM58	FCU installed in room 58.	brick:FCU	VRF9	PT.BL01.FL00.ROOM5
PT.BL01.FCU-Room55	FCU-ROOM55	FCU installed in room 55	brick:FCU	VRF5	PT.BL01.FL00.ROOM5
PT.BL01.FCU-Room60	FCU-ROOM60	FCU installed in room 60.	brick:FCU	VRF5	PT.BL01.FL00.ROOM6
PT.BL01.FCU-Room53	FCU-ROOM53	FCU installed in room 53.	brick:FCU	VRF5	PT.BL01.FL00.ROOM5
PT.BL01.FCU-Room52	FCU-ROOM52	FCU installed in room 52.	brick:FCU	VRF5	PT.BL01.FL00.ROOM5

Fig. 7. Example view on spreadsheet gathering knowledge on installed equipment in a Portuguese pilot building.

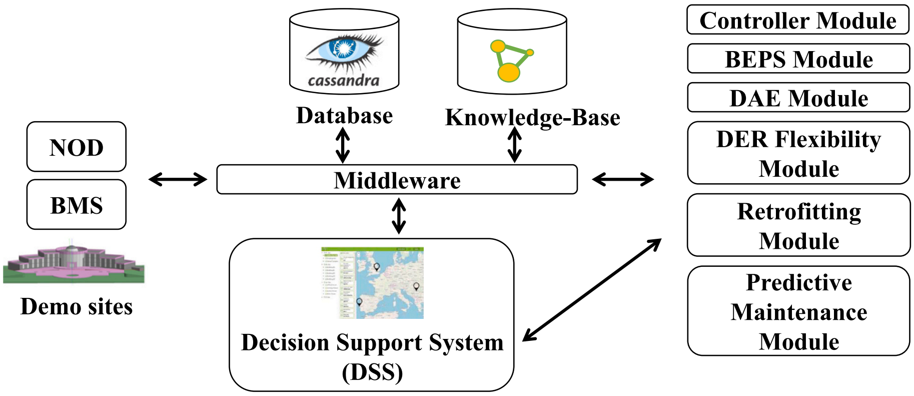
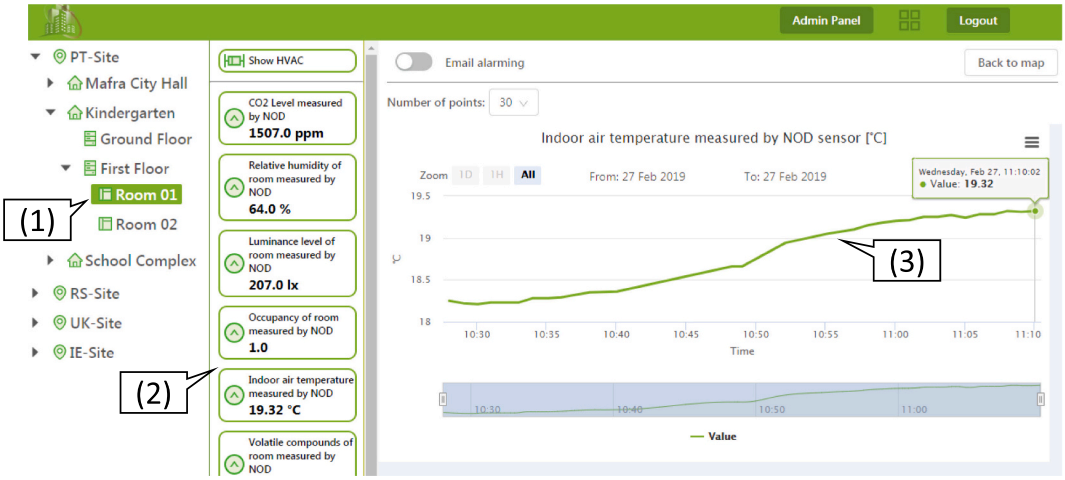


Fig. 8. Overview of deployed architecture and implementation of the MOEEBIUS solution. Map: © OpenStreetMap contributors used under CC BY-SA 2.0.



CO2 Level measured by NOD
1507.0 ppm

Relative humidity of room measured by NOD
64.0 %

Luminance level of room measured by NOD
207.0 lx

Occupancy of room measured by NOD
1.0

Indoor air temperature measured by NOD
19.32 °C

Volatile compounds of room measured by NOD

Number of points: 30

Indoor air temperature measured by NOD sensor [°C]

From: 27 Feb 2019 To: 27 Feb 2019

Wednesday, Feb 27, 11:10:02
● Value: 19.32

Fig. 9. Screen shot of dynamically from knowledge-base configured DSS GUI. (1) - Automatically rendered building topology. (2) - Sensors and actuators of locations retrieved from contextual query; (3) - Time series retrieved from No-SQL database through identifier provided by knowledge-base (OBDA, [61]).

2: *Operational Performance Knowledge* of the different building portfolio of interest including, among others, timely readings collected from sensors connected to a BMS.

Acquiring static building knowledge of sites distributed across three countries is a challenging task. As the adoption of the BIM method [63] yet is limited in most countries the respective knowledge is only available as drawings printed on paper or pdf, textual descriptions, etc. Hence, an approach has been established to allow the (semi-)automated extraction of this knowledge. To keep the approach as simple as possible and without the need for installing specific software an Excel spreadsheet based on the COBie standard [32] has been designed. Using the COBie standard for elicitation of assertional knowledge is in fulfilment of R2 (compare Table 1) and allows to use the approach also in future, when this knowledge can be exported from software tools. The respective entries in the spreadsheet are filled manually by site owners and local experts from the disparate site documentation. This process had to be iterated multiple times until the final consolidated spreadsheet has been composed. Finally, a script implemented in Python programming language is utilised to perform the batch-wise extraction, transformation and loading of the different pilot site spreadsheets to the triple store. An excerpt of the content of the spreadsheet for one demo building located in Mafra, Portugal is presented in Fig. 7. In the process of establishing the populated building knowledge base in total two knowledge engineers, three contact persons for each site and five building experts have been involved. Extracting the assertional knowledge and consolidating the spreadsheets took about two month with bi-weekly meetings. The developed system is maintained for the course of demonstration by the development team.

An excerpt of triples instantiating the first testing site, the KUBIK building [83], is given in Code 4 for reference. As mentioned above, the complete file is available from a web repository. The generated RDF triples are uploaded to a triple store, where the built-in reasoner for the OWL 2 RL profile [82] is activated.

The MOEEBIUS framework is deployed in a web-based platform [22], which enables the different associated tasks and provides the needed components, such as connecting the BMS from demo sites, a database for time series readings, a knowledge-base, a middleware for orchestration of all services and the different modules of the framework (see Fig. 8). Within this architecture the knowledge base is accessible from externally via the SPARQL API of the utilised triple store or a RESTful API performing the conversion of contextual questions to SPARQL as part of the middleware in the MOEEBIUS architecture (see overview in Fig. 8).

To access the operational performance knowledge of the building we choose to store the timely readings collected from the respective buildings in a single, performant database. In this case we use a NoSQL-database.[11] All time series data is accessible from this database via a RESTful API. As mentioned above we refrain from materialising operational performance data upfront (see triples of counterexample in

[11] Apache Cassandra, http://cassandra.apache.org, Last accessed: 24 August 2020

Code 1). Instead we follow the OBDA paradigm [61,69] and store unique identifiers of each signal in the knowledge base. The service implementing the automated deployment then retrieves this data using the identifier from the API of the NoSQL-database.

Code 4: Excerpt of RDF triples in turtle format [64] instantiating the KUBIK test building [83]. The complete file is available online as mentioned above.

```
# KUBIK Building
Abox:SI01.BL00 rdf:type owl:NamedIndividual ,
  brick:Building ;
  bf:isPartOf Abox:SI01 .

# Chiller
Abox:SI01.BL00.CL00.CH01 rdf:type owl:NamedIndividual ,
  brick:Chiller ;
  bf:hasLocation Abox:SI01.BL00.CL00 .

# Return water sensor
Abox:SI01.BL00.CL00.CH01.RT00 rdf:type owl:NamedIndividual ,
  brick:Chilled_Water_Return_Temperature_Sensor ;
  MOEEBIUS:cassandraID "SI01.BL00.CL00.CH01.RT00" ;
  moeeius:hasQuantity om:Temperature ;
  MOEEBIUS:hasUnit om:degreeCelsius ;
  bf:hasLocation Abox:SI01.BL00.CL00 ;
  bf:isPointOf Abox:SI01.BL00.CL00.CH01 .

# NOD unit in Room SP03
Abox:SI01.BL00.ST00.SP03.NOD00.TEMP rdf:type
owl:NamedIndividual ;
 [...]
  bf:hasLocation Abox:SI01.BL00.ST00.SP03 .
```

4.5. *Step 3: Automated deployment*

In the final activity the populated knowledge base obtained in Step 2 is tested a last time against the formalised KR implemented as SPARQL queries as mentioned above.

The modules of the framework are implemented in different programming languages by the project partners. The code has been redesigned, such that parameterised archetypes of all modules are created and these can be instantiated from knowledge retrieved through queries from the knowledge base. Three modules are described in detail in Sections 4.5.1–4.5.3. The KR of each module are denoted as CQs and examples are provided how these can be formalised to SPARQL queries.

The execution service is realised in a web-based platform and the solution is deployed in the MOEEBIUS solution architecture (see Fig. 8 and [22]). Services are implemented and setup, which perform the required tasks for the automated deployment of the different modules of the MOEEBIUS framework.

In the subsequent sections we describe in detail the automated deployment of the three examples modules. Hence, this is in particular fulfilment of R3 (compare Table 1). All presented results are obtained using a triple store with a reasoner activated, which supports the OWL2 RL profile. We discuss in detail why the results can only be obtained with the support of a reasoner in the knowledge base.

4.5.1. Dynamic and real time graphical user Interface of the decision support system

One module of the overall solution is an integrated Decision Support System (DSS), which is accessible through a dynamic real time capable Graphical User Interface (GUI)[12] (see Fig. 9). The GUI is implemented as stand alone, single page web application and is constantly deployed on a web server. By constantly querying the knowledge base it automatically configures itself dependent on the retrieved assertional knowledge and deploys to the respective buildings, sensors and technical equipment. Through the web application, building managers are facilitated to perform individual, aggregated and comparative assessment of the building performance through visualisation of collected monitoring data in real time. In addition alarming based on simple rules and actuation of, e.g., set points is possible.

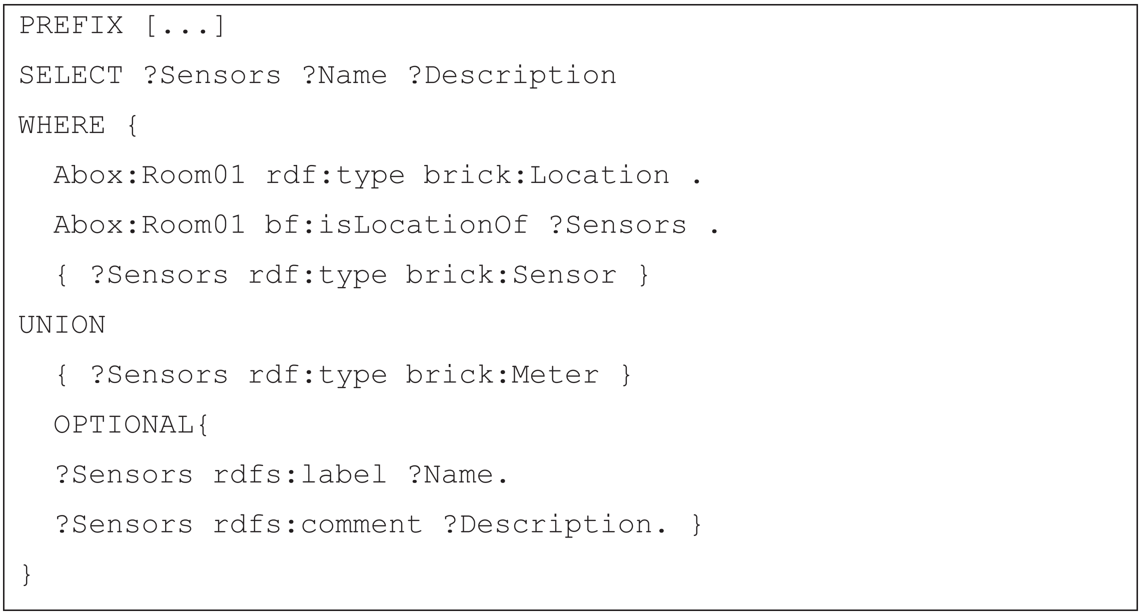
Similar capabilities as described for the developed GUI are available in state of the art BMS. For the deployment of these a significant amount of site knowledge is needed and the deployment in a specific building is typically a cumbersome, time consuming process. Typical CQs are:

1. *What are the considered sites?*
2. *What are the buildings at site x?*
3. *What are the storeys in building x?*
4. *Which sensors are at location x?*
5. *Which actuators are at location x?*
6. *Which unit has the value obtained from sensor x?*
7. *What is the database identifier of sensor x?*

…

In avoidance of the error-prone and cumbersome manual process the content of the GUI is dynamically retrieved from the knowledgebase. For instance, if the occupant of a building clicks on a location, e.g. Room 01 ((1), Fig. 9), a contextual query is sent to the knowledge base to obtain all sensors at this location and populate the GUI ((2), Fig. 9). For reference, the respective SPARQL query is listed below.

Code 5: SPARQL query to retrieve all sensors of a specific location (Abox:Room01).



```
PREFIX [...]
SELECT ?Sensors ?Name ?Description
WHERE {
  Abox:Room01 rdf:type brick:Location .
  Abox:Room01 bf:isLocationOf ?Sensors .
  { ?Sensors rdf:type brick:Sensor }
UNION
  { ?Sensors rdf:type brick:Meter }
  OPTIONAL{
  ?Sensors rdfs:label ?Name.
  ?Sensors rdfs:comment ?Description. }
}
```

The query exploits implicit knowledge, which can be retrieved through the SPARQL query as it is inferred by the active reasoner. In this particular case, the definition of brick:hasLocation as inverse property of brick:isLocationOf is utilised to retrieve the required knowledge. In other words, there does not exist an explicit triple Abox:Room01 bf:isLocationOf < SomeSensor > in the knowledge base.

4.5.2. Occupant profiling engine

As building occupants have a tremendous impact on the energy performance of a building it is important to consider their effect in building operation [72]. Within MOEEBIUS the occupant profiling engine [73] implements a functionality to extract context-aware user preferences and identify comfort (dis)satisfaction in zones from settings. In parallel, it introduces the occupant as an active element of the building operation strategy. Thermal and visual comfort profiles are learnt through the use of a naïve Bayes classifier and models are trained on a daily basis to reflect changes in occupant satisfaction [73]. The module has various knowledge needs to be deployed. Where an excerpt of respective CQs is listed below:

1. *What are the building storeys of building x?*
2. *What are the rooms of building storey x?*
3. *What are the sensors located in room x?*
4. *What is the outdoor air temperature of building x?*
5. *What are the lighting zones in building x?*
6. *What is the identifier of sensor x?*
7. *What are is the luminance set point for lighting zone x?*

…

The module distinguishes between explicit (dis)-comfort of users, which refers to occupant (dis)comfort as it can be extracted from physical actions he or she undertakes to customise the lighting settings to his liking and implicit comfort, which on the other hand, refers to the occupant comfort as it can be inferred by a lack of action. Fig. 10 illustrates how we infer implicit and explicit (dis)comfort from user non-actions and actions, respectively [73].

[12] https://MOEEBIUS.itegia.de, Last accessed: 24 August 2020

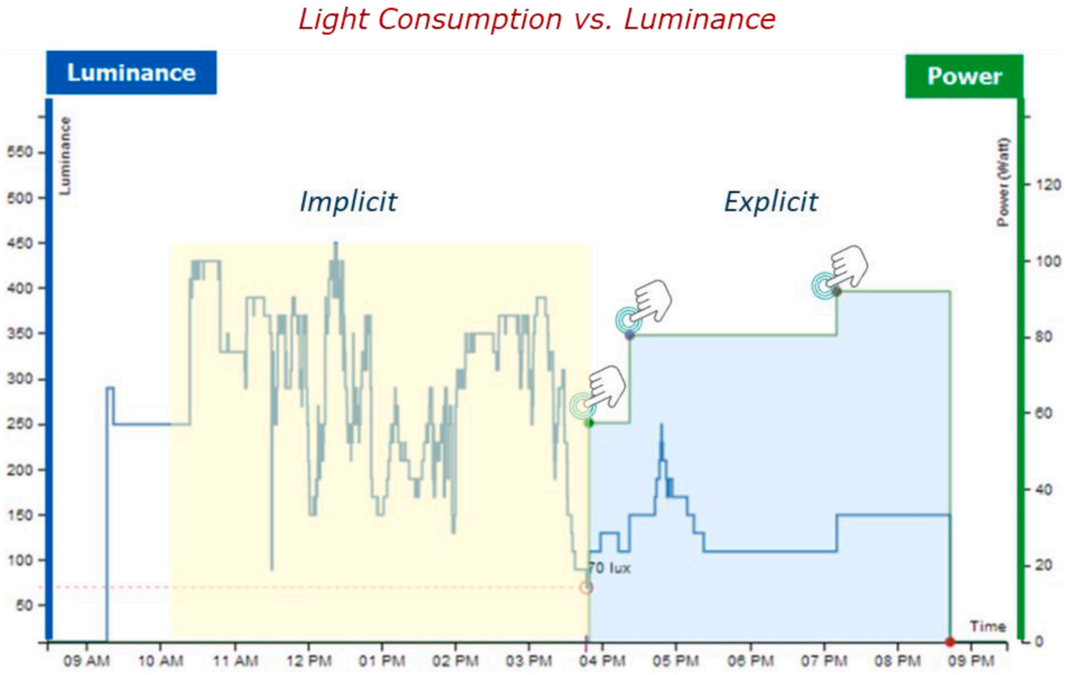


Fig. 10. Implicit and explicit visual comfort with user interactions. Left y-axis: Luminance in lux (blue), right y-axis: Power in Watt (green), x-axis: time. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

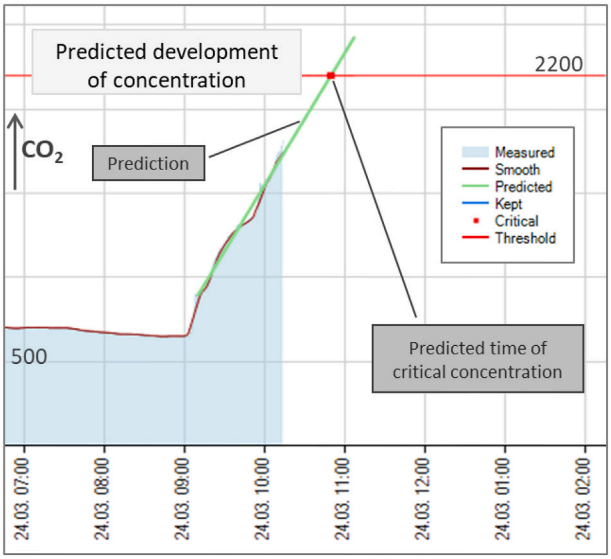
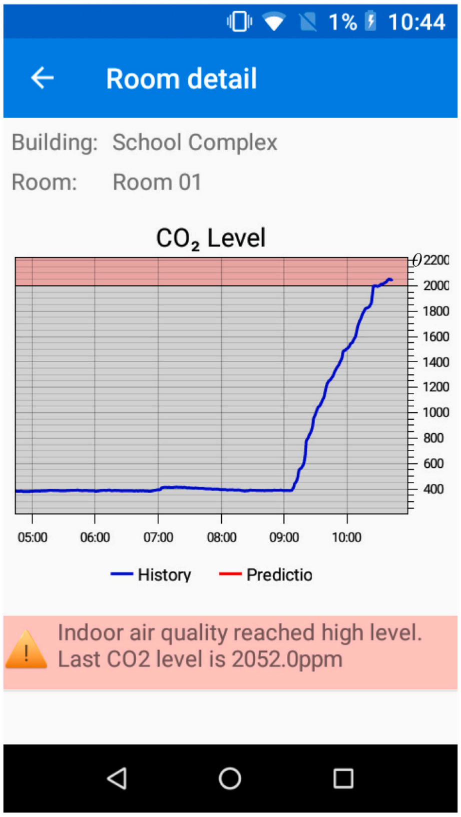


Fig. 11. Results of the module for predicting the course of CO2 concentration measured on 24 March 2019 in a room of the pilot building. Y-axis: CO2-concentration in ppm, x-axis: time.

To automatically deploy the module as depicted in Fig. 10 a service is implemented, which is hosted on a web server. It retrieves the necessary knowledge formalised as SPARQL queries from the knowledge base to configure an instance of the service per lighting zone discovered. It deploys the service on the web server and performs an update of the user profile periodically every night. One example query listed in Code 6 allows to retrieve all lighting zones of building (Here the binding is for Abox:PT.BL01).



Room detail

Building: School Complex

Room: Room 01

Indoor air quality reached high level.
Last CO2 level is 2052.0ppm

Fig. 12. Screen shot of dynamically from knowledge-base deployed end-user application running on a smart phone with alarm message displayed.

Code 6: SPARQL query implementing CQ 5 for the specific building Abox:PT.BL01.

```
PREFIX [...]
SELECT ?Zones ?Name ?Description
WHERE {
  Abox:PT.BL01 rdf:type brick:Building .
  Abox:PT.BL01 bf:hasPart ?Zones .
  ?Zones rdf:type brick:Lighting_Zone .
OPTIONAL{
  ?Zones rdfs:label ?Name .
  ?Zones rdfs:comment ?Description. }
}
```

Again the query retrieves knowledge from the knowledge base, which is only implicitly available and needs to be made available explicitly through a reasoner. It exploits the transitive property bf:hasPart to determine, which lighting zones are part of the building. In other words, the triple Abox:PT.BL01 bf:hasPart < SomeLigtingZone > does not explicitly exist in the knowledge base.

4.5.3. Predictive ventilation assistant

The last module presented in this study is the predictive ventilation assistant [74], which is a BMServ method to improve the indoor air quality in buildings. Through the prediction of indoor air quality metrics, such as Carbondioxide (CO2) concentration, the occupants can be informed or a ventilation system can be triggered to allow demand-based ventilation.

Again a service is implemented, hosted on a web server, which implements the automated deployment if a suitable building is found in the knowledge base. Initially, the service retrieves all buildings from the knowledge base, which are equipped with a CO2 sensor in HVAC zones. For each zone an instance of the module is created and it is deployed automatically on the web server.

The results for one automatically deployed module is presented in Fig. 11. Based on the operational performance knowledge, i.e. time series readings of CO2 concentration in the zone, it predicts the future CO2 concentration and sends warnings, if a threshold is exceeded.

The module is accompanied with a smart phone application (see Fig. 12), which allows end users to visualise the results and perform counter measures if alarms are displayed. Similar to the GUI presented in Section 4.5.1 the constantly deployed smart phone application is configured dynamically from retrieved knowledge on the sites and buildings.

For the automated deployment of the BMServ method the following CQs need to be answered by the knowledge base:

1. *In which buildings are CO2 sensors?*
2. *What is the HVAC zone of point x?*
3. *What is the location of point x?*
4. *What is the database identifier of point x?*
5. *Which buildings are at site x?*

…

As an example for all CQs the SPARQL query listed in Code 7 formalises CQ 1 and the query listed in Code 8 formalises CQ 4.

Code 7: SPARQL query to retrieve all buildings with CO2 sensors in the knowledge-base.

```
PREFIX [...]
SELECT ?Building ?Sensors
WHERE {
  ?Sensors rdf:type brick:Point .
  ?Sensors rdf:type brick:CO2_Level_Sensor
  ?Sensors bf:hasLocation ?Location .
  ?Location bf:isPartOf ?Building .
  ?Building rdf:type brick:Building .
}
```

Again a reasoner is needed to allow the query listed in Code 7 to return the expected results. The transitive property bf:isPartOf allows to determine the building the sensor is situated in, without traversing the topology path of the building from room, storeys up to the building. In addition, the definition of bf:CO2_Level_Sensor as a generalisation of all CO2 sensors allows to retrieve the respective sensors, even if they have been classified as other more special concepts.

Code 8: Template SPARQL query to obtain for sensor with the URI" Abox:*PARAMETER*" its identifier for time series data.

```
PREFIX [...]
SELECT ?CassandraID
 WHERE {
  { Abox:$PARAMETER$ rdf:type brick:Sensor }
  UNION
  { Abox:$PARAMETER$ rdf:type brick:Meter }
 Abox:$PARAMETER$ moeebius:cassandraID ?CassandraID .
}
```

5. Discussion

Despite the achieved results as presented some limitations and obstacles have been identified in the course of the implementation and usage of the proposed methodological approach.

Many approaches reviewed in Section 2 implement ontology-based ABMS using OWL for the implementation, despite the availability of a large number of other knowledge representation languages [26,85]. In the course of this work this is found beneficial as the implemented ontologies can be easily reused and integrated with other W3C standards such as SPARQL, etc., as well as a large number of existing formal knowledge models already exists in the domain for reuse. Nevertheless, in future the benefits and drawbacks of other implementation languages and formalisms such as Flogic [86] should be evaluated.

An experience from the real-world case study is that the technologies involved in the proposed approach are rather new for domain experts and technical staff in the field. The adoption of the technology in the domain will also depend on the training of experts and staff on the related topics to allow them to maintain the developed systems and solutions. Extensive, open training material is needed in this regard. A notable contribution in this direction is the Summer School of Linked Data in Architecture in Construction (SSoLDAC) [87], which was initiated in 2019 and aims at training students as well as industry experts on related technologies.

In the presented work we rely on the COBie standard [32] to extract and collect assertional knowledge on the respective sites and then formalise it using a converter. The standard is gaining attention in particular during the operation phase of buildings. In our implementation we had to add additional columns in the spreadsheets to capture all required inputs. Hence, further work is needed for the revision of existing standards and on harmonising open schemata such as BRICK and COBie.

An experience obtained during the case study is that the chosen formal approach to model the knowledge domain of interest has been found beneficial by the project participants and helped to develop a consistent understanding of the various buildings considered in the case study. In particular, the possibility to answer the defined CQs helped software engineers during the development of their applications. Still future research and experiments are needed to determine, whether this can be generalised as a characteristic of formal approaches.

6. Conclusion

In this work we present a novel methodological approach for the design of knowledge-based systems for the automated deployment of Building Management Services (BMServ), the Automatically enabled Building Management Services (ABMS) methodology. It covers the three main steps and their iterations to design these systems: (1) representation and capturing of terminological knowledge of a building, its equipment and automation system based on a well-established Knowledge Engineering Method (KEM); (2) representation and capturing of assertional knowledge on projects from disparate sources based on open standards; and (3) use of the acquired knowledge for the automated deployment of BMServ. We validate our approach by deploying it in a real-world large scale use case stemming from the MOEEBIUS project [29]. We capture terminological knowledge on the building domain using a qualified KEM [55]. We reuse and extend the BRICK ontology [30] and partly the Ontology of Units and Measures (OM) [31] for representing terminological knowledge. We populate a knowledge-base for the portfolio of buildings available from the project, which are distributed across different countries and climates in Europe, equipped with different technical equipment and Building Automation Systems (BAS). We do this by extracting the assertional knowledge based on open standards, i.e. the Construction Operations Building information exchange (COBie) standard [32], semi-automatically from spreadsheets for the different sites. The stored knowledge is used to support and automate the deployment of a diverse set of novel BMServ developed within the MOEEBIUS project, which contribute each to tackle the perceived gap between predicted and actual energy demand of buildings.

The results obtained in the case study show that the novel methodological approach fulfils the initially defined requirements. One core activity of the approach includes the use of a qualified KEM so that best practices in knowledge engineering, e.g. ontology reuse, are followed. In addition, the structured acquisition of knowledge from sources is supported and the design of knowledge-based systems for the automated deployment of BMServ methods is possible. With these characteristics the approach provides the basis for a wide spread adoption of BMServ in the building domain. Potentially new BMServ methods are introduced in future, which are accompanied with a (semi-)formal description of their Knowledge Requirements (KR). These KR could then be automatically matched with the present constellation of technical equipment, sensors, etc. and instantiated in a building equipped with a building knowledge base.

Despite the achievements, future topics of research exist and should be related to:

- Investigating other implementation languages for knowledge representation, which potentially have beneficial characteristics, e.g. closed world assumption [85]. Here, Flogic is an interesting candidate [86];
- A plethora of formal domain models of the built environment exists [35,79]. To support the reuse of existing ontologies it would be interesting to use proposed core ontologies, e.g. Building Topology Ontology (BOT) [53], to support cross-domain interoperability and portability of developed ABMS. Alignments to a number of domain ontologies have been proposed in the past [88];
- The usefulness of the approach is extensively validated in a case study related to a large-scale, multi-national set of buildings and diverse set of novel BMServ. Further empirical evaluation is needed to quantify the actual impact of the methodological approach on improving the energy efficiency of buildings by automatically deploying BMServ;
- The results obtained from the real-world large-scale case study are very promising. However, we plan additional validation studies on more buildings and including industry partners to further foster available empirical evidence;
- The modelling of terminological knowledge of a domain is still a manual task and requires considerable amount of human effort. In future it would be interesting to evaluate automated methods in this regard, e.g. for knowledge discovery [37] or bootstrap schemata from existing standards as demanded by Zhou et al. [24].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the anonymous reviewers for their thoughtful comments and help to improve the manuscript. The authors would like to gratefully acknowledge the generous funding provided by the European Union's Horizon 2020 research and innovation programme through the MOEEBIUS project under grant agreement No. 680517.

Appendix A

The prefixes utilised in the code examples presented in this paper are listed in Table A.1 and the graphical notation for Unified Modelling Language (UML) [51] activity diagrams is illustrated in Fig. A.1.

Table A.1
Alphabetically sorted namespaces and prefixes used in this work.

Prefix	Value
rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#
rdfs	http://www.w3.org/2000/01/rdf-schema#
owl	http://www.w3.org/2002/07/owl#
Abox	https://w3id.org/moeebius/DemoSites#
BDO	https://w3id.org/ibp/BasicDatatypeOntology#
bf	https://brickschema.org/schema/1.0.3/BrickFrame#
brick	https://brickschema.org/schema/1.0.3/Brick#
geo	http://www.w3.org/2003/01/geo/wgs84_pos#
moeebius	https://w3id.org/moeebius/MOEEBIUSOntology#
om	http://www.ontology-of-units-of-measure.org/vocabularies/om-2/
osh	https://w3id.org/ibp/osh/OpenSmartHomeDataSet#
sosa	http://www.w3.org/ns/sosa/

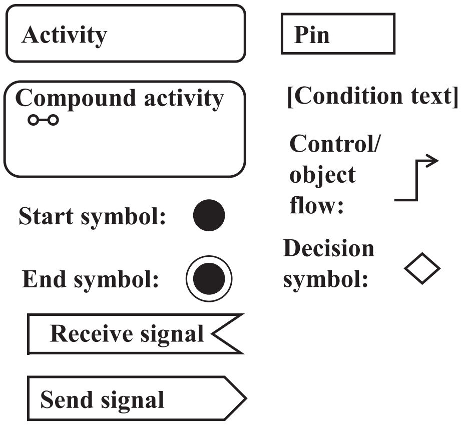


Fig. A.1. Nomenclature of the utilised graphical notation elements of UML [51] activity diagrams.

References

[1] S. Solomon, D. Qin, M. Manning, Z. Chen, M. Marquis, K. Averyt, M. Tignor, H. Miller (Eds.), Contribution of Working Group I to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change, Cambridge University Press, Cambridge, UK, 2007(ISBN: 9780521705967).

[2] AGEB e.V, Auswertungstabellen zur Energiebilanz Deutschland, Tech. Rep. 2017, Arbeitsgemeinschaft Energiebilanzen, Berlin, Germany, last accessed: 24 August 2020. URL, September 2017. https://ag-energiebilanzen.de/index.php?article_id=29fileName=bilanz17d.xlsx.

[3] EU, Directive (EU) 2018/844 on the energy performance of buildings, european Parliament, Brussels, Belgium and Strasbourg, France, Last accessed: 24 August 2020 (2018) URL http://data.europa.eu/eli/dir/2018/844/oj.

[4] EN 15232, Energy performance of buildings – Impact of Building Automation, Controls and Building Management, European Committee for Standardization (CEN), Brussels, Belgium, Last accessed: 24 August 2020 (2017). URL https://standards.cen.eu/dyn/www/f?p=204:110:0::::FSP_PROJECT,FSP_ORG_ID:41064,6228 cs=136D7260DB9278974E1080CA59AF1C88F.

[5] K. Roth, D. Westphalen, M. Feng, P. Llana, Energy impact of commercial building controls and performance diagnostics: Market characterization, energy impact of building faults and energy savings potential, Tech. Rep. D0180, TIAX LLC for DOE, 2005 last accessed: 24 August 2020. URL https://ntrl.ntis.gov/NTRL/dashboard/searchResults/titleDetail/PB2006100567.xhtml.

[6] S. Katipamula, M.R. Brambley, Methods for fault detection, diagnostics, and prognostics for building systems–a review, Part I, HVAC&R Research 11 (1) (2005) 3–25, https://doi.org/10.1080/10789669.2005.10391123.

[7] S. Katipamula, M.R. Brambley, Methods for fault detection, diagnostics, and prognostics for building systems–a review, Part II, HVAC&R Research 11 (2) (2005) 169–187, https://doi.org/10.1080/10789669.2005.10391133.

[8] J. Schein, S.T. Bushby, N.S. Castro, J.M. House, A rule-based fault detection method for air handling units, Energy and Buildings 38 (12) (2006) 1485–1492, https://doi.org/10.1016/j.enbuild.2006.04.014.

[9] E. Corry, P. Pauwels, S. Hu, M. Keane, J. O'Donnell, A performance assessment ontology for the environmental and energy management of buildings, Autom. Constr. 57 (2015) 249–259, https://doi.org/10.1016/j.autcon.2015.05.002.

[10] G.F. Schneider, P. Pauwels, S. Steiger, Ontology-based modeling of control logic in building automation systems, IEEE Transactions on Industrial Informatics 13 (6) (2017) 3350–3360, https://doi.org/10.1109/TII.2017.2743221.

[11] N.M. Tomašević, M.V. Batić, L.M. Blanes, M.M. Keane, S. Vrane, Ontology-based facility data model for energy management, Advanced Engineering Informatics 29 (4) (2015) 971–984, https://doi.org/10.1016/j.aei.2015.09.003.

[12] S. Plesser, M.N. Fisch, C. Pinkernell, T. Kurpick, B. Rumpe, The energy navigator - a web based platform for functional quality Mangement in buildings, Proceedings of the 10th International Conference for Enhanced Building Operations (ICEBO), Kuwait City, Kuwait, 2010, pp. 1–10 last accessed: 24 August 2020. URL https://arxiv.org/ftp/arxiv/papers/1409/1409.2364.pdf.

[13] Federal Statistical Office of Germany, Anzahl der Wohngebäude in Deutschland in den Jahren 2000 bis 2017 (in 1.000) (Engl.: Number of residential building in Germany from 2000 until 2017 (in thousands)), last accessed: 24 August 2020. URL, 2019. https://de.statista.com/statistik/daten/studie/70094/umfrage/wohngebaeude-bestand-in-deutschland-seit-1994/.

[14] H. Dibowski, Automatically enabled analytics in buildings and smart homes, at-Automatisierungstechnik 65 (9) (2017) 641–649, https://doi.org/10.1515/auto-2017-0025.

[15] H. Dibowski, O. Holub, J. Rojíček, Knowledge-Based Fault Propagation in Building Automation Systems, in: Proceedings of the International Conference on Systems Informatics, Modelling and Simulation (SIMS), IEEE Computer Society, Riga, Latvia, 2016, pp. 124–132, https://doi.org/10.1109/SIMS.2016.22.

[16] Z. Shi, W. O'Brien, Development and implementation of automated fault detection and diagnostics for building systems: a review, Autom. Constr. 104 (2019) 215–229, https://doi.org/10.1016/j.autcon.2019.04.002.

[17] E. Curry, J. O'Donnell, E. Corry, S. Hasan, M. Keane, S. O'Riain, Linking building data in the cloud: integrating cross-domain building data using linked data, Adv. Eng. Inform. 27 (2) (2013) 206–219, https://doi.org/10.1016/j.aei.2012.10.003.

[18] G.F. Schneider, M.H. Rasmussen, P. Bonsma, J. Oraskari, P. Pauwels, Linked building data for modular building information modelling of a smart home, in: J. Karlshoj (Ed.), 12th European Conference on Product and Process Modelling (ECPPM), CRC Press, Copenhagen, Denmark, 2018, pp. 407–414.

[19] L. Petnga, M. Austin, An ontological framework for knowledge modeling and decision support in cyber-physical systems, Adv. Eng. Inform. 30 (1) (2016) 77–94, https://doi.org/10.1016/j.aei.2015.12.003.

[20] G.F. Schneider, Y. Kalantari, G.D. Kontes, S. Steiger, D.V. Rovas, An ontology-based tool for automated configuration and deployment of technical building management services, in: J. Grunewald (Ed.), Proceedings of Central European Symposium on Building Physics (CESBP), Fraunhofer IRB Verlag, Dresden, Germany, 2016, pp. 865–872 (ISBN: 9783816797982).

[21] H. Dibowski, J. Vass, O. Holub, J. Rojíček, Automatic Setup of Fault Detection Algorithms in Building and Home Automation, in: Proceedings of the International Conference on Emerging Technologies and Factory Automation (ETFA), Institute of Electrical and Electronic Engineers (IEEE), Berlin, Germany, 2016, pp. 1–6, https://doi.org/10.1109/ETFA.2016.7733622.

[22] G.F. Schneider, Y. Kalantari, G.D. Kontes, G. Grün, S. Steiger, A platform for automated technical building management services using ontology, in: F. Bosché, I. Brilakis, R. Sacks (Eds.), Lean and Computing in Construction Congress (LC3): Proceedings of the Joint Conference on Computing in Construction (JC3), vol. I, Heraklion, Greece, 2017, pp. 321–328, , https://doi.org/10.24928/JC3-2017/0279.

[23] I. Esnaola-Gonzalez, J. Bermúdez, I. Fernandez, A. Arnaiz, Semantic prediction assistant approach applied to energy efficiency in tertiary buildings, Semantic Web 9 (6) (2018) 735–762, https://doi.org/10.3233/SW-180296.

[24] Z. Zhou, Y.M. Goh, L. Shen, Overview and analysis of ontology studies supporting development of the construction industry, J. Comput. Civ. Eng. 30 (6) (2016) 1–14, https://doi.org/10.1061/(ASCE)CP.1943-5487.0000594.

[25] H.S. Pinto, J.P. Martins, Ontologies: how can they be built? Knowl. Inf. Syst. 6 (4) (2004) 441–464, https://doi.org/10.1007/s10115-003-0138-1.

[26] O. Corcho, M. Fernández-López, A. Gómez-Pérez, Methodologies, tools and languages for building ontologies. where is their meeting point? Data & Knowledge Engineering 46 (1) (2003) 41–64, https://doi.org/10.1016/S0169-023X(02)00195-7.

[27] M.H. Rasmussen, P. Pauwels, C.A. Hviid, J. Karlshøj, Proposing a central AEC ontology that allows for domain specific extensions, in: F. Bosché, I. Brilakis, R. Sacks (Eds.), Proceedings of Lean and Computing in Construction Congress (LC3): Proceedings of the Joint Conference on Computing in Construction (JC3), vol. i, 2017, pp. 237–244, , https://doi.org/10.24928/JC3-2017/0153 Heraklion, Greece.

[28] J. Euzenat, P. Shvaiko, Ontology Matching, 2nd edition, vol. 18, Springer, Heidelberg, Germany, 2013, https://doi.org/10.1007/978-3-642-38721-0.

[29] A. Romero, B. Tellado, T. Tsitsanis, Moeebius energy performance optimization framework in buildings for urban sustainability, Proceedings of 41st IAHS WORLD CONGRESS Sustainability and Innovatioen for the Future, Albufeira, Portugal, 2016, pp. 1–10 ISBN: 9789899894945, Last accessed: 24 August 2020. URL http://dsp.tecnalia.com/bitstream/handle/11556/361/MOEEBIUS_IAHS2016_full%20paper.pdf.

[30] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Berges, D. Culler, R.K. Gupta, M.B. Kjærgaard, M. Srivastava, K. Whitehouse, Brick : metadata schema for portable smart building applications, Appl. Energy 226 (2018) 1273–1292, https://doi.org/10.1016/j.apenergy.2018.02.091.

[31] H. Rijgersberg, M. Wigham, J. Top, How semantics can improve engineering processes: a case of units of measure and quantities, Adv. Eng. Inform. 25 (2) (2011) 276–287, https://doi.org/10.1016/j.aei.2010.07.008.

[32] E.W. East, Construction operations building information exchange (cobie):

Requirements definition and pilot implementation standard, Tech. Rep. ERDC/CERL TR-07-30, US Army Engineer Research and Development Center, 2007 last accessed: 24 August 2020. URL https://apps.dtic.mil/dtic/tr/fulltext/u2/a491932.pdf.

[33] B. Zhong, H. Wu, H. Li, S. Sepasgozar, H. Luo, L. He, A scientometric analysis and critical review of construction related ontology research, Autom. Constr. 101 (2019) 17–31, https://doi.org/10.1016/j.autcon.2018.12.013.

[34] W. Kim, S. Katipamula, A review of fault detection and diagnostics methods for building systems, Science and Technology for the Built Environment 24 (1) (2018) 3–21, https://doi.org/10.1080/23744731.2017.1318008.

[35] P. Pauwels, S. Zhang, Y.-C. Lee, Semantic web technologies in AEC industry: a literature overview, Autom. Constr. 73 (2017) 145–165, https://doi.org/10.1016/j.autcon.2016.10.003.

[36] Smart applications for the maintenance of large buildings: How to achieve ontology-based interoperability at the information level, in: A. D'Elia, L. Roffia, G. Zamagni, F. Vergari, P. Bellavista, A. Toninelli, S. Mattarozzi, Institute of Electrical and Electronics Engineers (IEEE) (Eds.), The IEEE Symposium on Computers and Communications, Riccione, Italy, 2010, pp. 1077–1082, , https://doi.org/10.1109/ISCC.2010.5546639.

[37] H. Wicaksono, K. Aleksandrov, S. Rogalski, An intelligent system for improving energy efficiency in building using ontology and building automation systems, in: F. Kongoli (Ed.), Automation, InTech, 2012, pp. 531–548, , https://doi.org/10.5772/48006.

[38] J. Caffarel, S. Jie, J. Olloqui, R. Martínez, Bat-mp: An ontology-based energy management platform, in: J. Bravo, D. López-de Ipiña, F. Moya (Eds.), Ubiquitous Computing and Ambient Intelligence, Springer, Berlin, Germany, 2012, pp. 9–16, , https://doi.org/10.1007/978-3-642-35377-2_2.

[39] J. Han, Y. Jeong, I. Lee, A rule-based ontology reasoning system for context-aware building energy management, in: Y. Wu, G. Min, N. Georgalas (Eds.), Proceedings of International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing, Pervasive Intelligence and Computing, Liverpool, UK, 2015, pp. 2134–2142, , https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.317.

[40] A. Kučera, T. Pitner, Semantic bms: allowing usage of building automation data in facility benchmarking, Adv. Eng. Inform. 35 (2018) 69–84, https://doi.org/10.1016/j.aei.2018.01.002.

[41] H. Dibowski, O. Holub, J. Rojíček, Ontology-Based Automatic Setup of Virtual Sensors in Building Automation Systems, in: Proceedings of the International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Institute of Electrical and Electronic Engineers (IEEE), Lisbon, Portugal, 2016, pp. 375–381, https://doi.org/10.1109/ICUMT.2016.7765388.

[42] R. Ferrari, H. Dibowski, S. Baldi, A message passing algorithm for automatic synthesis of probabilistic fault detectors from building automation ontologies, IFAC-PapersOnLine 50 (1) (2017) 4184–4190, https://doi.org/10.1016/j.ifacol.2017.08.809.

[43] H. Dibowski, O. Holub, J. Rojček, Ontology-based fault propagation in building automation systems, International Journal of Simulation–Systems, Science & Technology 18 (3) (2017) 1–14, https://doi.org/10.5013/IJSSST.a.18.03.01.

[44] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Berges, D. Culler, R. Gupta, M.B. Kjærgaard, M. Srivastava, K. Whitehouse, Brick: Towards a Unified Metadata Schema for Buildings, in: Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments, Association for Computing Machinery, Palo Alto, USA, 2016, pp. 41–50, https://doi.org/10.1145/2993422.2993577.

[45] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, Ontology library (final), Tech. Rep. D18, Istituto di Scienze e Tecnologie della Cognizione - Consiglio Nazionale delle Ricerche (ISTC-CNR), Trento, Italy, 2003 last accessed: 24 August 2020. URL http://wonderweb.man.ac.uk/deliverables/documents/D18.pdf.

[46] P. Häfner, V. Häfner, H. Wicaksono, J. Ovtcharova, Semi-automated ontology population from building construction drawings, in: A. Fred, J.L.G. Dietz, K. Liu, J. Filipe (Eds.), Proceedings of 5th International Conference Knowledge Engineering and Ontology Development, Vilamoura, Portugal, 2013, pp. 379–386, , https://doi.org/10.5220/0004626303790386.

[47] P. Pauwels, W. Terkaj, EXPRESS to OWL for construction industry: towards a recommendable and usable ifcOWL ontology, Autom. Constr. 63 (2016) 100–133, https://doi.org/10.1016/j.autcon.2015.12.003.

[48] ISO 50001, Energy management systems - Requirements with guidance for use, International Standardisation Organisation (ISO), Geneva, Switzerland, 2018 Last accessed: 24 August 2020. URL https://www.iso.org/standard/69426.html.

[49] N.F. Noy, D.L. McGuinness, Ontology Development 101: A Guide to Creating Your First Ontology, Tech. Rep. SMI-2001-0880, Stanford University, Stanford, USA, 2001 last accessed: 24 August 2020. URL https://protege.stanford.edu/publications/ontology_development/ontology101.pdf.

[50] J. Ploennigs, B. Hensel, H. Dibowski, K. Kabitzsch, BASont - a Modular, Adaptive Building Automation System Ontology, in: Proceedings of the 38th Annual Conference on IEEE Industrial Electronics Society (IECON), IEEE Industrial Electronics Society, Montreal, Canada, 2012, pp. 4827–4833, https://doi.org/10.1109/IECON.2012.6389583.

[51] ISO/IEC 19505, Information technology – Object Management Group Unified Modeling Language (OMG UML), International Standardisation Organisation (ISO), Geneva, Switzerland, 2012 Last accessed: 24 August 2020. URL https://www.iso.org/standard/32624.html.

[52] A. Haller, J. Krzystof, S. Cox, D. le Phuoc, K. Taylor, M. Lefrançois, Semantic Sensor Network Ontology, Permanent, URL https://www.w3.org/TR/vocab-ssn/ Recommendation, World Wide Web Consortium (W3C), Cambridge, USA, 2017.

[53] M.H. Rasmussen, M. Lefrançois, G.F. Schneider, P. Pauwels, BOT: the Building Topology Ontology of the W3C Linked Building Data Group, Semantic Web, 1–21Accepted, Last accessed: 24 August 2020. URL, 2020. http://www.semantic-web-journal.net/system/files/swj2279.pdf.

[54] M. Grüninger, M.S. Fox, Methodology for the design and evaluation of ontologies, Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, Canada, 1995, pp. 1–10 last accessed: 24 August 2020. URL http://www.eil.utoronto.ca/wp-content/uploads/enterprise-modelling/papers/gruninger-ijcai95.pdf.

[55] M. Fernández-López, A. Gómez-Pérez, N. Juristo, Methontology: from ontological art towards ontological engineering, Tech. Rep. SS-97-06, Association for the Advancement of Artificial Intelligence, Stanford, USA, 1997 last accessed: 24 August 2020. URL http://oa.upm.es/5484/1/METHONTOLOGY_.pdf.

[56] Y. Sure, S. Staab, R. Studer, On-to-knowledge methodology (otkm), in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, Springer, Berlin, Germany, 2004, pp. 117–132, , https://doi.org/10.1007/978-3-540-24750-0_6.

[57] M.C. Suárez-Figueroa, A. Gómez-Pérez, M. Fernández-López, The neon methodology for ontology engineering, in: M.C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, A. Gangemi (Eds.), Ontology Engineering in a Networked World, Springer, Berlin, Germany, 2012, pp. 9–34, , https://doi.org/10.1007/978-3-642-24794-1_2.

[58] A. De Nicola, M. Missikoff, A lightweight methodology for rapid ontology engineering, Commun. ACM 59 (3) (2016) 79–86, https://doi.org/10.1145/2818359.

[59] W3C OWL Working Group, OWL 2 Web Ontology Language, Permanent, URL https://www.w3.org/TR/owl2-overview/ Recommendation, World Wide Web Consortium (W3C), Cambridge, USA, 2012.

[60] D. Garijo, Widoco: A wizard for documenting ontologies, in: C. d'Amato, M. Fernandez, V. Tamma, F. Lecue, P. Cudré-Mauroux, J. Sequeda, C. Lange, J. Heflin (Eds.), Proceedings of the International Semantic Web Confernce, Springer International Publishing, Cham, Switzerland, 2017, pp. 94–102, , https://doi.org/10.1007/978-3-319-68204-4_9.

[61] A. Poggi, D. Lembo, D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, Linking data to ontologies, in: S. Spaccapietra (Ed.), Journal on Data Semantics X, Springer, Berlin, Germany, 2008, pp. 133–173, , https://doi.org/10.1007/978-3-540-77688-8_5.

[62] G.F. Schneider, A. Bougain, P.S. Noisten, M. Mitterhofer, Information requirement definition for BIM: A life cycle perspective, in: R.J. Scherer, S.E. Christodoulou (Eds.), Proceedings of the 11th European Conference on Product and Process Modelling (ECPPM), CRC Press, Limassol, Cyprus, 2016, pp. 225–233 (ISBN: 9781138032804).

[63] R. Sacks, C. Eastman, G. Lee, P. Teicholz, BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers, John Wiley & Sons, Hoboken, USA, 2018, https://doi.org/10.1002/9781119287568.

[64] D. Beckett, T. Berners-Lee, E. Prud'hommeaux, G. Carothers, RDF 1.1 Turtle Terse RDF Triple Language, Permanent URI, https://www.w3.org/TR/turtle/.

[65] C.A. Knoblock, P. Szekely, J.L. Ambite, A. Goel, S. Gupta, K. Lerman, M. Muslea, M. Taheriyan, P. Mallick, Semi-automatically mapping structured sources into the semantic web, in: E. Simperl, P. Cimiano, A. Polleres, O. Corcho, V. Presutti (Eds.), The Semantic Web: Research and Applications, Springer, Berlin, Germany, 2012, pp. 375–390, , https://doi.org/10.1007/978-3-642-30284-8_32.

[66] M. Bonduel, J. Oraskari, P. Pauwels, M. Vergauwen, R. Klein, The IFC to linked building data converter–current status, in: M. Poveda-Villalon, P. Pauwels, A. Roxin (Eds.), 6th Linked Data in Architecture and Construction Workshop (LDAC), vol. 2159, CEUR-WS.org, London, UK, 2018, pp. 34–43 last accessed: 24 august 2020. URL http://ceur-ws.org/Vol-2159/04paper.pdf.

[67] J.-B. Lamy, Owlready: ontology-oriented programming in python with automatic classification and high level constructs for biomedical ontologies, Artif. Intell. Med. 80 (2017) 11–28, https://doi.org/10.1016/j.artmed.2017.07.002.

[68] M. Horridge, S. Bechhofer, The OWL API: a Java API for OWL ontologies, Semantic web 2 (1) (2011) 11–21, https://doi.org/10.3233/SW-2011-0025.

[69] D. Calvanese, B. Cogrel, S. Komla-Ebri, R. Kontchakov, D. Lanti, M. Rezk, M. Rodriguez-Muro, G. Xiao, Ontop: answering SPARQL queries over relational databases, Semantic Web 8 (3) (2017) 471–487, https://doi.org/10.3233/SW-160217.

[70] S. Das, S. Sundara, R. Cyganiak, R2RML: RDB to RDF Mapping Language, Permanent, https://www.w3.org/TR/r2rml/ Recommendation, World Wide Web Consortium (W3C), Cambridge, USA, 2012.

[71] G. Kontes, Moeebius/apar_rules_java: Apar rules and diagnostics, last accessed: 24 August 2020 Zenodo, 2018.

[72] P. De Wilde, The gap between predicted and measured energy performance of buildings: a framework for investigation, Autom. Constr. 41 (2014) 40–49, https://doi.org/10.1016/j.autcon.2014.02.009.

[73] T. Papapolyzos, A. Tsitsanis, D. Despina, P. Andriopoulos, K. Tsatsakis, T. Tsitsanis, N. Tsadaris, M. Bucur, B. Tellado, A. Romero, MOEEBIUS DER Flexibility Analysis, Aggregation and Forecasting Module, Tech. Rep. D6.2, MOEEBIUS Consortium, Brussels, Belgium, 2018 last accessed: 24 August 2020. URL https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5b8e52a59appId=PPGMS.

[74] Z. Schindler, J. Rojicek, J. Malanik, G. Kontes, S.A. Ozdemir, C. Beder, MOEEBIUS Predictive and Sanitary Maintenance advisor module development and VR Environment Final version, Tech. Rep. D6.4, MOEEBIUS consortium, Brussels, Belgium, 2018 last accessed: 24 August 2020 https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5bad5c35bappId=PPGMS.

[75] P.M.A. van Tooren, A. Nales, A.W. Stam, P. de Agustin, A. Armijo, MOEEBIUS Retrofitting advisor and investment evaluation module development Final version, Tech. Rep. D6.6, MOEEBIUS consortium, Brussels, Belgium, 2018 last accessed: 24

August 2020 https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds = 080166e5bbda74ddappId = PPGMS.

[76] G.D. Kontes, H. Qiu, L. Wang, M. Visser, G.F. Schneider, N. Panagiotidou, G. Grün, V. Sánchez, A. Romero-Amorrortu, P. de Agustin-Camacho, J. Rojicek, Z. Schindler, J. Malanik, A. Tsitsanis, T. Papapolyzos, C. Beder, M. Klepal, P. van Tooren, A. Nales, A. Stam, C. Malavazos, MOEEBIUS Integrated DSS (Final Version), Tech. Rep. D6.8, MOEEBIUS consortium, Brussels, Belgium, 2018 last accessed: 24 August 2020 https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds = 080166e5ba59ecebappId = PPGMS.

[77] R. Studer, R. Benjamins, D. Fensel, Knowledge engineering: principles and methods, Data Knowl. Eng. 25 (1) (1998) 161–197, https://doi.org/10.1016/S0169-023X(97)00056-6.

[78] P.-Y. Vandenbussche, G.A. Atemezing, M. Poveda-Villalón, B. Vatant, Linked open vocabularies (LOV): a gateway to reusable semantic vocabularies on the web, Semantic Web 8 (3) (2017) 437–452, https://doi.org/10.3233/SW-160213.

[79] L. Daniele, F. den Hartog, J. Roes, Created in close interaction with the industry: The smart appliances REFerence (SAREF) ontology, in: R. Cuel, R. Young (Eds.), Proceedings of the 7th International Workshop Formal Ontologies Meet Industries (FOMI), CEUR-WS.org, Berlin, Germany, 2015, pp. 100–112, , https://doi.org/10.1007/978-3-319-21545-7_9.

[80] S. Weibel, J. Kunze, C. Lagoze, M. Wolf, Dublin core metadata for resource discovery, Internet Engineering Task Force RFC 2413 (222), 132 (1998) last accessed: 24 August 2020 https://www.rfc-editor.org/pdfrfc/rfc2413.txt.pdf.

[81] M.A. Musen, The protégé project: a look back and a look forward, AI matters 1 (4) (2015) 4–12, https://doi.org/10.1145/2757001.2757003.

[82] B. Motik, B. Cuenca Grau, I. Horrocks, Z. Wu, A. Fokoue, C. Lutz, OWL 2 Web Ontology Language Profiles (Second Edition), Permanent, http://www.w3.org/TR/owl2-profiles/ Recommendation, World Wide Web Consortium (W3C), Cambridge, USA, 2012.

[83] J.A. Chica, I. Apraiz, P. Elguezabal, M.O. Rrips, V. Sánchez, B. Tellado, Kubik: Open building approach for the construction of an unique experimental facility aimed to improve energy efficiency in buildings, Open House International 36 (1) (2011) 63–72 last accessed: 24 August 2020 www.openhouse-int.com/pdf/OHI%20Vol.36%20No.1.pdf#page = 64.

[84] E. Prud'hommeaux, A. Seaborne, The SPARQL query language for RDF, Permanent, https://www.w3.org/TR/rdf-sparql-query/.

[85] J. Domingue, D. Fensel, J.A. Hendler, Handbook of Semantic Web Technologies, Springer, Berlin, Germany, 2011, https://doi.org/10.1007/978-3-540-92913-0.

[86] T. Tudorache, L. Alani, Semantic web solutions in the automotive industry, in: S. Biffl, M. Sabou (Eds.), Semantic Web Technologies for Intelligent Engineering Applications, Springer International Publishing, Cham, Switzerland, 2016, pp. 297–326, , https://doi.org/10.1007/978-3-319-41490-4_12.

[87] J. Beetz, M. Bonduel, R. de Klerk, K. McGlinn, M.H. Rasmussen, P. Pauwels, M. Poveda-Villalón, G.F. Schneider, W. Terkaj, A. Wagner, The 1th summer school of linked data in architecture and construction, in: R. de Klerk, J.N. Beirao (Eds.), Proceedings of the 1th Summer School on Linked Data in Architecture and Construction (SSoLDAC), Lisbon, Portugal, 2019, pp. 1–20, , https://doi.org/10.5281/zenodo.3596829.

[88] G.F. Schneider, Automated ontology matching in the architecture, engineering and construction domain-a case study, in: M. Poveda-Villalon, P. Pauwels, R.D. Klerk, A. Roxin (Eds.), Proceedings of the 7th Linked Data in Architecture and Construction Workshop (LDAC), Vol. 2389 CEUR-WS.org, Lisbon, Portugal, 2019, pp. 35–49 last accessed: 24 August 2020 http://ceur-ws.org/Vol-2389/03paper.pdf.