FAKULTÄT FÜR !NFORMATIK
Faculty of Informatics

Building Energy Management Optimization based on a Semantic Abstraction Layer

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Dipl.-Ing. Daniel Schachinger, BSc BSc (WU)
Matrikelnummer 00825086

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dr.techn. Wolfgang Kastner

Diese Dissertation haben begutachtet:

Jörn Plönnigs

Alexander Fay

Wien, 16. August 2018

Daniel Schachinger

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Building Energy Management Optimization based on a Semantic Abstraction Layer

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Dipl.-Ing. Daniel Schachinger, BSc BSc (WU)
Registration Number 00825086

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dr.techn. Wolfgang Kastner

The dissertation has been reviewed by:

Jörn Plönnigs

Alexander Fay

Vienna, 16th August, 2018

Daniel Schachinger

Erklärung zur Verfassung der Arbeit

Dipl.-Ing. Daniel Schachinger, BSc BSc (WU)
Bauernstraße 23, 4645 Grünau

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 16. August 2018

Daniel Schachinger

Acknowledgments

First and foremost, I am deeply grateful to my advisor, Wolfgang Kastner, for his steady support over the past several years at the Research Division of Automation Systems. His guidance, motivation, advice, and feedback were always valuable to drive my research work. Many thanks go to all my former and present colleagues who contributed to a positive and friendly working environment. In particular, I want to thank Andreas Fernbach, Stefan Gaida, Jürgen Pannosch, and Christoph Stampfel for their cooperation and contribution to our joint publications that are integrated in this thesis. In addition, this dissertation was financially supported by a netidee scholarship and the Lions Club Wien St. Stephan. Special thanks to my referees for taking the time to review this dissertation. Moreover, I want to thank my friends for the time we spent together as a welcome change to work. Last but not least, I owe my deepest gratitude to Tina, my parents, Theresia and Karl, and my brother, Benjamin. This thesis would not have been possible without their patience and support.

Abstract

Efficient operation is essential in order to tackle the increasing energy needs of residential and commercial buildings. In this context, building automation systems (BASs) provide a basis for advanced control of devices and appliances in building energy management systems (BEMSs) with respect to comfort compliance as well as energy consumption. Furthermore, proceeding smart grid integration of buildings benefits comprehensive energy management. Nevertheless, the heterogeneity in smart grid communication and building automation (BA), the lack of machine-readable semantics, and the tailoring of BEMSs to specific building types or comfort domains limit the development of flexible and reusable energy management solutions for cost-efficient and large-scale deployment.

This thesis presents the design of a BEMS based on a semantic abstraction layer that separates the generic optimization from the building environment. First, Web services are used for interoperable integration of BASs while an IP-centric protocol stack is defined for homogeneous smart grid communication. Second, BEMS-related information is abstracted and merged in an ontology as part of the semantic abstraction layer. In addition, a service-oriented interface is elaborated for semantic machine-to-machine communication based on this ontology. Third, the machine-readable information modeled in the ontology is exploited to support automated configuration and operation of optimization in BEMSs. A workflow for the extraction of optimization problems is defined in order to determine constraints, variables, and constants of an objective function. Data-driven models for time series prediction are generated in order to support the optimization. Based on this, universally applicable optimization strategies enable the identification of energy-efficient schedules for BASs.

Proof-of-concept implementations and prototypes are realized for the evaluation of the individual contributions. Feasibility analysis and case studies are used to demonstrate the applicability and functionality as well as to discuss the benefits and open issues of the approach. Overall, this work contributes to interoperable integration of smart grid communication and BA combined with abstract semantic modeling of BEMS-related context information in order to uncouple optimization from building and technology specifics. The automated and generic design process enables the reusable application for energy management optimization in smart buildings.

Kurzfassung

Ein effizienter Betrieb ist essentiell, um den steigenden Energiebedarf im Wohn- und Zweckbau in den Griff zu bekommen. In diesem Zusammenhang bilden Gebäudeautomationssysteme (GAS) die Basis zur Steuerung von Geräten und Anwendungen in Gebäudeenergiemanagementsystemen (GEMS) hinsichtlich Komforterfüllung und Energieverbrauch. Außerdem bietet die fortschreitende Integration von Gebäuden in das Smart Grid Vorteile für ein umfassendes Energiemanagement. Die Heterogenität in der Smart Grid-Kommunikation und der Gebäudeautomation (GA), das Fehlen von maschinenlesbarer Semantik und das Anpassen von GEMS an bestimmte Gebäudetypen oder Komfortbereiche schränken jedoch die Entwicklung von flexiblen und wiederverwendbaren Energiemanagementlösungen für einen kosteneffizienten und großflächigen Einsatz ein.

Die vorliegende Arbeit zeigt den Entwurf eines GEMS auf Basis einer semantischen Abstraktionsschicht zur Entkopplung der generischen Optimierung vom Gebäudeumfeld. Web Services werden zur interoperablen Integration von GAS verwendet, während ein IP-zentrischer Protokollstack für eine vereinheitlichte Smart Grid-Kommunikation sorgt. Die relevante Information für ein GEMS wird abstrahiert und in einer Ontologie als Teil der Abstraktionsschicht zusammengeführt. Zusätzlich wird eine serviceorientierte Schnittstelle zur semantischen Maschine-zu-Maschine-Kommunikation aufbauend auf der Ontologie erarbeitet. Die maschinenlesbare Information aus der Ontologie wird zur Automatisierung der Konfiguration und des Betriebs der Optimierung im GEMS herangezogen. Ein Workflow zum Extrahieren von Optimierungsproblemen wird zur Bestimmung der Bedingungen, Variablen und Konstanten einer Zielfunktion definiert. Datengetriebene Modelle für die Vorhersage von Zeitreihen werden generiert, um die Optimierung zu unterstützen. Darauf aufbauend dient die Entwicklung von universell einsetzbaren Optimierungsstrategien der Ermittlung von energieeffizienten GAS-Ablaufplänen.

Prototypische Implementierungen werden zur Evaluierung der einzelnen Beiträge entwickelt. Machbarkeitsanalysen und Fallstudien dienen zur Demonstration der Anwendbarkeit und Funktionsweise sowie zur Diskussion der Vorteile und offenen Fragen des Ansatzes. Insgesamt trägt die interoperable Integration von Smart Grid-Kommunikation und GA zusammen mit der abstrakten semantischen Modellierung von Kontextinformation zu einer Trennung der Optimierung von Gebäude- und Technologiespezifika bei. Der automatisierte und generische Entwurfsprozess ermöglicht den wiederverwendbaren Einsatz zur Energiemanagementoptimierung in intelligenten Gebäuden.

Contents

CHAPTER 1

Introduction

1.1 Motivation

Residential and commercial buildings can be identified as third main sector of final energy consumption apart from industry and transport [1]. Coming along with growing cities and a steadily increasing world population, the buildings sector with our single-family homes, apartment houses, office buildings, hospitals, or shopping malls was responsible for approximately 31% of global final energy use in 2007 [2]. According to the International Energy Agency (IEA), this share of the buildings sector raised already to 35% in 2010 [3]. Expecting a further increase of 50% until 2050 without significant changes [3], appropriate measures and actions need to be taken in order to improve energy efficiency of buildings in all phases of the life cycle. For example, new architectural concepts as well as advancements in materials sciences are exploited in order to design and construct an energy-efficient building. In addition, building services provide an elaborate basis for comprehensive management and control during operation. For this purpose, building energy management systems (BEMSs) can be utilized to handle energy consumption while following comfort requirements of building users [4]. Basically, BEMSs focus on preferably low building energy consumption or reduced energy costs as well as a small or minimized deviation from the intended user comfort in the building. This results in a conflict of objectives that needs to be resolved with respect to the requirements of relevant stakeholders [5]. Internal and external information sources, such as adequate weather forecasts, indoor comfort requirements, or current energy prices, act as input parameters to a BEMS. Moreover, expert knowledge regarding building constraints or process behavior should be incorporated into a BEMS in order to provide a suitable basis for energy-efficient optimization in building operation.

In this context, building automation systems (BASs) are utilized in order to interact with the building environment regarding domains of interest for building users, such as visual comfort including lighting and shading or thermal comfort covering heating, ventilation, and air conditioning (HVAC) systems. Basics about building automation (BA) and room automation (RA) are standardized, for example, in VDI 3814-1 [6] and VDI 3813-1 [7], respectively. While BA sensors gather monitoring data of physical quantities as inputs from the building environment, actuators represent output interfaces of the BAS that are able to influence these physical quantities [8]. Recent advancements in the Internet of Things (IoT) [9] as well as the Semantic Web [10] boost the development of more intelligent and powerful devices including sensors, actuators, and controllers in the BA domain. The resulting smart objects act as basic modules for more complex application scenarios [11]. As a result, BASs represent a fundamental basis for successful and advanced building energy management solutions. In addition, the embedding of buildings into the emerging smart grid is promising in order to balance energy supply and energy demand with a focus on preservation of the critical power infrastructure [12]. Likewise, building energy management benefits from this smart grid integration, for example, to avoid peak demands in periods of high energy costs. The bidirectional communication with agents in the smart grid, such as energy retailers or grid operators, provides the basis for the implementation of sustainable demand side management (DSM), such as demand response (DR) programs [13].

As existing BEMSs are often tailored to specific buildings, comfort domains, or BA technologies, their reuse is limited and requires potentially high reconfiguration effort for the adaption to new settings. However, generic applicability and reusability are required with respect to the intended large-scale deployment of BEMSs in residential and commercial buildings of different size, type, or equipment in order to stem global building energy demand. According to this, a logical separation or abstraction of BEMSs from underlying internal and external systems, such as BASs or smart grids, is sought. Therefore, initialization and operation of a BEMS and its optimization task can be generalized and automated, which minimizes the necessity of manual engineering by domain experts. Standardized information and communication technology (ICT), common Web infrastructure, and Semantic Web technologies provide the necessary basis for this development. In summary, seamless integration, interoperable communication, and modeling of machine-readable semantics regarding the relevant context information will support the design of a universally applicable, smart energy management optimization in building operation.

1.2 Problem statement and hypotheses

As mentioned, utilization of installed BASs is an important factor for an energy-efficient building operation [14]. The energy savings potential of BA is the basis for standardization efforts in this field of application, such as DIN EN 15232 [15] specifying efficiency classes to categorize BA equipment and BASs. Nevertheless, the heterogeneity of existing BA technologies and standards prevents the development of a generic, flexible, and reusable energy management optimization uncoupled from particular BASs. While some technologies cover multiple BA domains, others are only focused on distinct fields of application [16]. Significant differences in communication media, data encoding, or proprietary and open communication stacks limit cross-technology interoperability. However, this would be necessary for a homogeneous, vertical integration of BASs towards applications at the management level (e.g. BEMSs) with respect to the automation hierarchy presented in [17]. Coming along with the heterogeneity, BA technologies are based on distinct information models with different levels of expressiveness. Although these models are usually clearly defined in the particular technology specifications, manual mapping effort is required to support cross-technology interoperability.

Problem statement 1. *Heterogeneous BA technologies and standards defining individual protocols and information models inhibit a seamless and interoperable integration of BASs into common management applications.*

This interoperability issue already needs to be addressed in the engineering process of BASs [18, 19]. As the focus of this work is on the building operational phase taking the installed BAS as given, a remedy for the interoperability problem can be found in service-oriented architectures (SOAs) and Web services (WSs) focusing on a seamless integration of BASs [20]. Utilization of state-of-the-art ICT and approved Web infrastructure provides a solid basis for compatibility and interoperability of BA technologies. Gateway solutions enable the consideration of legacy systems while smart objects that are already equipped with the necessary communication interfaces can be integrated directly [21]. Nevertheless, integration solutions providing WS interfaces specify individual information models, as well. Thus, model-driven methodology is used in a second step in order to ease and automate the integration process as well as to support multiple WS-based target technologies. An extensible and technology-independent meta-model is the common base for BAS modeling while automatic transformations bridge the gap to the information models of the target technologies for interoperable runtime integration of BASs.

Hypothesis 1. *WS-based SOAs provide a suitable basis for interoperable integration of BA technologies into management applications. A technology-independent, tag-based modeling of BASs as well as model-driven transformation rules support automated BAS integration into multiple WS-based integration technologies.*

A continuous integration of buildings into the smart grid offers remarkable opportunities for building energy management. Regarding the connection to smart grid agents (e.g. energy retailers, grid operators, energy aggregators) in terms of participation in DSM programs, the analysis of existing communication paradigms is necessary. According to the application domain (e.g. generation, distribution, transmission), different approaches and standards exist [22]. Protocol stacks are either based on wireless or wired physical media [23]. Moreover, the available interaction scenarios depend on the domain of the communication partner and the type of the DSM measure. In this context, relevant categories of DSM are, for example, market-oriented or physical DR programs [13].

Problem statement 2. *Different protocol stacks and communication languages are used in the smart grid domain. This leads to interoperability issues with respect to a preferably homogeneous interaction between the building and other smart grid stakeholders.*

Similar to the integration of BASs, the Internet protocol stack with the Internet Protocol (IP) as central element provides a reasonable basis for harmonized smart grid communication between residential or commercial customer buildings and other smart grid agents. Various physical layers can be supported, which is necessary to satisfy spatial or temporal communication requirements adapted to the actual situation. Focusing on DR interaction scenarios, the Energy Interoperation (EI) standard [24] published by the Organization for the Advancement of Structured Information Standards (OASIS) is used as appropriate communication language on top of the protocol stack.

Hypothesis 2. *An IP-based protocol stack enables the utilization of different physical layers in smart grid communication. On top, OASIS EI is able to cover relevant interaction patterns in a DR context between buildings and other smart grid agents.*

A successful integration of BASs and a uniform communication with smart grid agents are the first steps towards a more abstract BEMS. Moreover, flexible handling of energy consumption and comfort satisfaction requires, for example, the identification of BA appliances with variability in energy demand patterns and influences on comfort domains. Information about the embedding of the BA functionality in the building context needs to be known. As a result, a structured and semantically enriched representation of the controlled and linked systems is necessary according to the intention of a smart building energy management. Nevertheless, the strong fragmentation of technologies and the lack of machine-readable semantics about the building context, the smart grid interaction, or the user comfort requirements aggravate the design of a superordinate, abstract optimization to realize energy-efficient operation and load management for residential and commercial buildings.

Problem statement 3. *There is a lack of machine-readable semantics regarding BASs and their relations to user requirements, building structures, or external influences. Moreover, semantic modeling of smart grid communication is relevant for high-level, uncoupled building energy management due to diverse interaction scenarios.*

As the required semantics for a BEMS is not provided in a homogeneous, standardized form by the communication languages and information models in the smart grid and the BA domain, a common abstraction layer enabling semantic modeling is introduced. Ontologies known from the Semantic Web are utilized to allow for the structured representation of semantics [25]. As stated in [10], the Semantic Web idea of giving information a well-defined meaning would improve the ability of machines and devices to communicate with each other. Thus, there is not only an interoperability on the syntactic level but also regarding semantics, which is a basis for the intended high-level BEMS. This is supported by means of technologies like the Web Ontology Language (OWL) [26] or the Resource Description Framework (RDF) [27]. Interpretation of information by machines in order to infer knowledge for further processing is the desired result. Knowledge engineering and ontology-based modeling of semantics seem to gain importance in the BA design and engineering as well as operation. Related work in this field provides a basis for the elaboration of a BEMS-specific ontology that enables explicit modeling of domain expert knowledge in a machine-readable form. This eases configuration and operation of optimization in BEMSs. Moreover, the abstraction layer can be extended by a semantic machine-to-machine (M2M) communication interface based on the ontology.

Hypothesis 3. *An ontology as part of a semantic abstraction layer between BEMSs, BASs, and smart grids provides the required basis for structured modeling of semantics in the field of building energy management. The resulting representation of the building context is essential for independent and automatic processing.*

Optimization in BEMSs as a forward-looking planning task to design energy-efficient BAS schedules needs to know about the behavior of relevant building processes, such as the temperature sensitivity of certain building zones in response to set point changes. Furthermore, the estimated user behavior represents an important input for optimization and calculation of DR potential. Thus, predictions for target comfort and estimated energy supply combined with forecasts for energy demand and comfort impacts according to schedule changes are required in the optimization process. One option is the manual engineering of models for time series prediction, which requires detailed domain expert knowledge [28, 29]. Depending on the structure of the building and the number of building zones as well as the managed comfort domains and BA trades, this can be both very building-specific and expensive.

Problem statement 4. *Manual engineering of building process behavior models is an expensive task that is often very specific to a particular building and its equipment.*

BASs provide a continuously growing amount of monitoring data regarding energy consumption, indoor comfort conditions, external weather data, energy production of decentralized energy resources (DERs), or set point changes. Although manual analysis may be limited due to the size of the data set, data-driven machine learning techniques can be used to handle this bulk of data. Instead of engineering physical behavior models, the historic data that implicitly contain the characteristics of interesting processes are used to train black box models [30]. Based on the semantic abstraction layer and its ontology, automatic generation and reconfiguration of these data-driven models are supported. The modeled semantics gives information about correlations between different data sets and is the basis for adaptive time series prediction in terms of building behavior.

Hypothesis 4. *Monitoring data embedded into the semantic abstraction layer implies knowledge on building process behavior that is required for optimization in BEMSs. Neural networks for time series prediction can be automatically designed and reconfigured utilizing the context information modeled in the ontology.*

Likewise, the design of optimization problems and strategies in BEMSs often requires high manual effort and domain expert knowledge. If this knowledge is directly encoded in the resulting solution, reuse of the approach in other settings is limited. The BEMS needs to be reconfigured in case of significant changes in the building. The objective functions of the optimization algorithm covering minimization of energy procurement costs, optimization of grid stability, or maximization of consumption of locally produced renewable energy have to be adapted. Therefore, an abstract and building-independent approach is missing that solves this problem. Furthermore, building energy management is faced with a probably large solutions space, which requires heuristics in optimization in order to find solutions in reasonable time and quality [31].

Problem statement 5. *Optimization in BEMSs is often specific to particular buildings, building types, comfort domains, or equipment. This limits cost-saving reuse for deployment in other settings.*

The semantic abstraction layer to uncouple BEMSs from the underlying internal and external systems supports the conceptualization of building-independent and abstract optimization in BEMS. By means of the ontology, which hosts relevant semantics about the building context in structured, machine-readable form, both an automatic workflow to extract configuration information for the optimization problem formulation and a set of generic optimization strategies exploiting the building context can be derived. This leads to a generic BEMS design that uses the semantic abstraction layer instead of relying on manual configuration and implementation by domain experts.

Hypothesis 5. *Context information modeled in the ontology can be exploited to automatically extract optimization problems and to design abstract and generic optimization strategies for universal application in BEMSs.*

1.3 Aim of the thesis

The overall result of the thesis is the conceptualization of a BEMS for optimizing energy-efficient operation of smart buildings. This main aim can be divided into the subgoals of (1) interoperable integration of BASs and smart grid communication, (2) definition of a semantic abstraction layer, and (3) design of a universally applicable, high-level optimization approach on top of the semantic abstraction layer. Figure 1.1 shows these three blocks in a visualized overview of this thesis.

Integration. In order to interact with the environment, a BEMS needs connections to internal and external systems. Most relevant are the utilization of BASs and the consideration of smart grid communication. First, a WS-based integration of BASs is analyzed and adapted for homogeneous integration using common Web technologies. Moreover, a model-driven framework supports automatic transformation of technology-independent BA models into the information models of WS-based technologies in order to ease the integration process. Regarding smart grid communication, a suitable protocol stack is identified that provides the basis for common DR interaction between smart buildings and other smart grid agents.

Abstraction. The core of the thesis is the introduction of a semantic abstraction layer that uncouples the BEMS and takes the energy-efficient optimization to a higher, more abstract level. An ontology generalizes the modeling of typically incompatible BA technologies and the representation of smart grid interaction. The focus is on the requirements of BEMSs considering energy consumption patterns, scheduled tasks, user comfort requirements, energy market information, or smart grid load information. Based on Semantic Web technologies, this generic layer links the superior BEMS with the

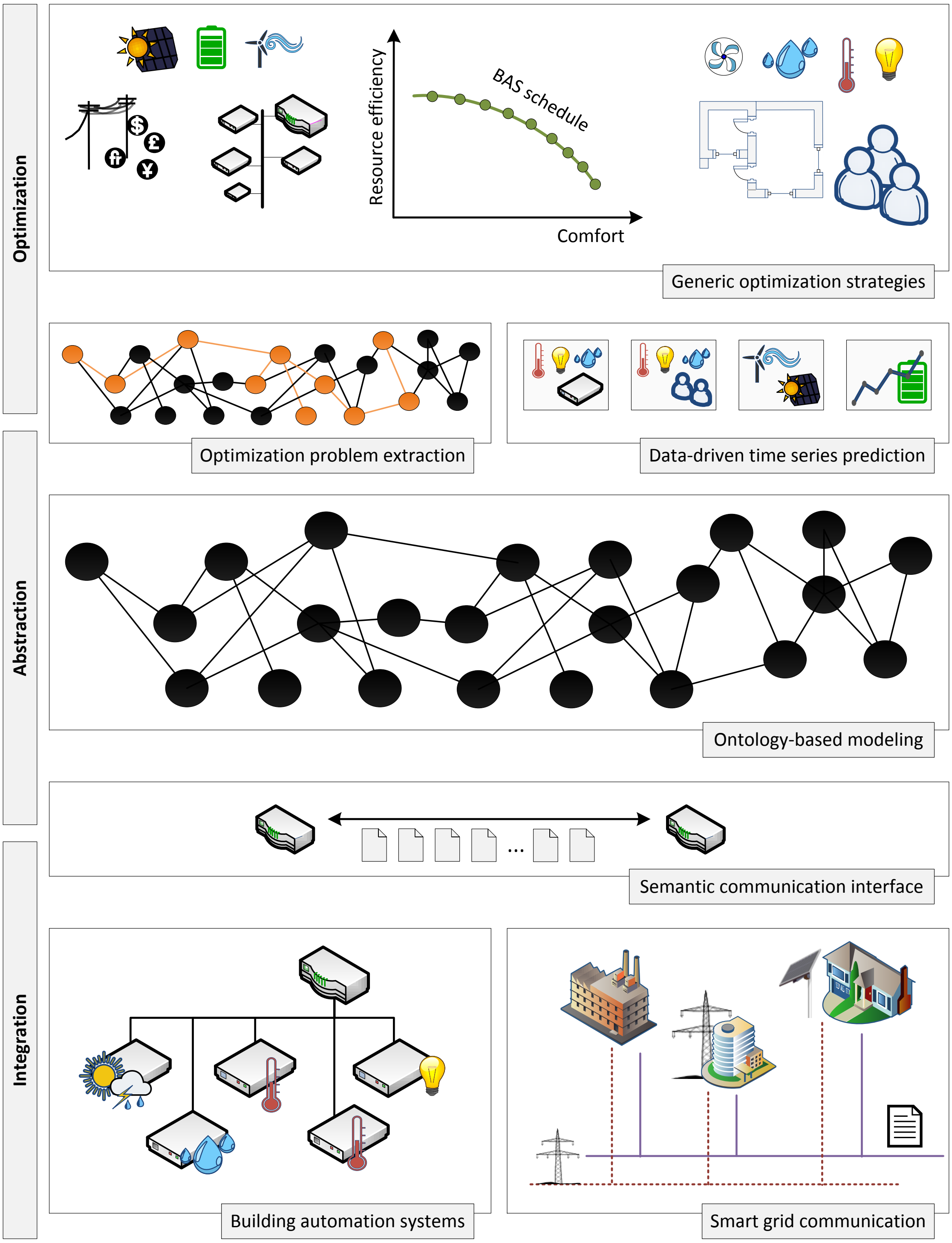

Figure 1.1: Thesis overview

internal BASs and the external smart grid. Cross-domain information and data exchange on top of shared semantics are realized by means of a semantic M2M communication interface. In general, the semantic abstraction layer introduces a linguistic basis for machine-readable abstraction independent of technology-specific details. The Semantic Web approach should be continued by linking the developed ontology with other ontologies and reusing approved modeling concepts.

Optimization. The semantic abstraction layer provides the basis for generic optimization in BEMSs. First, an extraction procedure for automated creation and initialization of an energy management optimization problem is defined. Utilizing the ontology, an extraction path can be identified to instantiate the decision variables, control variables, constants, and constraints. In order to support the actual optimization with data forecasts for the particular optimization period, a data-driven prediction framework is introduced. With respect to actual and target comfort as well as energy supply and energy demand, models based on neural networks are derived and reconfigured by means of the ontology. On top, context-aware strategies for optimization in BEMSs are formulated. Both resource efficiency and comfort are considered by these strategies that are embedded into common metaheuristics. They are characterized by a generic design, which enables reuse and universal application in conformance with the focus of this thesis.

In summary, the goal is not to specify a completely new optimization algorithm for building energy management but to define a common and comprehensive basis in the form of a semantic abstraction layer in order to abstract from approved optimization approaches with respect to universal and generic application independent of particular buildings, comfort domains, or BA trades. The configuration and operation of optimization in BEMSs are simplified due to the semantic abstraction layer and the interoperable integration of relevant systems, which lowers costs and effort for individual deployments.

1.4 Methodology

In this section, the methodology to test the stated hypotheses is outlined. First, the requirements for the intended BEMS design based on the semantic abstraction layer are acquired. A state-of-the-art survey is conducted to identify relevant related work in this field. Information modeling methodology is used to support integration of heterogeneous BASs. Analysis of relevant technologies and protocol stacks helps with the identification of a suitable smart grid communication stack. Moreover, ontology engineering methodology is applied in order to define the basis for the semantic abstraction layer. The semantic M2M interface is developed by means of a systematic technology selection and an interface specification. Algorithm design techniques are used for the prediction framework, the optimization problem extraction, and the heuristic optimization strategies. The results lead to proof-of-concept implementations that are evaluated by means of feasibility analysis and case studies.

State-of-the-art survey and requirements engineering. A state-of-the-art survey for BAS integration, smart grid communication, and optimization in BEMSs is conducted. Moreover, related work for semantic modeling regarding BA, smart grids, building information modeling (BIM), and user comfort requirements is investigated. Recent standards and technologies in the fields of BA, DR, and the Semantic Web are considered in this analysis phase. Most relevant existing approaches are studied in order to identify suitable modeling patterns and technologies for further reuse. Prior to the definition of a semantic abstraction layer, an analysis regarding the requirements for energy management optimization in buildings is performed. Additionally, the interoperability requirements concerning the heterogeneous BA technologies and the smart grid communication with respect to the intended semantic M2M interface are collected. The requirements engineering is accompanied by a survey of interesting use cases for building energy management.

Information modeling and ontology engineering methodology. Information models of established BA technologies and standards are reviewed to identify common intersections. Technology-independent type definitions are derived that can be used in system integration, which is realized by means of WS-based gateway technologies. Moreover, model-driven methodology is applied to specify a meta-model supporting the technology-independent description of BASs by means of tags. According to the model-driven paradigm, model transformations are introduced to support an automated and eased integration process. Communication languages in the smart grid context are analyzed with respect to their capability of mapping information of different interaction scenarios. The models and meta-models identified in the state-of-the-art survey provide the basis for the ontology engineering process. Literature is searched for suitable ontology engineering methodologies [32, 33, 34]. Ontology Development 101 [33] is chosen due to the clear, simple, and modular structure. However, principles of other approaches are considered, as well. As available models described by means of common Semantic Web standards are preferred, model transformations are not required for reuse during the definition of the proposed ontology. The querying of information and the inference of new knowledge are realized according to knowledge engineering methods, for example, to automatically instantiate optimization problems.

Interface specification. The defined ontology is the basis for sharing knowledge between communicating systems in the proposed BEMS concept. The specification of an interface based on Semantic Web technologies for high-level semantic M2M communication, such as between a BAS and a BEMS, takes into account existing M2M approaches. A technology selection process identifies and assesses appropriate communication protocols that utilize IP and support the elaborated requirements. The study of relevant services in the field of BEMSs leads to the determination of individual service categories summarized in a SOA.

Algorithm design techniques. Algorithm design methodology is used in the development of the adaptive time series prediction framework, the optimization problem extraction, and the definition of the context-aware optimization strategies. The prediction framework introduces a procedure for generating necessary forecast models. Neural networks as learning-based techniques are used for the data forecasting task. A heuristic approach is developed enabling continuous evaluation and improvement of the instantiated models. The algorithmic realization of the optimization problem generation workflow follows the identified extraction path in order to map modeled semantics of the ontology to data structures of the runtime environment. Moreover, the universally applicable optimization strategies incorporate proven algorithm design paradigms like divide and conquer to tackle the energy management optimization problem in a heuristic way. Furthermore, the strategies are embedded into common metaheuristics that are identified in a previous literature review and state-of-the-art analysis.

Proof-of-concept implementations. The concepts, algorithms, and models developed in the individual publications that are combined in this thesis use proof-of-concept implementations as a basis for further evaluation. The integration of BASs using BACnet Web services (BACnet/WS) is tested with a prototype realization of a BACnet/WS server. The elaborated smart grid communication stack is examined in a proof-of-concept implementation of a demo grid setting. The models and transformation rules for the technology-independent modeling framework for BASs are analyzed in the course of the KNX Web services (KNX WS) specification. The ontology for smart control of BA and modeling of smart grid interaction is realized using Semantic Web standards, such as OWL. A Java-based proof-of-concept implementation of the semantic abstraction layer adopts the semantic interface, incorporates the OWL ontology, and shows the automatic extraction and formulation of optimization problems. Furthermore, technology connectors to integrate subsystems are developed in this prototype. The prediction framework is implemented in MATLAB using the Neural Network Toolbox. For the generic optimization, a prototype of the described strategies that are embedded into the variable neighborhood descent (VND) metaheuristic [35] is developed in a Java application.

Feasibility analysis and case studies. The evaluation of the proposed contributions is performed using the developed prototypes and proof-of-concept implementations. Feasibility analysis shows the applicability and functionality of the integration part of the thesis including the semantic M2M interface. Common interaction patterns and sequences are tested to state their feasibility. The modeling approaches and the optimization problem extraction are evaluated using case studies and use cases identified during the literature review. The building simulation tool EnergyPlus is employed to provide the necessary data for the evaluation of the prediction framework's performance. Case studies are defined with the intention to analyze the behavior and the problem solving capability of the optimization strategies. The results are finally discussed covering the benefits and the open issues of the approaches and their proof-of-concept implementations.

1.5 Synopsis

Within this cumulative dissertation, a compilation of peer-reviewed publications is presented in Chapters 2 to 10. For the sake of consistency, layout alignments are implemented in order to maintain readability and referencing. Examples are the introduction of sequential section numbers, the adaption of tables and figures regarding sizes or positions, a consistent keyword coloring in the listings, or the unification and refinement of acronyms and literature references. Nevertheless, the contents of the chapters are identical to the underlying original publications. In this synopsis, the basic data, the own contributions, and the original abstracts are listed to present an overview of the publications. This section is followed by a comprehensive summary in Section 1.6 that links the compiled publications in the context of this thesis. Further details beyond this summary can be found within the individual publications.

Chapter 2 deals with interoperable communication in the smart grid domain as smart buildings are key players for the emerging smart power grids. The chapter presents the work "Smart grid communication at the interface of customer buildings with focus on demand response" that was authored by Daniel Schachinger, Stefan Gaida, and Wolfgang Kastner and published at the *International Symposium on Smart Electric Distribution Systems and Technologies* in September 2015 [36]. The own contribution to this work is the development and discussion of the communication system consisting of an IP-centric protocol stack and a common language in accordance with general smart grid requirements. Preliminary work regarding the selection of an appropriate communication language is done by Stefan Gaida. The proof-of-concept implementation to evaluate the underlying work is developed together with the co-authors.

Abstract [36]: Traditional power grids lack an appropriate infrastructure to link the involved stakeholders and domains for balancing energy demand and supply. The transmission infrastructure is hierarchically oriented with active bulk generators and passive consumers. Therefore, a bidirectional communication system is needed, which is an essential component of the future smart grid. However, a set of requirements and challenges has to be addressed in order to realize the intended communication infrastructure. In this work, a multi-agent system architecture is presented that tackles these requirements. With a focus on customer buildings and demand response communication patterns, an interoperable and scalable as well as standardized system is defined, which uses the Internet Protocol as central element in the communication stack. OASIS Energy Interoperation standard is used as agent communication language for data exchange between smart grid stakeholders. Furthermore, a proof-of-concept implementation is realized to illustrate the functional capability of the presented approach.

Chapter 3 shows an approach for the integration of BASs based on BACnet/WS. This chapter presents the work "Interoperable integration of building automation systems using RESTful BACnet Web services" that was authored by Daniel Schachinger, Christoph Stampfel, and Wolfgang Kastner and published at the *Annual Conference of the IEEE Industrial Electronics Society* in November 2015 [37]. The own contribution is the conceptualization and refinement of reusable BAS type definitions on top of the formalized BACnet/WS object model. This contribution is based on preliminary work in the context of a BACnet/WS server reference implementation realized by Christoph Stampfel.

Abstract [37]: Building automation domain is affected by a diversity of standards and technologies. With the upcoming Internet of Things heading for a pervasive network of interconnected smart things and the need for efficient and intelligent energy management systems, the necessity of integrating these heterogeneous building automation environments soars. Thus, standardized, interoperable, secure, and scalable solutions are required. Well-established Web service technologies based on the Internet Protocol act as key enabler to realize this vision. In this work, an approach for the seamless and interoperable integration of building automation systems based on RESTful BACnet/WS is presented. In order to ease the integration process, the BACnet/WS specification is described as formal, machine-readable object model. Additionally, most common building blocks of building automation systems including logical as well as physical resources are specified in the form of type definitions to unify integration. Furthermore, a proof-of-concept implementation of a BACnet/WS server is realized in order to illustrate the functional capability of the presented approach.

Chapter 4 introduces a modeling framework that aims at automating and generalizing the integration process of BASs into WS-based integration technologies for runtime communication. The chapter presents the work "Modeling framework for IoT integration of building automation systems" that was authored by Daniel Schachinger, Andreas Fernbach, and Wolfgang Kastner and published in *at - Automatisierungstechnik* (vol. 65, no. 9) in September 2017 [38]. The own main contribution is the definition of the tag-based modeling and transformation concept including the common meta-model, the core vocabulary, and the generic transformation schema. The identified requirements for interoperable BAS integration are used as basis for this development. The technology-specific transformation rules for OPC Unified Architecture (OPC UA) are provided by Andreas Fernbach while Open Building Information Exchange (OBIX) and BACnet/WS rules are part of the own contribution, as well.

Abstract [38]: Advancements within the Internet of Things are leading to a pervasive integration of different domains including also building automation systems. As a result, device functionality becomes available to a wide range of applications and users outside of the building automation domain. In this context, Web services are identified as suitable solution for machine-to-machine communication. However, a major requirement to provide necessary interoperability is the consideration of underlying semantics. Thus, this work presents a universal framework for tag-based semantic modeling and seamless integration of building automation systems via Web service-based technologies. Using the example of the KNX Web services specification, the applicability of this approach is pointed out.

Chapter 5 addresses semantic modeling of smart grid interaction patterns for utilization in BEMSs. This chapter presents the work "Ontology-based abstraction layer for smart grid interaction in building energy management systems" that was authored by Daniel Schachinger, Wolfgang Kastner, and Stefan Gaida and published at the *IEEE International Energy Conference* in April 2016 [39]. The own contribution covers the development of the ontology using the selected engineering methodology. Preceding modeling work conducted in collaboration with Stefan Gaida provides the basis for this contribution. The evaluation is mainly based on the proof-of-concept implementation and the demo grid setting already used in Chapter 2.

Abstract [39]: Replacing traditional power grids by future smart grids opens manifold opportunities for energy-efficient operation of buildings and cities as well as improved coordination of energy demand and supply. Current information and communication technology provides a suitable basis for the bidirectional flow of information between buildings and other smart grid stakeholders. However, a common notion of shared knowledge is essential in order to unify heterogeneous grid environments, incorporate information of smart grid participants, and process this information in building energy management. In this work, an abstraction layer based on an OWL ontology is presented that enables semantic representation of knowledge for interaction between building energy management systems and smart grids. A well-proven methodology is used to develop this ontology. Furthermore, the ontology application into building energy management systems and smart grid environments is illustrated, and the functional capabilities of this approach are shown.

Chapter 6 targets semantic modeling of BASs and their relations to buildings, building users, or external influences with focus on high-level, smart control. The chapter presents the work "Semantics for smart control of building automation" that was authored by Daniel Schachinger and Wolfgang Kastner and published at the *IEEE International Symposium on Industrial Electronics* in June 2016 [40]. The own contribution is the specification of an abstract ontology for smart control in the BA domain. Related work is reused in order to elaborate the ontology's core components to describe building structure, devices, data services, and control services. An additional contribution is the definition of a basic interface in order to access the ontology representing the central information source as part of a semantic abstraction layer.

Abstract [40]: Building automation is an important part of state-of-the-art building management in order to attain most efficient operation in accordance with comfort requirements, energy consumption, or budget allowance. For this purpose, current building management systems enable communication with subjacent systems at the field and automation level by definition of mostly syntactical technology mappings. However, integration of building automation systems for management and control purposes also needs to address the semantics of these subsystems, their cooperation, and their interference. In this work, such an integration approach is presented that enables smart control of building automation resources by the use of semantic technologies. An OWL ontology is developed in order to represent and link knowledge of all relevant domains. Furthermore, an interface concept for seamless and interoperable cross-border communication in the heterogeneous building automation environment is introduced. Finally, an application scenario illustrates the functional capabilities of this approach for smart control in building management.

Chapter 7 introduces an interface for M2M communication reusing existing standards of the Semantic Web. The chapter presents the work "Semantic interface for machine-to-machine communication in building automation" that was authored by Daniel Schachinger and Wolfgang Kastner and published at the *IEEE International Workshop on Factory Communication Systems* in May 2017 [41]. The own contributions are the analysis of requirements for a semantic M2M communication, the definition of the interface, and the feasibility study to evaluate the proposed concept. Distinct service categories are identified, and requirements for the underlying semantic modeling are elaborated. The selection of an appropriate communication protocol forms the basis for the specification of the service set for M2M communication.

Abstract [41]: Current trends and advancements in the Internet of Things and the Semantic Web have already found their way into the domain of building automation. As machine-to-machine communication and integration of heterogeneous building automation technologies are of increasing importance, interoperability is a necessary precondition. In order to support building automation communication, a customized set of services needs to be available. Additionally, semantics of exchanged information has to be described in a machine-readable way to enable automatic interpretation of message contents. In this work, an interface based on Web technologies and Semantic Web standards is presented, which supports platform-independent machine-to-machine communication for building automation. A requirements analysis for such an interface leads to the definition of a service-oriented architecture. The semantics of exchanged message contents is described in an ontology that provides the basis for a common understanding. Moreover, feasibility and hardware requirements of the proposed approach are evaluated.

Chapter 8 deals with the automatic extraction and formulation of optimization problems based on context information. The chapter presents the work "Ontology-based generation of optimization problems for building energy management" that was authored by Daniel Schachinger and Wolfgang Kastner and published at the *IEEE International Conference on Emerging Technologies and Factory Automation* in September 2017 [42]. Based on previous work, the automatic workflow for extraction and configuration of optimization problems in the context of BEMSs represents the own contribution. This includes the ontology-based extraction and the actual formulation of variables, constants, and constraints in order to initialize a generic objective function. An additional contribution is the case study that shows the feasibility of the approach.

Abstract [42]: In general, a trade-off between comfort satisfaction and minimization of energy consumption or overall costs needs to be found by building energy management systems. Additionally, the design of energy management strategies often requires high effort and expert knowledge in order to model the dynamics within a building, which leads to very specific solutions with limited reuse. Thus, this work presents an approach for the automatic generation of optimization problems for building energy management based on machine-readable semantics. For this purpose, an ontology hosts all relevant information necessary for the optimization problem formulation. Information extraction and transformation into the optimization problem domain are addressed. Moreover, a case study demonstrates the functionality of the proposed procedure.

Chapter 9 shows an adaptive prediction solution for target and actual cross-domain comfort as well as energy supply and demand. This chapter presents the work "Adaptive learning-based time series prediction framework for building energy management" that was authored by Daniel Schachinger, Jürgen Pannosch, and Wolfgang Kastner and published at the *IEEE International Conference on Industrial Electronics for Sustainable Energy Systems* in January 2018 [43]. In collaboration with Jürgen Pannosch, learning-based models for the prediction of time series to support BEMSs are analyzed. Based on the analysis results, the own contribution is the definition of the adaptive prediction framework using neural networks, which includes the methods for model identification, generation, evaluation, and reconfiguration.

Abstract [43]: Sustainable building energy management is inevitable in order to reduce global energy demand. For this purpose, building energy management systems need to know the expected behavior of building automation systems, energy production units, or thermal dynamics. Designing the underlying models by domain experts might be a complex and expensive task. However, the models are already inherent in the growing amount of available monitoring data. Thus, this work proposes a framework utilizing learning-based modeling for the prediction of relevant time series in order to support comfort satisfaction and resource efficiency in building energy management. A set of neural networks is generated and trained using monitoring data and building context information modeled in an ontology. Autonomous and building-independent application is achieved by continuous performance evaluation and conditional adaption of the neural networks. The evaluation presents exemplary results and discusses the major findings.

Chapter 10 describes the design of universally applicable optimization strategies that exploit contextual information of an ontology. The chapter presents the work "Context-aware optimization strategies for universal application in smart building energy management" that was authored by Daniel Schachinger and Wolfgang Kastner and published at the *IEEE International Conference on Industrial Informatics* in July 2018 [44]. The own contribution covers the development of the generic optimization strategies that are abstracted from common requirements regarding optimization in BEMSs. These strategies are used for subproblem identification, partial schedule modification, and impact assessment. Moreover, the combination with metaheuristics and the case study-based evaluation mark own contributions to this work.

Abstract [44]: In building operation, the continuous forward planning of energy-efficient schedules to maintain user comfort is a challenging task. Although the design of building energy management systems is an active field of research, existing solutions are often faced with limited reusability due to specialization on certain buildings, comfort parameters, or building automation technologies. Thus, this work introduces a set of context-aware strategies that are generally applicable for the optimization in building energy management systems. For this purpose, machine-readable semantics of the building and the building automation system is exploited in order to design a heuristic approach. The aim is to reduce the optimization effort while targeting both energy efficiency and cross-domain comfort satisfaction on a building-independent level. An embedding of the proposed approach into common metaheuristics is described to provide a basis for further reuse. Finally, case studies are used for evaluation of a proof-of-concept implementation.

1.6 Summary of this work

This thesis presents a systematic approach to design an energy management optimization for operation of smart buildings based on a semantic abstraction layer. First, basic interoperability and integration of smart grid communication and BA are summarized in Section 1.6.1. Second, semantically enriched modeling of context information relevant for BEMSs is part of Section 1.6.2. Third, the methods to exploit this machine-readable semantics for optimization problem and forecast model generation as well as development of universally applicable optimization strategies are introduced in Section 1.6.3. The structure of this summary that outlines the publications of Chapters 2 to 10 corresponds to the stated objectives of this thesis that are visualized in Figure 1.1. In order to introduce and link the contributions of the underlying publications in terms of a coherent overview, this summary also discusses the state of the art already mentioned in the publications as well as additional literature beyond that.

1.6.1 Integration

A BEMS requires connections to the automation systems installed in a building and the energy systems linked to a building in order to master the task of energy-efficient building operation. Sensors and actuators of BASs supply the BEMS with monitoring data and implement control commands to influence appliances in the building. On the other hand, a connection to the smart grid enables the participation in energy-related conversations with external smart grid stakeholders. However, the heterogeneity in existing BA technologies as well as in smart grid communication standards needs to be tackled as already highlighted in Problem statement 1 and Problem statement 2 of Section 1.2. Thus, interoperable integration and communication for both the smart grid and the BA domain are analyzed with the intention to define a more abstract and homogeneous base for uniform data and information exchange in the BEMS context.

Smart grid communication

In general, the smart grid is intended to be the next evolution step of the traditional power grid and is characterized by a bidirectional flow of information and energy [45]. Figure 1.2 sketches a generic smart grid setting with the two distinct infrastructures for exchanging information and exchanging energy. Based on the two-way communication, DSM strategies like DR programs can be used to establish a more consumption-oriented balancing of energy demand and supply [13]. In this context, DR refers to the process

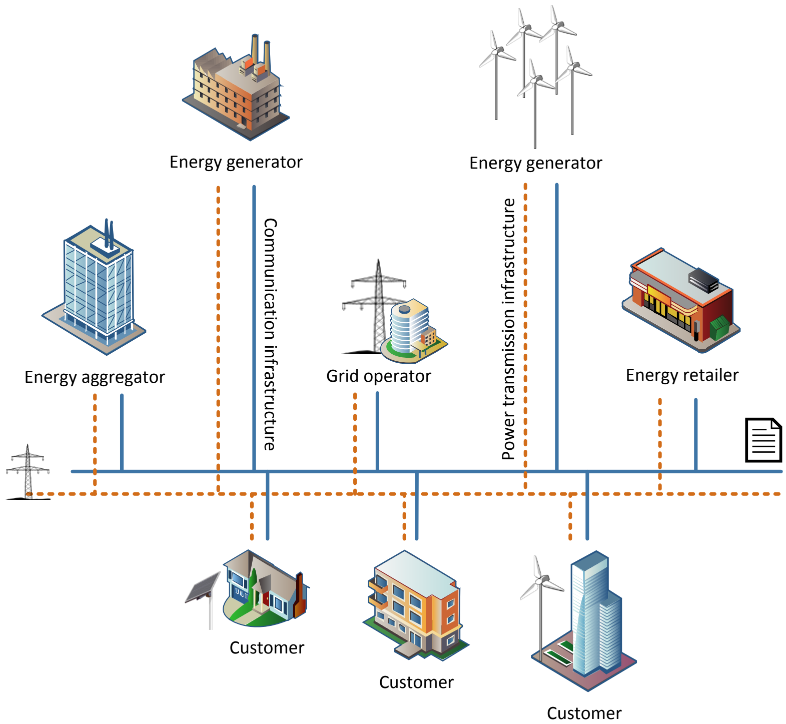

Figure 1.2: Smart grid with power transmission and communication infrastructure [36]

of managing energy consumption by the customer according to price changes or other incentives and conditions in the smart grid [12]. A comparison of current DR implementations is presented in [46]. However, cross-domain interaction in the smart grid requires standardized communication systems to support the contemplated benefits [47]. Thus, European respectively US organizations and institutions have already attended the smart grid topic resulting in a series of studies and international standardization efforts like the EU Mandate M/441 on smart metering in Europe [48]. Besides standardization, literature states several other requirements for smart grid technologies. In [49], interoperability for end-to-end communication, performance regarding network bandwidth or processing power, scalability in number of nodes or geographical distribution, security, and efficiency for installation and network management are considered as technical requirements for ICTs. In addition, system reliability, robustness, and availability as well as Quality of Service (QoS) measures are identified [23, 45, 47]. Based on these criteria, we define a smart grid communication system including an interoperable communication stack and a common communication language with a focus on DR interaction between smart buildings and other smart grid agents in Chapter 2 [36], which is summarized in the following paragraphs.

For the definition of an overall communication architecture, the multi-agent system (MAS) paradigm is reused as it fits the challenges in the smart grid domain [50]. In this context, an agent is characterized by autonomy having control over its internal states, social ability to communicate with other agents, reactivity to the environment, and pro-activeness regarding flexibility in problem-solving [51]. Focusing on the interaction of customer buildings with other smart grid agents, distinct categories of relevant smart grid agents can be identified as shown in Figure 1.2. Energy generators represent abstract bulk generation independent of the utilized primary energy source. Grid operators cover the functionality of power transmission and distribution grids. Energy retailers and energy aggregators are market agents aiming at coordination of energy demand and supply. Residential or commercial customers consume energy from the grid but can also feed in locally produced energy. The relevant interaction scenarios that are addressed in our work are based on common use cases presented in [52]:

- Energy retailers interact with customers by publishing energy price information, such as real-time pricing (RTP) or time-of-use (TOU) pricing, and collecting energy consumption forecasts for future pricing.
- Grid operators use physical DR in the form of regulatory commands to force customers increasing or decreasing their energy consumption in case of emergencies.
- On the other hand, grid operators also buy flexibilities from customers as market-oriented DR measure in order to be able to balance upturns and downturns in the power supply.
- Interaction between energy aggregators and customers aims at trading flexibilities and surplus energy of local production for reselling on the energy market.

The basis for a homogeneous communication between the smart grid agents, which follows these four standard scenarios for market and physical DR, is the selection of an appropriate protocol stack utilizing existing ICTs. With respect to the application domain and the spatial conditions, both wireless and wired technologies can be utilized on the physical layer [23]. As existing infrastructure should be reused as much as possible in order to reduce, for example, installation costs, technologies used in common Internet infrastructure provide a suitable basis. Wired options that can be used in local area networks (LANs) and wide area networks (WANs) are Ethernet, digital subscriber line (DSL), or powerline communication (PLC) [23, 49]. On the other hand, wireless alternatives for short distances in local networks are IEEE 802.11 or IEEE 802.15.4-based ZigBee [49]. In sparsely populated areas, cellular networks represent promising solutions to span long distances. Regarding the requirements, the most suitable medium should be selected for the particular application domain. Nevertheless, the main aim of the proposed communication system is that the end-to-end communication between the grid agents on the application level is independent of the underlying physical medium. This can be ensured by utilization of IP on the network layer of the protocol stack. On the one hand, IP is a de facto standard for information technology (IT) in general and the Internet in particular by now [53]. On the other hand, IP is a scalable technology fulfilling QoS criteria and other relevant requirements for smart grid communication [54]. Both the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) can be used as transport layer protocols on top of IP [49]. As reliability is a requirement for smart grid information and data exchange, the connection-oriented TCP fits better for usual smart grid communication scenarios. A set of standardized protocols and technologies is available for the application layer on top of IP and TCP/UDP. In the smart grid domain, the Extensible Messaging and Presence Protocol (XMPP) [55] or the Session Initiation Protocol (SIP) [56] can be applied besides the popular Hypertext Transfer Protocol (HTTP). In parallel, encoding of sent and received messages is required on the application layer. An agreement is important for interoperability between the grid agents. Examples are the Extensible Markup Language (XML), the JavaScript Object Notation (JSON), or the Efficient XML Interchange (EXI) [49].

In order to enable interoperable communication between smart grid agents using the identified, IP-centric protocol stack in the defined MAS, the selection of a common and standardized agent communication language (ACL) is essential [51]. With respect to the relevant interaction scenarios described before, the ACL has to support several communication mechanisms:

- The ACL needs to provide a function for the distribution of price information. This can be used by grid operators or energy retailers to influence customer demand.
- Future or past energy consumption reports are communicated between smart grid agents. This helps grid operators to plan infrastructural investments, for example.
- For the initialization of communication links, some kind of registration or enrollment must be provided by the ACL.
- In order to trade flexibilities, a tendering mechanism with different phases (e.g. solicitation, offering, ordering) has to be supported by the ACL.
- Finally, an event mechanism should inform smart grid agents about grid stability problems, power fluctuations, or blackouts.

Three technologies for the use as ACL are compared in our work. The Smart Energy Profile (SEP) version 2.0 [57] is analyzed but not selected as it mainly aims at communication within buildings although most of the required patterns are supported [49]. Moreover, Open Automated Demand Response (OpenADR) version 2.0 [58], which is a subset of OASIS EI version 1.0 [24], is considered. Finally, OASIS EI is chosen as it is more powerful compared to OpenADR and offers mechanisms to realize the discussed communication patterns. The quote service, the transactive service, the enroll service, and the event service are used for price distribution, energy consumption information publication and energy tendering, registration, and notifications, respectively.

In a proof-of-concept implementation, we explore the feasibility of the communication system to handle interaction of customer buildings with other smart grid agents in the context of DR patterns. IEEE 802.11 is used as network technology while TCP and XMPP are selected on top of IP. Operational domains, such as local neighborhoods with customers and grid operators as XMPP clients, are grouped together and assigned to distinct XMPP servers, which are connected in a mesh topology. The interoperable end-to-end communication is evaluated with OASIS EI using XML encoding. In this setting, the wireless technology is chosen to avoid additional wiring and keep installation costs low. Nevertheless, more dependable and robust network technologies and media should be used for critical grid segments and interaction types. By means of the proposed system, the agents are able to interact on an abstract level without any knowledge about the internal communication of particular subsystems in the domain of an agent. Thus, heterogeneity, for example, between the customer domain and the transmission operator domain is hidden, which leads to an interoperable basis for high-level DR applications. Scalability is already addressed by IP and XMPP [54, 55], but increased data transfer due to growing smart grids requires compact data representations. Finally, sustainable and efficient security concepts need to be considered due to the long life cycle of power grids although these are out of scope of this work [23]. In summary, we contribute an IP-centric protocol stack and a common ACL for smart grid communication mapped to a MAS. This offers the basis for exchanging information between a BEMS that is behind the smart grid interface of a customer building and the relevant smart grid stakeholders including energy retailer and grid operator as counterparts in DR interaction scenarios.

Building automation systems

While smart grid communication connects the building management applications with the external grid infrastructure, the link to the building and the building users is realized by the BASs. A wide range of technologies and standards in the BA domain can be utilized in order to increase comfort for building users, reduce energy consumption or costs, and implement security and safety measures [59, 60]. In general, BASs are organized in a three-tiered system architecture covering field layer, automation layer, and management layer [17]. Within this structure, multi-vendor BA resources, such as sensors, actuators, or controllers, are dedicated to various domains like climate control, visual comfort, safety, or security [61]. As a result, the available technologies and standards, such as BACnet [62], KNX [63], or LonWorks [64], are arranged in a heterogeneous landscape. Although most of the BA technologies provide typical abstraction mechanisms based on datapoints connecting functional blocks, they have their own protocols and information models limiting the interoperability between them. However, there is a growing need to interconnect individual BASs with each other inside a building and with BASs and BA resources of other buildings in the context of developments in the IoT [9] and the smart cities [65]. In a first step, we explore the seamless and interoperable integration of BASs utilizing WS technologies in Chapter 3 [37], which is summarized in the following paragraphs.

In principle, interoperable integration of BASs has to address both a uniform communication interface and a general representation of data and information [16]. The problem of interoperable bindings already needs to be targeted in the design and engineering of BASs [18]. The task of appropriate device selection, which is often done manually by system integrators, can be automated by means of greedy [66] or evolutionary algorithms [18, 67] based on a repository of device descriptions. Moreover, interoperability of engineering tools is discussed in [68]. In building operation, a basis to solve the interoperability issues is provided by SOAs that are used to build interoperable but autonomous systems [69]. In a SOA, the implementation of services is hidden behind the exposed interface. A popular implementation of SOAs is provided by platform-agnostic WSs [69]. Besides utilization in common IT systems and enterprise applications, WSs are also relevant for the integration of BASs due to the increased focus on IP technologies in the BA domain [70]. WSs can follow the Representational State Transfer (REST) paradigm or the WS-* architecture. WS-* fits business integration better while the more lightweight REST with its resource-orientation, the small set of services, and the better scalability is more suitable for BAS integration [71]. In order to integrate already existing, non-IP devices and systems in the BA domain, WS-based technologies emerged that can also be utilized as gateways in order to hide the BA technology specifics behind an abstract WS interface. Examples are OPC UA [72], OBIX [73], or BACnet/WS [74] that can be used to integrate (legacy) systems into an IoT-enabled environment. Moreover, these technologies define information models to describe BASs independent of the underlying technologies, which is necessary for abstract interoperability [75].

By introducing a high-level WS interface as well as an abstract object model, the BACnet/WS standard forms a suitable solution for interoperable integration and uniform communication with management applications. In our work, we use the BACnet/WS specification of the proposed Addendum am to ANSI/ASHRAE standard 135-2012 [76], which is integrated into the standard by now [74]. In contrast to the previous BACnet/WS version published in the Addendum c to ANSI/ASHRAE standard 135-2004 [77] that is based on the Simple Object Access Protocol (SOAP), the new version follows the REST paradigm. On top of an HTTP binding as application layer protocol, a set of basic and advanced services is specified. In addition to the elementary read, write, invoke, create, and delete operations, functionality to filter items, read logs, or initialize subscriptions is provided. Data encoding is achieved by means of XML, JSON, or plain text. For secure communication, the use of Transport Layer Security (TLS) is required.

The BA resources that are accessible via the proposed WS interface need to be modeled in a technology-independent way. For this purpose, the BACnet/WS object model is not as generic as the OBIX object model but provides additional standardized classes to describe BASs in more detail. On the other hand, the object model is not as complex as in OPC UA, which eases the BAS integration by means of BACnet/WS. As the BACnet/WS standard describes the object model only in textual form, we contribute a machine-readable description of this model as basis for automatic processing and interaction. The result is a meta-model for the instantiation of actual system models. Basically, the modeling approach in BACnet/WS differentiates between data to hold process value information and metadata to describe properties of the data:

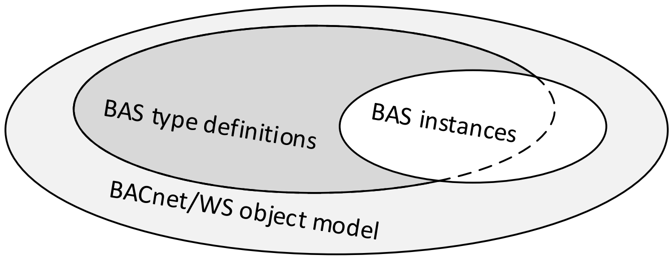

Figure 1.3: Modeling domains for BAS integration [37]

- Data are modeled by means of data items that form the classes in the resulting machine-readable meta-model. Primitive data items for integer, time, or string values directly hold the actual value. On the other hand, constructed data items, such as complex objects, structs, or collections, are containers for other primitive or constructed data items. Both are inherited from the abstract data item class and can be used for individual modeling of BA structures and resources. In addition to the distinct data items necessary to represent a particular BAS, the BACnet/WS specification provides several standardized data items as default entry points for type definitions, subscriptions, server information, or authorization information.
- Metadata, on the other hand, hold semantic information of data items. In the derived meta-model, metadata are realized as attributes of data item classes or in the form of associations describing relationships between these classes. In RESTful environments, uniform resource identifiers (URIs) are used to specify unique paths for accessing modeled instances. For this purpose, BACnet/WS uses the identification metadata *name*. In order to define dependencies and other relationships between data items, the metadata *extends*, *implements*, or *returns* are utilized, for example. Besides metadata to model multilingual texts, customized tags can also be defined to describe additional characteristics.

Although the meta-model representing the standardized BACnet/WS object model already enables a generic modeling of systems and resources, an intermediate layer introducing special type definitions for common building blocks regarding BAS-specific modeling is defined. This layer is visualized in Figure 1.3 between the BACnet/WS object model and the BAS instances. The analysis of relevant BA technologies leads to the identification of common core components. Similar to [78], they are aggregated and abstracted resulting in three main types covering physical, functional, and organizational dimensions. The derived type definitions can be used to support interpretation of information in the BA context at runtime as well as to ease the configuration in terms of BAS modeling:

- The device type is used to describe the physical BA resources (e.g. sensors, actuators, controllers). Product-specific information (e.g. manufacturer, order number) as well as installation-specific information (e.g. address) is covered by this type definition. Moreover, the device type contains the functional endpoints of the actual device (i.e. datapoints). The type is defined as customized object data item with associated children as property nodes.

- The datapoint type is instantiated to model the functionality of a BAS. For this purpose, an abstract type is provided that can be extended to define atomic functional endpoints like temperature values of a sensor. Child elements of this abstract datapoint type contain additional information (e.g. datapoint priority).
- The view type enables the arrangement of datapoints and devices into logical or organizational structures. For example, the physical topology or the building structure can be specified by means of instances of this view type. The devices and datapoints can be linked to these instances within the topology or building structure (e.g. rooms). The view type allows self-referencing in order to describe hierarchies and nested views.

In a proof-of-concept implementation, these type definitions are published in the definition context of the BACnet/WS server. Thus, client applications are able to read these types and understand their inherent semantics in order to process the integrated BAS models. It has to be noted that the BAS models in terms of BACnet/WS do not aim at representing internal BAS conditions, such as connections of datapoints forming complex BA functions, but the integration prepares access points to input and output functionality of BA resources for utilization in external applications and systems. Figure 1.4 visualizes the cooperation of client applications (e.g. BEMSs) that try to access resources in the BAS remotely via the BACnet/WS server. The client requests are processed by the RESTful WS interface. Then, the encoded message content is transformed into the internal representation according to the basic BACnet/WS object model and the advanced type definitions. The internal services interact with specific technology adapters in order to execute the requested functionality.

The evaluation of the BACnet/WS-based integration of BASs for the purpose of interoperable and technology-independent communication with BEMSs uses this prototypical server implementation. Regarding interoperability on the communication and information level, the BACnet/WS approach satisfies the requirements due to the utilization of WSs on top of approved IP communication. The machine-readable meta-model provides the syntactic and semantic basis for BA-specific modeling independent of a certain technology. Configuration effort, which needs to be taken only once, mainly depends on the size of the BAS in terms of the number of resources. Runtime performance can vary according to the used communication protocol, the server implementation, and the message encoding format. As BACnet/WS is limited to HTTP, only different message representations can be considered to reduce, for example, message sizes [20]. The functionality is evaluated by executing a set of client requests on a KNX test bed behind the BACnet/WS server instance.

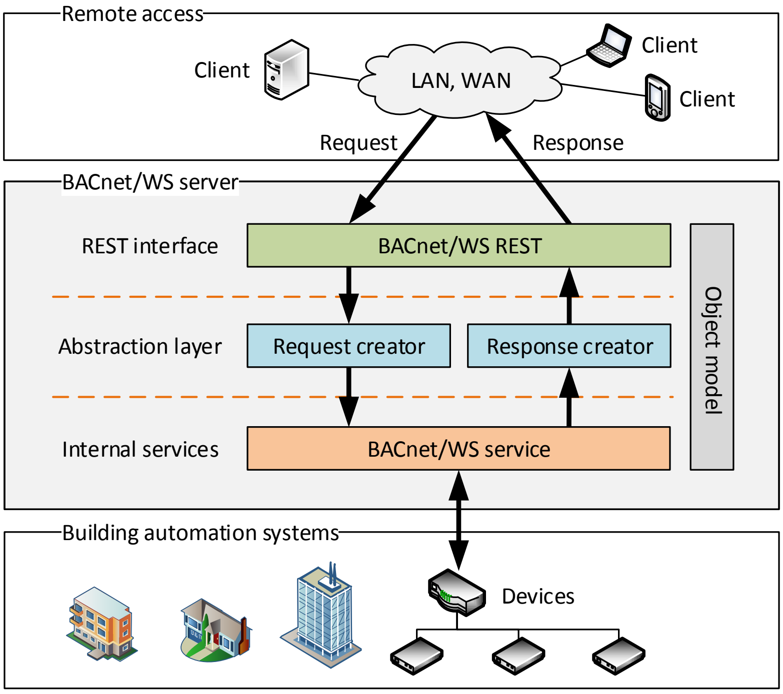

Figure 1.4: BAS integration based on the BACnet/WS proof-of-concept architecture [37]

The results of the BACnet/WS-based integration show the suitability of WSs as basis for interoperability and uniform integration of BASs. As a consequence, they can be used as enabler for IoT integration and eased access by management applications. However, the process of modeling the BA structure, the resources, or the functionality can be an extensive manual task although this needs to be performed only once. Hence, an automatic workflow is necessary in order to tackle this manual modeling effort and to provide a reusable integration procedure. The utilization of generic modeling is also an integral part of automated design and engineering approaches for BASs. For example, a top level topology model is introduced as common basis for automatic processing and model transformations in [79]. According to the Model-Driven Engineering (MDE) methodology [80], we develop a framework for abstract, semantically enriched, and technology-independent modeling as well as transformations towards popular WS-based integration technologies in Chapter 4 [38]. This work, which is summarized in the following paragraphs, describes an automatic and seamless integration workflow for accessing BASs at runtime that is independent of underlying BA technologies and utilized WS-based target solutions.

Following the IoT visions of Atzori et al., a key factor for interoperability is the consideration of semantics while the things-oriented and Internet-oriented aspects are covered per se due to the focus on BASs that are incorporated into an IP-enabled environment [9]. Thus, the modeling approach needs to be able to represent semantics regarding the BA resources. Although ontologies become an established method to realize semantic modeling, we chose a more light-weight and intuitive way in the form of tag-based modeling extending the Project Haystack approach [81]. This allows different user groups (e.g. BA manufacturers, facility managers) an easy system description by means of simple semantic annotations. Moreover, manufacturers can use this method to specify product libraries that can be imported by system integrators while engineering a BAS.

The developed method combines the two major principles of MDE, modeling and model transformation [80]. The modeling is embedded into a layered hierarchy described by the Model-Driven Architecture (MDA), the MDE variant of the Object Management Group (OMG) [82]. In this context, the lowest level is the system that needs to be represented (e.g. BAS). On the next level, a system model is defined, which conforms to a meta-model specifying a common modeling language. In order to make meta-models comparable, they conform to a common meta-meta-model on top. The aim of this work is to automate the process of making data and information about a BAS available to external clients by means of a WS interface. Thus, a generic modeling concept is introduced in order to abstract from both the diverse BA technologies and the particular WS integration solutions with their various information models. For this purpose, a meta-model is defined that enables the instantiation of a system model as well as the formulation of a tag vocabulary, which is used in a system model to describe the semantics. This vertical relationship is called meta-modeling [82]. On the other hand, the relation between the tag vocabulary and the system model, which are both on the same hierarchy level, is known as meta-programming [82].

The main class of the common meta-model to describe system models is the *Entity*. On the other hand, the class *Tag* is the core concept for creating vocabularies for semantic annotation. Entities are containers for one or multiple *Feature* instances that link a tag and a corresponding value in order to characterize an entity. Tags are subdivided into *Basic* tags, *Reference* tags, and *Marker* tags. Simple data types (e.g. *string*, *real*) are combined in the *type* enumeration with the intention to attribute basic tags. Marker tags are used to show the membership of an entity to a specific concept in the BA domain (e.g. *device*). Reference tags are used for relationships between entities. For example, the marker tag *view* is used to identify an entity as a BAS view while the reference tag *viewRef* can link an entity with another entity that is already marked as view. For a comprehensive tag vocabulary, the meta-model provides the concept of tag compositions. Thus, the formation of a virtual class as fusion of individual tags to describe the complex characteristic of concepts in the BA domain can be realized. In this context, also inheritance of such compositions is enabled. The meta-model is encoded as XML schema in order to support several visual and textual editors in the process of system modeling.

Besides the meta-model, a core vocabulary is defined in the developed modeling framework, which covers the most important tags for BAS modeling as basis for further extension due to future needs. In addition to general tags for identification (*id*, *name*) and multilingual description (*description*, *locale*, *translation*), four major blocks of tags can be identified in this core vocabulary:

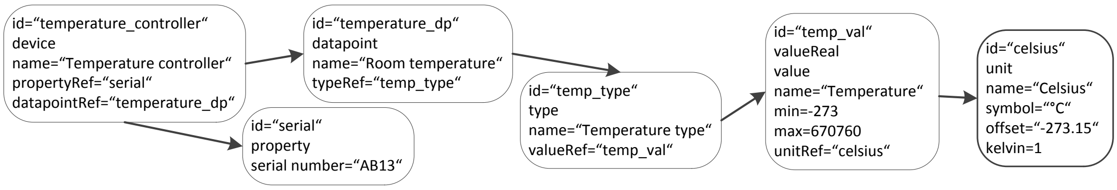

Figure 1.5: Tag-based modeling example [38]

1. Devices are the physical elements of a BAS that are indicated by the marker tag *device*. Devices can have a list of properties (*property*), such as serial number, dimensions, supply voltage, or access flags. Moreover, they host a set of datapoints (*datapoint*) for control inputs and monitoring outputs.

2. Arrangements in the form of topologies (*topology*), functional trades (*functionality*), or building structures (*building part*) are inherited from the *view* concept.

3. The basics to model meta-information, such as units for data values (*unit*) or custom enumerations (*enumeration*), are part of the core vocabulary, as well. Unit modeling makes use of the elementary SI unit system. Enumerations are realized with literals and corresponding binary or numeric values.

4. Atomic datapoints or other functional blocks rely on a specific *type*. Thus, the fourth block introduces tags for modeling basic values (*value*), such as temperatures, timestamps, or binary states, in order to describe these functional types.

Figure 1.5 depicts a simple example how the tag vocabulary can be used to form valid models that conform to the common meta-model. The root element is a BA device (*temperature_controller*) with an additional property (*serial*) and one datapoint (*temperature_dp*). Furthermore, the datapoint has a distinct type (*temp_type*) with an associated value (*temp_val*) that specifies its semantics. The blocks describe entities while each row in an entity is a feature. Features with marker tags do not need a value. The value of reference tags represents the identifier of referred entities, which is also visualized by arrows. Actual process data are not part of such models as they are only intended to integrate the static information of BASs into WS interfaces at configuration time providing the context for subsequent runtime communication.

According to the model-driven paradigm, transformation rules are defined in order to map the tag-based models in the scope of the technology-independent modeling framework to the information models of the common WS-based integration technologies OPC UA [72], OBIX [73], and BACnet/WS [74]. Management applications as clients of gateway components and BA resources that implement these technologies are able to access BASs via the provided interfaces without having specific knowledge of the underlying BA communication. As the tag vocabulary is extensible, the transformation rules need to be dynamic and flexible. Therefore, the rules can also be applied on new

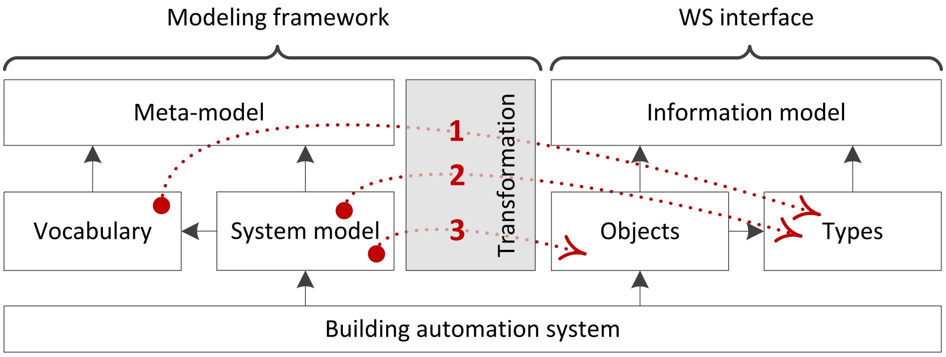

Figure 1.6: Transformation process for model-driven BAS integration [38]

tags that rely on those specified in the core vocabulary. Figure 1.6 illustrates the generic transformation schema consisting of three rule sets:

- Rule set 1 analyzes the tags and the tag compositions of the vocabulary to create individual types in the WS interface. For this purpose, the typing concepts of the WS-based integration technologies are utilized.
- Rule set 2 is based on the system model and transforms the specified types for datapoints or functional blocks to types in the WS interface domain. Existing default concepts are reused as far as possible.
- Rule set 3 maps all not yet considered entities of the system model to objects in the WS interface. Thus, this rule set addresses the actual BAS characteristics while rule sets 1 and 2 are focused on the transformation of meta-information.

These rule sets need to be adapted to the supported target technologies in order to consider specifics like predefined data types or specialized object types. The rules for OBIX and BACnet/WS are quite similar as their object models follow the same basic concepts. While OBIX uses a so-called contracting mechanism to define new classes, BACnet/WS offers the definition context for inheritable types. Both offer several classes to model simple data as well as collections and constructed objects. The unit definition in OBIX is built on the SI system, but BACnet/WS relies on the BACnet Engineering Units. On the contrary, transformation rules for OPC UA take into account the elaborated and comprehensive libraries for information modeling. The semantics provided by standard concepts should be reused, but the mapping rules should be kept as simple as possible to provide maintainability.

Our modeling framework with its generic, tag-based modeling and the transformations of technology-independent BAS representations to the information models of WS-based integration technologies is an integral part of the KNX WS specification. In the KNX standard [63] as representative in the home and building automation domain, current integration techniques require specific knowledge about the KNX protocol. Therefore, the KNX WS specification makes use of WSs to uncouple KNX networks from applications in common IP-based systems. In the KNX WS specification, a slightly different meta-model and additional tags in the core vocabulary are used but the basic principle is the same. Information about the BAS is extracted from the Engineering Tool Software (ETS) and extended by additional information leading to the KNX Information model as system representation based on the tag vocabulary. This model is integrated into the KNX WS gateway by applying the transformation rules regarding the selected WS interface. At runtime, clients can access the KNX network that is hidden behind the gateway. Thus, the KNX WS specification is a good example for the feasibility and applicability of the model-driven BAS integration based on WSs and common IP technology.

In order to apply the automatic integration workflow to the previous work of manual BAS integration using BACnet/WS [36], the BAS types on top of the BACnet/WS object model are translated to compositions using tags of the technology-independent vocabulary. Execution of the transformation rules for BACnet/WS will reproduce these manually defined BAS types. However, the developed model-driven integration method enables a wider field of application due to the support of multiple WS-based integration technologies. In summary, the modeling of BASs benefits from the easily manageable, tag-based modeling framework that automates the integration process. As a result, BASs become accessible by applications like the intended generic BEMS without the need for additional knowledge about the underlying technology, its protocol, or device specifics.

1.6.2 Abstraction

After the integration of heterogeneous BASs using WS-based solutions in order to abstract from technology-specific communication as well as the elaboration of an IP-based stack for homogeneous smart grid interaction between different stakeholders, the next step towards the intended generic BEMS design is the introduction of a separation between the high-level BEMS and the underlying BA and smart grid systems. Regarding an abstract information representation and exchange, there is a need for modeling semantics in a structured, machine-readable form that covers context information about BASs and their relations to the building and the environment. Additionally, semantics about relevant smart grid domains and smart grid interaction is required to finally build a basis for smart, abstract, and automatic processing in uncoupled management applications. This necessity is already highlighted in Problem statement 3 of Section 1.2. Therefore, the concept of a semantic abstraction layer is formulated, which is called abstraction layer in [39] and semantic layer in [40]. This intermediate tier combines a model for the description of semantics regarding the fields of interest in building energy management and an interface specification to access and manage this model. The aim is to design the foundation for a central knowledge base that can be used by manifold systems and subsystems in the BEMS context. The semantic abstraction layer forms a concentration point for the exchange of data, information, and knowledge [40]. Depending on the intended system architecture, the semantic abstraction layer is established either as central, distinct hardware or software entity that separates the systems linked in a star topology or as logical uncoupling layer in the point-to-point communication of two systems. Thus, connected systems use the same common notion of shared knowledge for interpretation of data and information. In the following paragraphs, our work regarding an ontology for semantic modeling and an interface for semantic communication is summarized.

Ontology-based modeling

In general, the modeling of information and knowledge focusing on the smart grid interaction and the BAS context can either be realized by a relational database or an ontology [25]. The former has become a common methodology for information storage and querying especially for specific applications or organizations. In the course of Semantic Web developments, ontologies, which are rather new in computer science compared to databases, become a popular method for semantic modeling [25]. Although ontologies are usually not that efficient regarding data management, they are preferred in the context of this work due to their benefits regarding sharing and linking of distributed knowledge, knowledge reuse, logic inference, implementation independence, high level of abstraction, and availability of standardized technologies that support structured and formalized definition of semantics [25, 83]. By means of the machine-readable representation of semantics, the models should be interpretable for both humans and machines while ambiguities caused by the use of natural language for human-readable semantic descriptions are eliminated [49]. In Chapter 5 [39], we elaborate on an ontology that addresses the representation of semantics regarding relevant concepts in smart grid interaction. Furthermore, we model an ontology for smart control in the BA domain in Chapter 6 [40]. Both publications are summarized in this section to give an overview of the ontology-based modeling as part of the semantic abstraction layer.

Related work and available literature indicate that ontologies are a popular method for standardized and structured modeling of semantics nowadays. The relevance of Semantic Web technologies and semantic models for interoperability in the smart grid is addressed in [84]. Standards like OWL [26], RDF [27], or RDF Schema (RDFS) [85] allow for the formal representation of a certain knowledge domain [84]. Regarding semantic modeling in the smart grid, a basic ontology and data model is provided by the Common Information Model (CIM) [86, 87]. Rohjans et al. introduce semantic WSs utilizing this CIM in combination with OPC UA [88]. The use of ontologies for decision support in the domain of distribution service operators is discussed in [89]. Moreover, dynamic DR motivates the development of a semantic smart grid information model where relevant domains, such as real-time consumption, infrastructure information, customer behavior, schedule information, and natural conditions, are integrated in a set of component ontologies using the OWL standard [90]. Although general grid characteristics can be modeled, the approach is more focused on basic building events instead of aggregated smart grid interaction patterns.

On the other hand, there are a lot of approaches using ontologies to describe semantics in the building domain (e.g. construction, commissioning, operation). In this context, the SAREF ontology for smart appliances presents a device-centric approach [91]. Device functionality is modeled by means of functions for sensing, metering, or actuating. Moreover, energy production and energy consumption can be described in terms of energy profiles. The result is a basis for semantic interoperability in the heterogeneous building environment. In [92], controllable and uncontrollable building things (e.g. appliances, furniture, plants) are modeled in the context of the building environment (e.g. rooms). Moreover, the resulting DogOnt ontology enables functionality, state, and network modeling. Based on DogOnt, the ThinkHome ontology describes a combination of component ontologies used to model process information, energy-related information, buildings, resources, or weather influences regarding the smart home domain [93]. Thus, BA resources and their functionality can be linked with the building context as well as the energy providers to cover the required consumption. The BOnSAI ontology defines concepts regarding hardware, functionality, services, or QoS [94]. For this purpose, the ontologies OWL-S and CoDAMoS are integrated. In order to support advanced and flexible energy management, a facility data model is defined in [95]. Based on a core ontology, abstract and physical entities can be modeled including data types, communication protocols, policies, plants, or topology.

The design and commissioning phase in the BA life cycle is supported by ontology-based device descriptions presented in [96]. A layered ontology architecture combines a domain-specific vocabulary (layer 1), predefined platform-specific data (layer 2), platform- and manufacturer-specific types (layer 3), and platform- and manufacturer-specific device descriptions. As a result, devices are comparable, and characteristics of hardware and software in the BA domain can be modeled. Originating from the projects HESMOS and SCUBA, the BASont ontology utilizes the device description ontology of [96] as basis for technology-independent BAS modeling [97]. The work is intended to cover relevant use cases in a BAS life cycle including design, commissioning, operation, and refurbishment. Thus, the ontology is used, for example, to plan device deployments, provide information to maintenance staff, or run fault detection and diagnosis. In [98], a tool chain for a systematic engineering of BASs based on BASont is presented. A knowledge-based engineering approach for automation systems combining OWL and AutomationML is introduced in [99]. Moreover, a structured representation of requirements for BAS engineering that are defined in an OWL ontology is presented in [100]. The explicit modeling of BAS control logic is intended by the CTRLont ontology that enables the representation of UML state machines, VDI 3814-6 state graphs, and schedules [101]. Additionally, the ontology provides concepts to cover conditional logic that is required, for example, to specify state transitions.

In the context of BIM, a mapping of the Industry Foundation Classes (IFCs) [102], which are used to represent construction data, to the ifcOWL ontology is discussed in [103] and [104]. Together with the Semantic Sensor Network (SSN) ontology and the SimModel ontology, the ifcOWL ontology is used in the performance assessment ontology aiming at building energy management [105]. Performance metrics and performance objectives are key concepts in this ontology. Another approach targeting BIM in combination with building management systems (BMSs) is presented in [106]. Knowledge engineering by means of ontologies can also be used in fault detection and diagnosis of BASs and BMSs as proposed in [107]. The introduced domain specific ontology makes use of the ifcOWL to integrate BIM data. In [108], an approach to detect faulty building behavior is presented. Non-semantic BAS descriptions are mapped to an extended SSN model that enables the derivation of physical models. Based on this, diagnosis rules can be automatically generated to analyze the building behavior by means of monitored sensor data. The Brick schema tackles the heterogeneity of available building representation approaches [109]. With a focus on commercial buildings, an ontology of tags and tagsets is defined in order to support the comprehensive semantic modeling of building infrastructures. This is the basis for smart building applications in general and building energy applications in particular. A work exploiting the potential of ontology reasoning to reduce building energy use is proposed in [110].

Grassi et al. introduce an ontology framework that connects and extends existing domain ontologies in order to build a basis for energy management in smart homes [111]. Concepts of DogOnt, BOnSAI, or ThinkHome are combined to describe energy, service, device, user, location, and building aspects. Due to the growing number of ontologies in the architecture, construction, engineering, and facility management domain, alignments to the central Building Topology Ontology (BOT) are proposed and evaluated in [112]. The work analyzes the domain ontologies SAREF, ifcOWL, Brick, DogOnt, and ThinkHome. In [113], a linked data approach to share building data in the cloud is presented that aims at a holistic management of buildings.

As can be seen, specialized and general-purpose ontologies are widely used in research nowadays. Some concepts of this related work are actually suitable for reuse in the ontology-based modeling of this thesis. For this purpose, the utilized Semantic Web technologies to describe the ontologies provide standardized mechanisms to link ontologies in order to avoid isolated developments. Thus, the ontology of this work is focused on the core requirements regarding energy management in buildings concentrated on smart management of available BASs and communicating with smart grid agents from a customer's perspective. The ontology is built from scratch in order to remain clear and tight. However, related concepts of other ontologies are linked to form a basis for shared understanding of a certain knowledge domain as defined in [34]. In this thesis, the definition of the ontology models depends on the engineering methodology Ontology Development 101 [33]. A procedure that provides the modeling of terms, classes, properties, facets, and instances of the ontology and considers reuse of existing work is specified forming a simple and clear structure. In addition, principles of alternative methodologies to master the iterative process of ontology development are considered [32, 34]. Although none of the ontologies identified in the state-of-the-art analysis is directly imported, suitable individual concepts are linked following the alignment mechanisms offered by the OWL standard in order to enable interoperability of knowledge.

Based on the IP-centric communication stack elaborated in [36], the active participation of a flexible and smart BEMS in DR programs requires consideration of several information domains on a homogeneous level. Static and dynamic characteristics of the smart grid and its stakeholders are needed, such as hierarchical grid structures or provided services. An ontological basis to model interaction in DSM or DR scenarios, which are already discussed in [36], is important for a preferably autonomous processing in applications like BEMSs. The scope of the resulting ontology presented in [39] covers four major building blocks that are visualized in Figure 1.7 and discussed in the following paragraphs:

- First, the actors or stakeholders in the smart grid (i.e. agents) need to be modeled. For this purpose, ontological concepts to specify their identity, location, operating area, or functionality are created. Interesting stakeholders are energy retailers, energy aggregators, energy generators, grid operators covering transmission and distribution grids, and customer agents.
- Agents provide and require services at the interface to the smart grid, which is shown in Figure 1.7 using UML notation. Based on these service definitions, interaction scenarios can be composed that express the sequence of service executions and data exchange between involved stakeholders. This machine-readable representation of communication patterns is an important feature for autonomous and automatic processing of smart grid interaction.
- The communication in the smart grid is based on common ICT. In order to check compatibility of services and agents, a shared knowledge about the supported communication technologies is necessary. Concepts to model relevant configuration parameters, protocols, and ACLs are provided.
- Besides communication-related aspects of the smart grid, also an adequate representation of the power transmission infrastructure is important. This includes spatial arrangements of smart grid stakeholders as well as the hierarchical structure of grid segments. On the other hand, the smart grid communication infrastructure using existing ICT infrastructure is not explicitly modeled in the ontology.

A top-down approach is chosen for the definition of a class hierarchy with the root elements of *Agent*, *Service*, *Node*, *Segment*, *Technology*, *Protocol*, *Language*, and *Parameter*. Below, specialized classes, for example, to differ between the mentioned grid agents are created. On the service level, there is a distinction between offered and required services that are able to send or receive data characterized by parameters. The hierarchy of grid segment classes is subdivided using geographical, physical, and functional aspects. For identification purposes, the ontology introduces data properties like a name and an identifier. The location of agents and grid nodes is supported by data and object properties to describe the geographical coordinates as well as the address. The vertical and horizontal composition of grid structures is based on object properties to express hierarchy and adjacency. Moreover, agents are connected to grid nodes and cover a certain operating area. The services are offered and required by agents while the exchanged values are characterized by parameter configurations linked to these services.

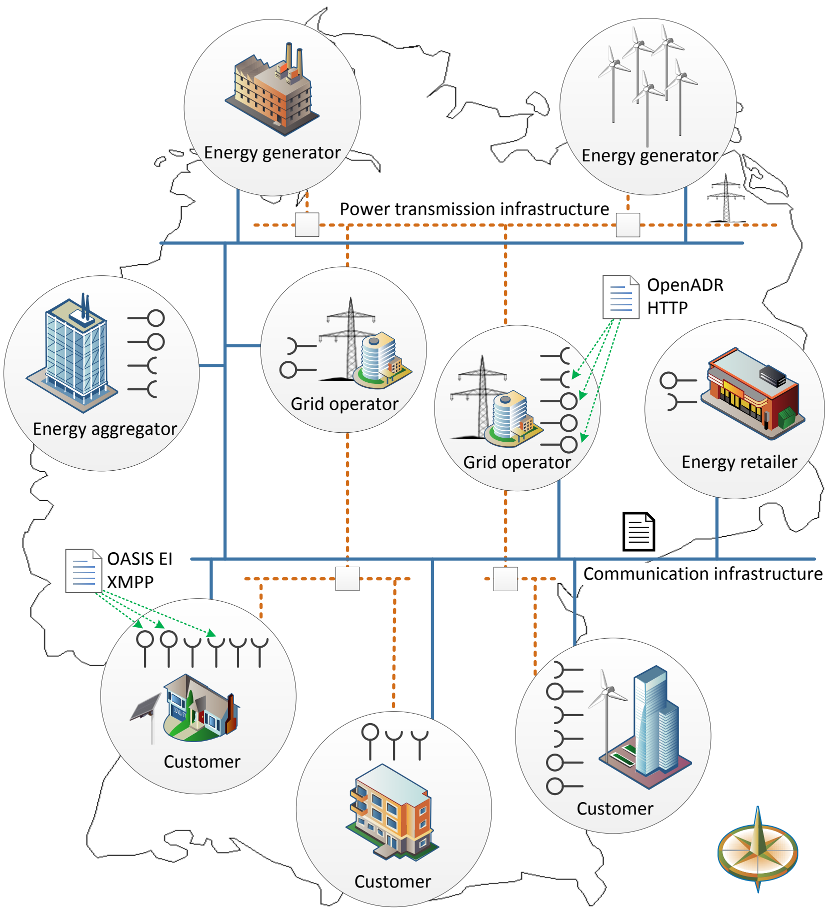

Figure 1.7: Scope of the smart grid interaction ontology [39]

The composition of services in order to form interaction patterns is realized by means of an object property indicating logical dependencies. The communication technology combines the used protocol and the supported language. Services are associated with communication technologies via object properties. In general, integer, float, and string types are used for the specified data properties. On the other hand, the object properties describe the domain and the range. Necessary and sufficient conditions are added to the classes in order to determine their scope, which leads to a set of primitive and defined classes.

The aforementioned context information about the smart grid agents, their services, the interaction patterns, or the grid structure is formalized in an abstract but machine-readable OWL ontology, which is the basis for advanced applications that obtain additional knowledge for dynamic and flexible management. There are either local knowledge bases that cover the ontology and the system representations per agent or central servers that host the knowledge base for a set of agents. Ontologies or knowledge bases can be linked while applications do not have to take care about the distribution of information.

A proof-of-concept implementation is used to evaluate the modeling capabilities and the support of smart grid interaction of our ontology design with respect to use cases based on the test bed introduced in [36]. First, the grid situation shown in Figure 1.7 is represented using the ontology in order to show the semantic modeling capabilities. Then, the reasoner is run to infer new knowledge and check the consistency of the system model. For example, compatibility of services according to the supported communication technologies of the particular agents can be inferred using rules in the ontology. This helps management applications to select appropriate communication partners. Finally, mockup implementations deployed on Raspberry PIs are realized for the agents to simulate DR interaction patterns. For example, the distribution of day-ahead energy prices by energy retailers followed by the publication of energy consumption forecasts by customer buildings is executed. Authentication and authorization are out of scope of this work, but this needs to be considered in future real-world deployment due to criticality of the power infrastructure.

In parallel to the semantically enriched description of smart grid interaction to support BEMSs, the semantic modeling of BASs and their relations to buildings, building users, or external influences with focus on smart control is targeted in the ontology of the semantic abstraction layer [40]. The consideration of machine-readable semantics takes the interoperability between technologies in the heterogeneous BA environment to the next level. This is an important factor for autonomous and smart management and control of BAS functionality that is embedded into a building environment. The underlying BASs can use the ontology for semantically enriched communication, and the BEMS as management application on top can be implemented independently of the utilized technologies and systems in the building and its environment. In order to abstract from the technology-specific characteristics, the developed ontology is mainly based on the notion of services for representing BA sensing and actuating functionality similar to [39]. These services are used to interact with the building environment by sensing environmental (comfort) parameters, such as temperature or brightness, and changing the states of these parameters by means of actuating tasks. The assignment of logical services to the physical devices and to the building is also addressed in the ontology, which results in a semantic modeling language that supports smart BA management and control [40]:

- Building structure concepts are used to describe hierarchical building topologies as well as arrangements of adjacent building elements. The main concept for hierarchies is the *Zone*. Based on a zone or its subclasses (e.g. room, floor, building part, site), arbitrary compositions can be defined. For this purpose, invertible object properties are used to express the relationships between superordinate and subordinate zones. Object and data properties for identification and location are reused from the smart grid interaction ontology [39]. Moreover, similar and equivalent classes of other ontologies (e.g. ifcOWL, ThinkHome) are aligned to enhance interoperability. For zone arrangements, object properties are introduced to define relative structures of zones (e.g. *isRightOf*, *isBehind*, *isAbove*). Additionally, the absolute orientation of zone delimiters (e.g. *Wall*) can be modeled. Furthermore, the ontology contains properties to describe the nature of inner and outer zone delimiters in more detail (e.g. *transparency*).

- Devices and appliances are integrated into the building structure in two ways. First, they are installed somewhere in a building. Second, controllable devices and appliances have an operating area according to the provided services, which is not necessarily congruent with the installation place. The abstract concept of a *BuildingResource* is used as root class to model both controllable and uncontrollable resources. In this context, a building resource can act as a container for services to bridge the gap between the BA functionality and the building context. Again, alignment is used to enable a more detailed description of building resources by means of other ontologies (e.g. DogOnt, SSN, BASont).

- Data services are primarily used to describe monitoring data and their semantics (*DataService*). These services are made available by service providers like sensor devices. Data services use the parameter configuration concept to specify the semantics of the provided data values. Each parameter configuration combines two parameters. For example, a temperature sensor provides a data service that offers temperature values per instants of time. Thus, the parameter configuration's parameters are temperature and time, which enable an automatic interpretation of associated data values. Additional context information is given by the operating area of the data service (e.g. building zone). Similar to [39], the BA technology of services can be modeled as technology connector.

- Control services represent the actuating functions that are able to influence comfort parameters in building zones (*ControlService*). Thus, they have an active role in controlling the BA resources. Similar to data services, control services are provided by building resources and have a particular operating area. Moreover, they can be concatenated if one control service triggers other services. A control service hosts a set of ordered states that can be set in the underlying, physical BA resource. Based on these states, control services describe parameter variations, which contain the semantics behind the available actuating functions. The elements of a parameter variation are the associated parameter, the variation trend, and the relative state value change. The variation trend determines the direction of the parameter value change (i.e. up, down) while the relative state change is geared to the ordered list of states. Additional conditions can constrain the execution of parameter variations.

The energy-related aspects of control services and devices are added to this ontology in subsequent work that targets the optimization based on the semantic abstraction layer. These extensions are supported by the extensible design of the ontology. In [41], the control service concept is supplemented by the definition of energy consumption profiles in order to model the energy demand of a control service's states. This is further extended in [42], where the energy service concept is introduced to model local and external energy

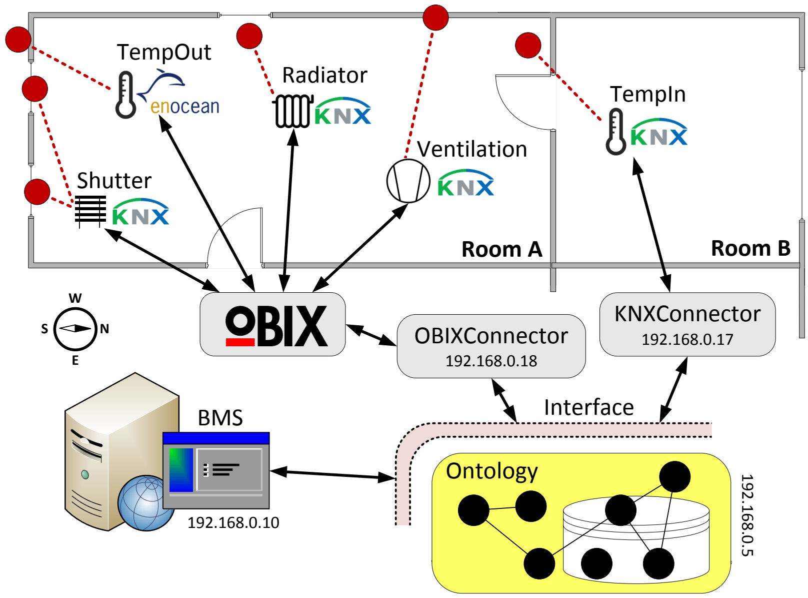

Figure 1.8: Application scenario for the smart control ontology [40]

supply functionality. Energy providers and energy consumers are linked via energy types. In [44], energy services are used to describe both energy demand and energy supply similar to data and control services.

Exemplary modeling of test cases is applied in order to evaluate the feasibility of the ontological approach to provide enough context information for sophisticated decision making in smart BA control. A simple application scenario is illustrated in Figure 1.8, where all parts of the BA control ontology's scope are outlined. The adjacent rooms separated by walls represent a cutout of the building structure. Sensing and actuating devices are installed in the rooms. In this example, their respective operating area is directly derived from the installation place. Each device provides control or data services to interact with the building and the comfort parameters perceived by building users. This setting is modeled in the ontology of the semantic abstraction layer. Here, the semantic abstraction layer is realized as distinct hardware entity that links the management application and the available technology connectors mapping the BA technologies.

Based on this modeling, the management application in Figure 1.8 uses the technology-independent description of the building structure, the devices, the control services, and the data services to interpret monitoring data and decide on set point changes in order to satisfy comfort requirements. For this purpose, simple test cases are executed to validate our ontology model. It has to be noted that high-level decision making does not replace low-level control tasks in the field and automation level. Reasoning can be used to infer new knowledge for the BMS or BEMS, as well. Finally, the developments for semantic modeling of BA control are merged with the smart grid interaction ontology to build a central base for data, information, and knowledge in the semantic abstraction layer necessary for an independent BEMS. In summary, the smart control ontology and especially the combination with the smart grid interaction ontology provide a suitable basis for abstract and flexible optimization in BEMSs uncoupled from underlying systems in the BA and smart grid domain.

Semantic communication interface

The developed ontology that introduces semantic modeling of smart grid interaction and smart control of BA qualifies for a semantics-based communication. Thus, we develop a uniform M2M interface in order to ease sharing and distribution of knowledge between communication partners, such as BASs, BMSs, or BEMSs. Both vertical and horizontal system integration are supported by this interface, which makes use of the formalized ontological concepts. Semantic Web technologies can be utilized to represent the exchanged semantic information or to encode queries for information requests over the interface. In order to reuse existing Web infrastructure for communication, the interface is based on common technologies and standards of the Internet protocol suite. In the following paragraphs, the work of Chapter 6 [40] and Chapter 7 [41] on such a semantic communication interface is summarized.

In general, M2M communication has to address architectural needs concerning scalability, security, reliability, latency, power consumption, or mobility [114, 115]. Reusing technologies of the Internet protocol suite already overcomes several of these issues. In BA domain, latency and bandwidth requirements are usually quite moderate [114]. Moreover, Web standards tackle reliability, scalability, and security while wireless technologies solve the mobility problem. Nevertheless, the selection of the communication protocol for the proposed interface has to support several patterns. Besides request-response, also publish-subscribe interaction is required. Furthermore, semantic querying needs to be integrated into the interface. Thus, relevant protocols are analyzed regarding their suitability. On top of IP as central element of the Internet protocol suite, TCP and UDP are most common standards on the transport layer. Regarding the application layer, HTTP and SOAP do not support publish-subscribe mechanisms. Polling would be required which increases the overhead in message exchange. The WS-based integration technologies OPC UA [72], OBIX [73], and BACnet/WS [74] would be suitable for M2M communication. However, they do not use a standardized representation like OWL but describe their own information models. The same applies for KNX WS offering mappings to these technologies. More interesting application layer protocols for M2M communication are the Constrained Application Protocol (CoAP), the Message Queue Telemetry Transport (MQTT), XMPP, or WebSocket. While CoAP targeting resource-constrained devices runs on UDP, XMPP and MQTT utilize the reliable TCP. By default, MQTT does not support request-response interaction. On the other hand, both CoAP and XMPP apply request-response as well as publish-subscribe mechanisms [116, 117]. Nonetheless, they do not describe a mechanism for semantic querying like the SPARQL protocol [118]. With the WebSocket protocol, full-duplex and bidirectional communication can be realized [119]. The analysis of existing M2M approaches leads to the selection of the WebSocket protocol as basis for the SOA of the semantic M2M interface due to the mentioned drawbacks of other protocols.

In Chapter 6 [40], a simplified semantic interface on top of the WebSocket protocol is defined that provides basic access to the ontology. In this work, the semantic abstraction layer is a distinct hardware or software entity that acts as central broker and hosts the common system knowledge as illustrated in Figure 1.8. As the concept of the semantic abstraction layer is generalized later on in [41] to a logical construct hiding technology specifics behind an ontology-based modeling approach, the distinct semantic layer entity is called semantic core to prevent misunderstandings. This semantic core is also used in the proof-of-concept implementations. The focus of the simple interface for accessing the semantic core is the transfer of basic monitoring and control data in the BA context. Data and information are exchanged between the semantic core, and basic services are offered to read and write data values from and to the modeled data services and control services of BA resources. In addition, a querying service allows for retrieving specific context information from the knowledge base. Technology connectors are implemented to map the technology-specific communication in the particular BAS to the abstract semantic interface. Thus, basic syntactic and semantic interoperability between the communication partners is enabled.

On the other hand, a more comprehensive, platform-independent interface for M2M communication by means of Semantic Web standards and common ICT is presented in Chapter 7 [41]. Thus, systems should be able to communicate autonomously avoiding human intervention [114]. Although the focus of the underlying work is on BA communication, data and information about smart grid interaction can also be exchanged as the corresponding concepts are modeled in the underlying ontology, as well. Based on identified requirements regarding interface architecture and M2M communication, a set of relevant service categories is determined that are also visualized in Figure 1.9. In addition to ordinary services for the exchange of process data, special services for publication of available data services and control services as well as direct querying of the ontology are designed. The result is a service-oriented interface consisting of twelve services with the focus on asynchronous peer-to-peer connections of communicating systems:

- Identification services are intended to describe communication partners when initiating a connection as basis for access control, authentication, and authorization. The final service set provides a *register* and a *deregister* service within this category. Partner information is encoded in RDF-based descriptions.

- Publication services should inform connected communication partners about available functions, such as data services and control services in terms of the common BA and smart grid ontology. Based on these service publications, other systems gain knowledge about the distributed functionality in the modeled domain. Publication is realized with the *add* service while published services can be deleted with the *remove* service.

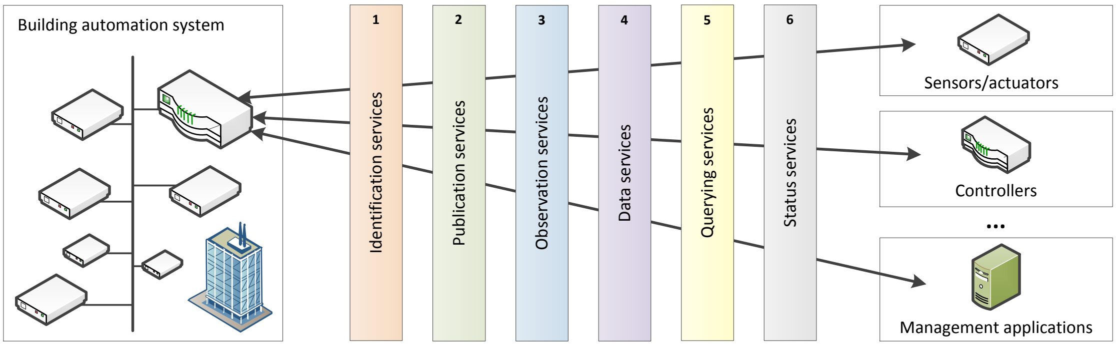

Figure 1.9: Service categories for semantics-based M2M communication [41]

- Observation services are used to listen to changes of published services. For example, value changes of a data service that is modeled in the ontology and provided by a BA sensor can be received without active polling. The service set contains services for registering observations at the communication partner (*observe*) and removing active observations remotely (*detach*).

- Data services summarize the tasks of exchanging process data like sensor values and actuator set point changes. The transferred information is modeled in accordance with the corresponding data services and control services of the ontology. The *put* service is used to push new process data. This can be done due to an active observation of a registered communication partner or after the reception of a *get* call following the request-response pattern.

- Querying services go beyond common M2M interaction of pushing and pulling simple data and information. The idea is to provide services for requesting and updating context information described in the underlying ontology. With the *query* and the *update* service, SPARQL queries are sent while the *query result* functionality returns the result set to the requester.

- Status services are used for error handling and acknowledgments of message reception. The actual *status* service sends a status code as well as a human-readable status definition that can be used for application flow control and debugging. Several of the mentioned services specify a mandatory response in the form of a status message. The status messages are necessary to indicate the processing state on the application level.

The service payloads that are enveloped in WebSocket messages are encoded in a simple, predefined structure similar to HTTP. A message identifier is followed by mandatory and optional header fields. Then, the actual message content based on shared domain knowledge is added, such as SPARQL queries or RDF graphs. Semantic descriptions in the message contents are encoded by means of RDF/XML or Turtle, query results use XML or JSON serialization, and queries are based on the content types specified by the SPARQL protocol [118]. The structured, ontology-based modeling is important in order to be able to tackle future changes and provide necessary extensibility. The individual communication partners have knowledge about their own scope. This islanded knowledge is distributed and shared using the interface services. In contrast to [41], the simple interface of [40] covers only the interface's data service category and the querying service category to read context information from the ontology.

The developed semantic interface is evaluated using a KNX installation as BAS and a simple BMS as high-level application. In addition, the mentioned semantic core is used as message broker between the BAS and the BMS. Thus, two connections are established while there is no direct link between the BMS and the BAS. In order to show the feasibility of the approach, basic test cases are specified that are combined to more complex scenarios. For each of the defined services, at least one basic test case exists. All test cases are successfully executed, and the hardware requirements indicate that deployment on partially constrained devices is possible (e.g. Raspberry PIs). Transmission and processing times of messages are acceptable for non-critical BA applications. Compared to other M2M approaches, this work is built on machine-readable and distributed semantics by means of the contributed ontology that combines context information for smart BA control and smart grid interaction. In summary, the interface and the ontology are key enablers for automatic processing and interpretation in intelligent applications.

1.6.3 Optimization

Within the last part of this thesis, the optimization in BEMSs is addressed based on the proposed semantic abstraction layer. Interoperable communication with integrated BASs and smart grid stakeholders is hidden behind this layer while the designed ontology provides machine-readable semantics of the building context covering building structures, BASs, external influences, or smart grid interaction patterns and enables knowledge inference based on reasoners. As introduced in [42] and [44], building energy management is usually faced with the conflicting goals of comfort maximization and resource efficiency. The task of the corresponding optimization is to find Pareto optimal solutions (i.e. trade-off solutions) trying to tackle both objectives [5]. Within this thesis, a solution is defined as a planned execution schedule determining the state changes of building resources in an optimization period. Comfort maximization or minimization of discomfort costs relies on the building users' requirements concerning different comfort parameters, such as brightness, humidity, temperature, air quality, or noise level. On the other hand, resource efficiency dealing with minimization of energy costs and energy consumption is influenced by the installed BASs, the external energy supply, the local energy supply with decentralized production and buffer storage units, or the information available on the smart grid, such as energy prices. The optimization that takes care of these goals is executed in a building environment that depends on the internal structure of building zones and the external influences including the weather. Figure 1.10 outlines this optimization setting.

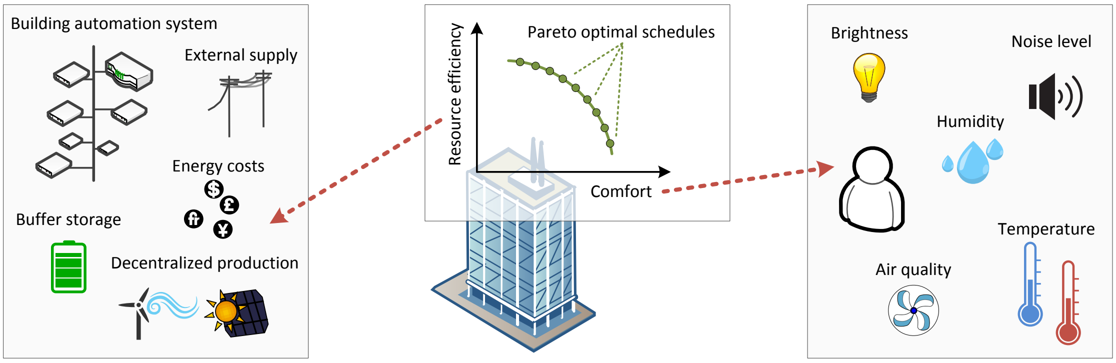

Figure 1.10: Conflicting objectives for optimization in BEMSs [42]

Requirements for BEMSs, which are focused on the HVAC domain but are also relevant for BEMSs in general, refer to independent applicability regarding the building type and the installed equipment as well as simple implementation and design [120]. In this context, it is important that BEMSs have knowledge on building process behavior in order to plan schedules for BAS execution. According to Problem statement 4 of Section 1.2, manual engineering of underlying behavior models can be very expensive with limited reusability. Thus, the suitability of the ontology-based, abstract semantics and the corresponding monitoring data to define data-driven prediction models is analyzed in the following paragraphs. In addition, the design of the optimization problem and the optimization algorithms that utilize the prediction models require detailed expert knowledge. The necessary manual modeling and engineering can result in specific solutions for particular building types, comfort domains, or trades as outlined in Problem statement 5 of Section 1.2. Thus, we decided to use machine-readable context information modeled in the ontology to overcome this issue by specifying an automatic generation process for optimization problems as well as universally applicable strategies for optimization in BEMSs. We show that the semantic abstraction layer introducing the ontology-based modeling is able to solve both problems and provides the suitable basis for a generic and reusable optimization design in BEMSs.

Optimization problem extraction

Taking advantage of the machine-readable context information, the optimization problem with its variables, constants, and constraints can be extracted and formulated automatically as presented in Chapter 8 [42], which is summed up in the following paragraphs. In contrast to manual engineering of BEMSs by domain experts, this extraction approach significantly reduces the required configuration effort and is not tailored to a specific building type or BAS. This eases the BEMS design as the automatically extracted optimization problem is the basis for the optimization task.

A generic objective function that combines both goals of resource efficiency and comfort satisfaction is utilized as basis for automatic initialization. The weighted-sum method converts this bi-objective problem into a single-objective problem formulation [121]. Equation 1.1 shows the resulting objective function that minimizes the sum of the solution's fitness F_t over the optimization period n [42]. Costs of comfort dissatisfaction c_t and energy consumption e_t are weighted using the factor ω.

$$min \sum_{t=1}^{n} F_t \quad \text{where} \quad F_t = \omega \cdot c_t + (1 - \omega) \cdot e_t \tag{1.1}$$

In more detail, the functions c_t and e_t are shown in Equation 1.2 and Equation 1.3 [42]. The evaluation of comfort dissatisfaction c_t considers the square deviation of the estimated actual value of an environmental parameter v_{tpz} and the desired value r_{tpz} per parameter p and zone z at time t. The actual values depend on immutable influences $\boldsymbol{i}_t$ (e.g. weather) as well as states of controllable devices $\boldsymbol{l}_t$ (e.g. BA actuators). Occupancy o_{tz} and parameter priority λ_{tp} are used to weight the deviation. Active measurement and control of parameter p in zone z is indicated by the binary constant m_{pz}.

$$c_t = \sum_{p} \sum_{z} m_{pz} \cdot \lambda_{tp} \cdot o_{tz} \cdot (v_{tpz}(\boldsymbol{l}_t, \boldsymbol{i}_t) - r_{tpz})^2 \tag{1.2}$$

Likewise, costs to balance energy demand and energy supply are summed up over all supported energy types g and the corresponding energy suppliers y. The overall demand per energy type d_{tg} at time t depends again on the immutable influences and the device states. This demand is allocated to the suppliers using the quote q_{ty}. Availability of supplier y for energy type g is specified using j_{gy}. The binary state s_{ty} indicates the activity of the supplier while f_{ty} represents the specific price per energy unit.

$$e_t = \sum_{g} \sum_{y} j_{gy} \cdot q_{ty} \cdot f_{ty} \cdot s_{ty} \cdot d_{tg}(\boldsymbol{l}_t, \boldsymbol{i}_t) \tag{1.3}$$

In order to provide the basis for utilization in the optimization problem formulation, information needs to be queried from the ontology, first. For this purpose, SPARQL queries are sent to the ontology, where semantics about the building environment (e.g. BAS, building structure, smart grid connection) is modeled. The ontology graph is traversed along predefined paths to extract the interesting information. A visualization of the extraction procedure in Figure 1.11 shows this process of reading information from the ontology following certain paths in the modeled graph. The optimization problem formulation that maps the requested information to the objective function and the corresponding variables, constants, and constraints concludes this workflow.

A suitable entry point for the traversal of the ontology graph is the set of modeled comfort parameters. Thus, the first task is to analyze the comfort-related information before the energy-related domains are considered. As sensors and actuators specify monitored and controlled comfort parameters, these devices and their data services as well as control

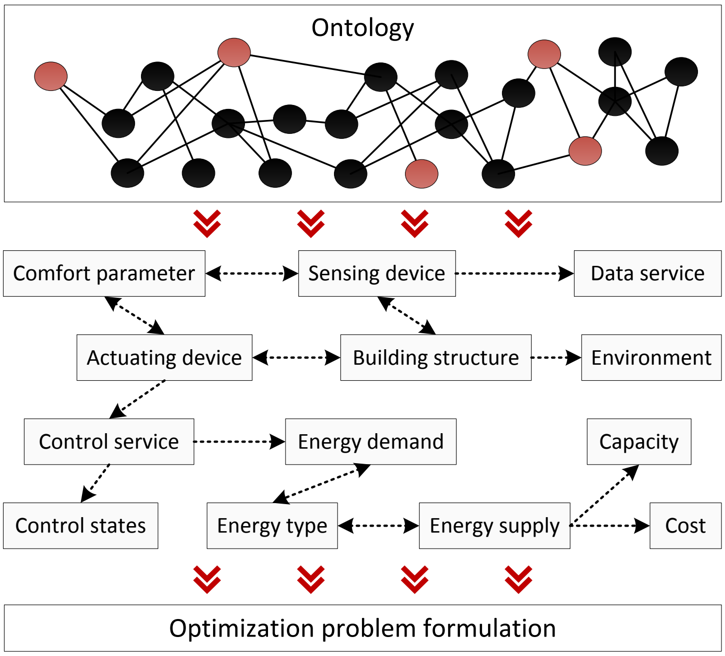

Figure 1.11: Information extraction procedure [42]

services can be queried next. The locations of the devices and the operating areas of the services are used to link the building environment. The states of control services are important in the optimization of BEMSs. Therefore, they are queried in addition to the energy demand required by control services to maintain comfort in their sphere of control. This marks the transition from comfort-related to energy-related information within the extraction procedure. Energy demand requires certain energy types that are provided by energy suppliers. These suppliers and their constraints regarding costs or capacity need to be read, as well. It has to be noted that only energy needs to change or maintain the states of comfort parameters in the building are considered while static energy consumption of BA resources is neglected.

After the extraction of the information from the ontology, the optimization problem is formulated by mapping the information to variables, constants, and constraints of the objective function. Variables are divided into decision variables, which can have a certain range and are modified during the optimization process, and control variables, which are used for indexing purposes. The control variables for comfort parameters p, building zones z, energy types g, energy suppliers y, and time slots of the optimization period t are organized in 1-dimensional collections. The set of decision variables, on the other hand, represents the schedule as output of the optimization in BEMSs. The quote q to describe the coverage ratio of energy demand per supplier, the states s of energy suppliers, and the states l of controllable devices are decision variables of the presented objective function. For the states, the range of discrete or continuous values is cosigned. Moreover, the parameter variations of control services are mapped to data structures in order to enable decision making in the abstract BEMS. Constants that cannot be modified by the optimization are stored in 2-dimensional (e.g. energy prices, occupancy) and 3-dimensional data structures (e.g. desired comfort values). The external, immutable influences are additionally mapped to a 2-dimensional collection. As the optimization is subject to basic conditions, default constraints as well as device-specific and building-specific constraints are derived from the ontology. Regarding storage devices, the constraints cover flow conservation, positive charging levels, or avoidance of charging and discharging at the same time. Moreover, the sum of quotes per energy type needs to be 1. An exemplary, device-specific constraint is the energy supply capacity of a PV plant. Finally, thresholds for comfort in the building are transformed to constraints.

A case study-based evaluation illustrates the applicability and functionality of our approach. A set of BA resources is modeled within a building structure consisting of two office rooms. Energy supply is provided by a battery storage, a PV plant, and the external grid. The building structure and the BA resources as well as the services for data, control, and energy are modeled in the ontology. Based on this setting, the extraction procedure is started by means of reading information from the ontology. The result sets in response to the SPARQL queries are mapped to the components of the optimization problem. The data structures are filled, and the objects for control variations are initialized. An additional CO_2 threshold is modeled to verify the generation of individual constraints. All in all, the developed method of automatic optimization problem generation based on machine-readable semantics modeled in an ontology has considerable advantages. In contrast to the manual design by domain experts, the optimization problem can be derived with limited human intervention. In addition, existing data and information sources like BIM models can be spiled to populate the ontology. Moreover, the populated ontology provides a basis for other applications in building operation, as well. Although there is some non-negligible effort needed for the population of the ontology, the reusable automatic workflow is more efficient than the building-specific, individual configuration by domain experts. Thus, BEMS design is eased due to the utilization of the semantic abstraction layer for automatic generation of optimization problems.

Data-driven time series prediction

In terms of BAS schedule planning, the optimization in BEMSs that minimizes the objective function of [42] demands knowledge about the future behavior of building processes. For example, the local energy production, the estimated energy demand per energy type, or the actual values of comfort parameters in the optimization period need to be predicted. Based on [28] and [30], Khosravani et al. describe three categories of building modeling with focus on energy consumption [29]. This classification can be adopted for building modeling in general. Engineering methods make use of physical and structural properties. As a consequence, there is high need for specific expert knowledge. Historic data, on the other hand, are the basis for statistical methods that correlate relevant inputs with a target output. Regression models or autoregressive integrated moving average (ARIMA) models are examples for this category. Similar to statistical methods, artificial intelligence methods, such as artificial neural networks (ANNs) or support vector machines (SVMs), utilize historic data to model nonlinear or linear process behavior. According to [4], artificial intelligence methods and data-driven models gain importance in the development of more intelligent BEMSs. In the context of this work, the semantic abstraction layer provides a suitable basis for utilization of these data-driven models that helps reducing the need for expert modeling and manual engineering effort. Therefore, an adaptive prediction framework covering relevant time series for target and actual cross-domain comfort as well as energy supply and demand is investigated in Chapter 9 [43], which is outlined in this section.

Related work already uses artificial intelligence methods and especially neural networks in building energy management. In [122], neural networks are used to estimate energy consumption considering several building characteristics for training purposes. A comparison of neural network design approaches in order to predict electric power demand is targeted in [29]. In order to evaluate detected solutions, neural networks as part of an optimization framework based on genetic algorithms (GAs) are addressed in [123]. Yokoyama et al. present a global optimization approach to design neural networks used in energy demand forecasting [124]. The task of feature selection and evaluation for neural network modeling is realized as automatic process in [125]. In contrast to this related work, our approach incorporates prediction of comfort-related time series in addition to the usually addressed energy consumption and production. Based on neural networks, a framework for automatic identification, generation, evaluation, and reconfiguration of data-driven models relevant for building energy management is defined. According to [126], utilization of neural networks is best practice for processes that are complex or difficult to describe. Instead of designing the behavior models manually by means of expert-based engineering, the models that are necessary for the prediction of time series in the optimization period utilize machine learning techniques trained with already available historic monitoring data.

The main aim of the prediction framework is the support of the optimization in BEMSs with preferably accurate forecasts. While some data for the optimization period are fetched into the semantic abstraction layer from external sources, others need to be predicted in advance or during the optimization process considering the changes in the output schedule. Therefore, the prediction framework is located between the ontology of the semantic abstraction layer and the actual optimization in the BEMS, which is illustrated in Figure 1.12. For each relevant time series, a neural network-based model is generated to calculate the required forecasts. Four distinct categories of prediction models are identified in this work:

1. Target comfort covers all models that are used to predict desired indoor comfort values of parameters, such as brightness or temperature. For each parameter modeled and controlled in a building zone, a neural network is generated and trained. The desired comfort values are calculated once at the start of the optimization task.

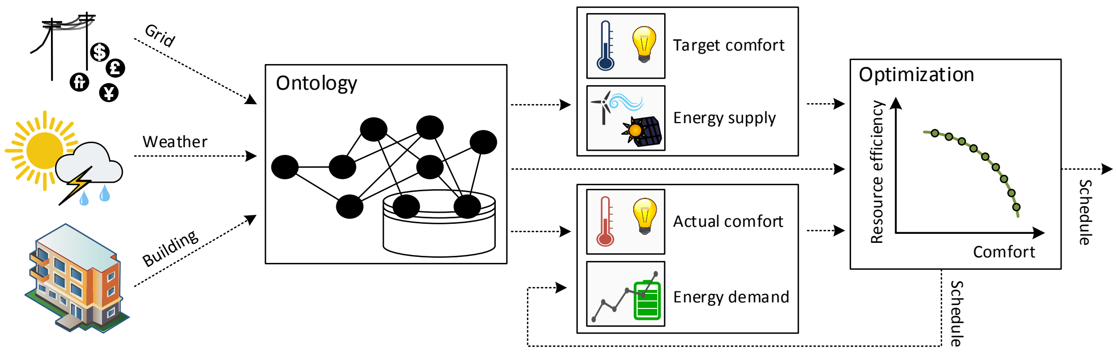

Figure 1.12: Integration of the prediction framework into the optimization workflow [43]

2. Likewise, the production of local energy supply units is estimated prior to the optimization run. For example, the PV production is estimated for the optimization period in order to consider the amount of renewable energy in the schedule planning.

3. The estimation of actual comfort values depends on set point changes of related BA resources. Thus, the corresponding models are run after the optimization leads to changes in the output schedule. The results are used to calculate the comfort dissatisfaction per comfort parameter and building zone.

4. In parallel, the set point changes influence the energy demand in the optimization period. Thus, the neural networks for this category have additional inputs for the states of the relevant BA resources in order to estimate the energy demand per supported energy type.

The prediction framework consists of methods and procedures for model identification, neural network design, performance calculation, model improvements, and online assessment. Thus, the forecast task can be automated without mentionable human intervention. Similar to the extraction of the optimization problem in [42], the prediction model identification makes use of the semantics modeled in the ontology. The main entry points are the data services provided by the BA resources. These data services describe the semantics of data values and are linked to the building context. Monitoring data used to train the neural networks are related to the data services. With respect to the estimation of desired comfort (category 1), data services capturing comfort parameters in building zones are identified as basis for prediction models. Moreover, dependencies of other data services (e.g. outdoor temperature) are added as inputs to these models. Models for the estimation of actual comfort (category 3) have additional inputs for the data representing state changes. Data services measuring local production (category 2) become prediction models, as well. Finally, data services representing metering functionality with respect to energy and power (category 4) necessitate neural networks.

After determining the required neural networks, they are initialized and trained. For this purpose, appropriate training and test data sets are selected taking the characteristics of upcoming optimization periods into account. A test run of a trained network triggers the performance calculation that decides on validity and fitness of the model. If a neural network leads to invalid outputs with respect to the target data, the configuration is modified and the updated model is trained and tested again. This is done until either a valid network according to certain thresholds is designed or the range of available configuration options is exhausted. In the latter, the best invalid network is returned.

The performance calculation method applies several measures m_i $(i = 1..n)$ in order to estimate the fitness of a prediction model with respect to the accuracy of forecasts. This step requires the availability of monitoring data to determine the forecast error $e_t = y_t - f_t$ for the calculation of the performance measures. In this error term, y_t is the monitored value while f_t is the estimated value. Scale-dependent error metrics as well as scale-independent measures are supported. Examples are the mean absolute error (MAE), the maximum absolute error (MAX), the mean absolute scaled error (MASE), or the symmetric mean absolute percentage error (SMAPE). For each model, nonzero thresholds t_i are specified for the measures m_i. If the relative deviation $p_i = (\frac{t_i - m_i}{t_i} \cdot \omega_i) \geq 0$ for all i, the output of the model is valid. The comparison of models utilizes the sum over all p_i in order to decide on the best neural network configuration. The weights ω_i as well as the thresholds t_i are subject to user preferences.

In case of an inaccurate forecast indicating the invalidity of the neural network, the configuration or setup of the neural network is modified in terms of four basic variables. First, the inputs of the model that are known as feature set provide potential for reconfiguration. On the one hand, the number of inputs can be varied. Here, optional features are removed or added again while mandatory features are always used. On the other hand, the length of the training data time frame for the features can be changed between 30 and 180 days. Furthermore, the number of neurons in the hidden layer is adjustable in a range of 4 to 20. Finally, a tapped delay line of variable length between 4 and 12 elements is subject to network reconfiguration. A heuristic is used to traverse through the available network setups in order to avoid trying all possible combinations.

Finally, the online assessment procedure is used during continuous execution of the models in the optimization run. Thus, this forms the counterpart of the design phase where initialization of the neural networks is done. Likewise, the performance calculation and the improvement heuristic are used. The procedure is triggered when monitoring data of a past period are available. Then, these data are compared to the corresponding forecast values leading to a performance metric of the underlying prediction model. If retraining of invalid models is not successful, the configuration needs to be modified.

The evaluation of the approach is based on simulated data due to the lack of a comprehensive data set that describes all relevant time series. Thus, an EnergyPlus simulation model with an HVAC system and a PV plant is utilized. The simulation inputs and outputs provide time series for desired and actual comfort values regarding temperature and humidity as well as energy consumption and production of electricity. The proof-of-concept implementation of the prediction framework is based on the MATLAB Neural Network Toolbox. Tests are run for different time intervals within the simulated year.

The outputs of the continuously checked and improved models are compared to both the actual monitoring data and the forecast produced by neural networks without incremental improvements. In general, performance gains can be identified using the adaptive prediction framework although the accuracy strongly depends on the chosen training set. Enough training data need to be available, and the used time frame has to be selected carefully depending on the current and next optimization periods. A useful extension of the approach is a trend analysis in order to detect potential performance problems in advance. In this case, retraining or reconfiguration can be initiated prior to the occurrence of a potentially bad forecast. Nevertheless, the framework represents an adequate alternative in comparison to the process modeling using engineering methodology keeping in mind the reduced effort and the automatic exploitation of machine-readable semantics to reduce human intervention.

Generic optimization strategies

The last part of this thesis is dedicated to the design of generally applicable optimization strategies that exploit the context information modeled in the ontology. While existing solutions of BEMSs are often tailored to specific buildings, building types, BA equipment, or comfort domains, the semantic abstraction layer supports the definition of building-independent and technology-agnostic optimization strategies by uncoupling the BEMS from the underlying systems and technologies. This work builds on the preliminary work of extracting the optimization problem [42] and preparing data-driven prediction models for optimization support [43] that already takes advantage of the semantically enriched representation of the building environment in the ontology. In order to tackle the issue highlighted in Problem statement 5, we research the design of an intelligent and generic search for feasible solutions with respect to a reusable, building-independent optimization of BAS schedules in Chapter 10 [44], which is summed up in the following paragraphs.

Building energy management is an active field of research. A variety of methods is used for optimization of energy efficiency as well as comfort satisfaction. Merabti et al. review and compare approaches in the HVAC domain based on classical controllers, fuzzy-based controllers, neural network controllers, and GA controllers [127]. In [128], an extensive survey on control systems used in energy and comfort management of smart buildings is presented. In this context, existing work is categorized into the considered objectives, the covered comfort parameters, or the implemented optimization methods, such as mixed-integer linear programming (MILP) or ANNs. Another survey analyzes optimization approaches for design, planning, and control of renewable energy [129]. In [130], MILP methodology is used in the residential domain in order to solve a mathematical problem formulation. The appliances are modeled using the concepts of end-user, intermediate, and support services. Another MILP-based energy management tackles renewable energy sources, thermal models, and storage devices in a smart home setting [131]. While this approach is designed to find the optimal solution, a heuristic-based approach is presented in [31] due to the potential complexity of solving optimization problems in BEMSs. Here, thermal comfort in terms of air conditioning as well as energy costs considering

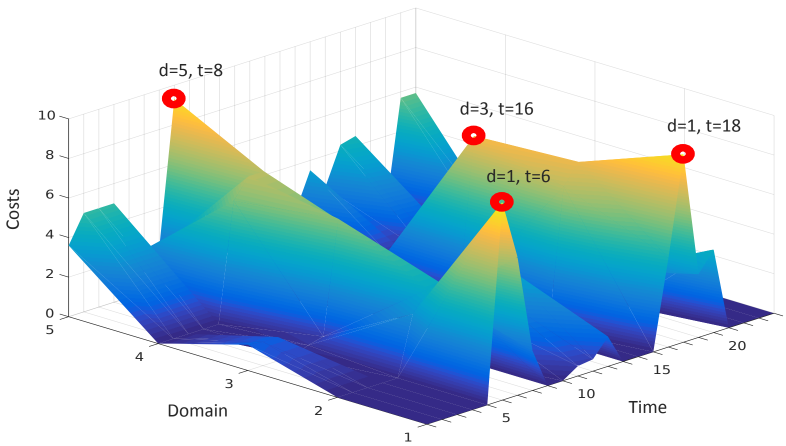

Figure 1.13: Subproblem identification based on an exemplary cost distribution [44]

electricity is optimized with three heuristic algorithms. The proposed BEMS in [132] integrates multiple domains, such as HVAC, lighting, and shading. The optimization utilizes Lagrangian relaxation and stochastic dynamic programming (DP). Moreover, optimization in building energy management is often coupled with model-predictive control (MPC). An example for thermal and non-thermal appliances in the residential domain is presented in [133]. In addition, evolutionary algorithms for building energy management are utilized in [134]. Here, candidate solutions as output of the optimization algorithm are evaluated in an energy simulation core. Particle swarm optimization (PSO) as another population-based approach is applied to a multi-objective optimization discussed in [5]. This work is reused in a multi-agent system in order to realize indoor comfort control spread over several building zones [135].

Although suitable algorithms and heuristics are used to solve the actual optimization task, the overall BEMS designs are often very specific to certain buildings, comfort parameters, or BA trades. For example, a lot of work exists for the HVAC domain as this has usually high energy needs. This tailoring limits an easy and convenient deployment and reconfiguration for reuse in different settings. Therefore, we use algorithm design methodology in order to develop three generally applicable, context-aware strategies that can be integrated into generic metaheuristics as a common basis for customized BEMS implementations. The main idea is to develop a smart search using context information that inherits problem-specific knowledge.

First, problems in terms of high costs in the fitness of a solution (i.e. BAS schedule) are analyzed in order to identify a proper starting point for improvements. This method is in line with the divide and conquer principle as the overall schedule and the corresponding costs as output of the objective function are split into subproblems that can be resolved subsequently. For this purpose, the costs of comfort dissatisfaction per domain (zone and parameter) as well as the costs of energy per domain (grid and supplier) are fed into an $n \times m$ matrix with n as the length of the optimization period and m as the number of

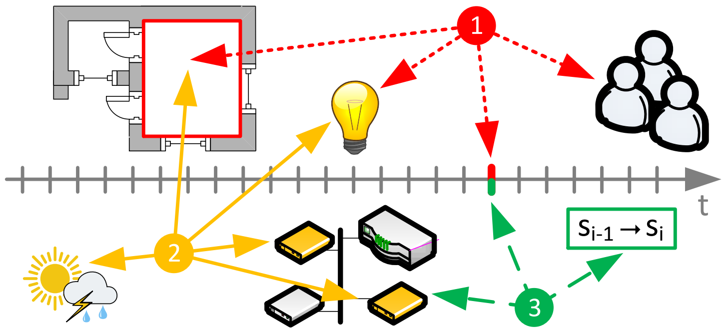

Figure 1.14: Simplified schedule modification schema [44]

domains. In general, high costs indicate problems in the schedule, for example, when lots of energy is consumed from the external grid at a high price level. In order to find a set of candidate subproblems for further processing, the components of the matrix are prioritized according to their costs. In addition, the temporal position in the optimization period is considered. Thus, earlier time slots have higher priority of being chosen. The final decision of subproblem selection utilizes the roulette-wheel method. Figure 1.13 shows an exemplary cost distribution over 5 domains and 24 time slots as well as the marked candidate subproblems.

Second, the selected subproblem is solved by means of finding the cause of the high costs, which is based on exploiting the underlying context information modeled in the ontology. This task leads to schedule adaptions in terms of state changes. This partial modification method can be described in three steps that are illustrated in Figure 1.14:

1. The main input to the partial modification is the domain of interest. If the selected subproblem describes a comfort domain, the corresponding comfort parameter, the building zone, and the intended user target are considered. On the other hand, subproblems related to energy costs take into account the local grid segment and the energy supplier. There is no user target as the energy costs should be zeroed out, in general.

2. Based on this input, the modification procedure uses context information in order to identify candidate solutions for solving the subproblem. For example, other BA resources acting in the same comfort domain are selected in accordance to their spatial and temporal impact on the current subproblem. Moreover, constraints and external conditions are integrated into this search. A set of neighborhoods is defined in proximity to the subproblem under investigation. Neighbors in these structures are characterized by state changes that are called moves. Each neighbor differs in at least one move from the current schedule. A set of multiple moves is called a path. In terms of comfort problems, the paths and moves in the neighborhoods are focused on a reduction or increase of the current comfort value in order to reach a lower or higher target value, respectively. For energy-related problems,

the neighbors try to substitute energy-intensive consumers, make use of cheaper suppliers, or reduce comfort over-fulfillment. Load shifting is realized in terms of storage resources.

3. The final step is to search the neighborhoods in an efficient way. Hence, the neighbors are ranked and a priority-proportional selection is used to decide on a suitable path to change schedule S_{i-1} leading to schedule S_i. For prioritization, the direct and indirect influences of moves and paths on the domain under investigation and the related domains are observed. In summary, better candidates are picked with higher probability, but all neighbors in the various neighborhoods have a statistical chance of being selected.

Third, the experiences made in the modification procedure are returned to the ontology. Thus, the smart optimization can infer new knowledge based on the detected impacts of schedule changes with respect to the comfort deviation, the energy consumption, or the energy costs. This additional knowledge is available for future optimization runs supporting faster convergence to better solutions. Based on the differences between two consecutive schedules S_{i-1} and S_i, the differences in domain values and the temporal impacts of state changes can be estimated, which are manifested in three rules:

- Specific rules represent the exact impact of a certain move. This is done for the domain value as well as for the delay between the set action and the response of the underlying building process.
- Generic rules are built similar to the control variations of the ontology. Relative state changes of the ordered device states are related to relative changes of the domain values.
- Basic rules describe the generic associations between building zones, comfort parameters, energy grid segments, and BA resources.

Common metaheuristics that are abstract and problem-independent are used as containers for the developed strategies. Thus, a reusable basis is provided for individual BEMS implementations combining both universal applicability and context-awareness. Metaheuristics are divided into single-solution and population-based methods [35]. In the GA as an example for population-based metaheuristics, the impact assessment is placed in the population evaluation. The partial modification is used in the mutation operator while subproblem identification is integrated into the recombination phase and can also provide starting points for the mutation. An adapted version of the VND metaheuristic is realized in a proof-of-concept implementation. Multiple neighborhoods are searched in ascending order until a better solution is found. With this new solution, the loop starts again with the first neighborhood. In the implementation, this VND core is extended by an outer loop to iterate over the selected subproblems. Partial modification is used to pick neighbors that are created for each subproblem. Impact assessment is done after the selection of a new neighbor.

The proof-of-concept implementation of this extended VND is evaluated by means of a variation of case studies. Different settings are defined in order to test the optimization strategies regarding building processes with different dynamics, artificial (e.g. privacy) and real (e.g. temperature) comfort parameters, load shifting, or conflict resolution. In [44], we highlighted three exemplary case studies in order to show the functionality of the optimization strategies and to describe the feasibility of smart solution space reduction and inference of gained knowledge. One case study is focused on the trade-off between room brightness, user privacy, and electricity consumption. While artificial lighting can be used to influence the brightness, installed blinds are able to change the room brightness as well as the intended privacy. Another case study targets the thermal comfort with a central heating system. The third case study is inspired by electric vehicle charging combined with a local energy storage. Starting from an initial situation, the optimization behavior is observed and analyzed. Moreover, the execution of the use cases points out the derivation of rules describing state change impacts. This continuous extension of the ontology with new rules is a major advantage of the context-aware optimization approach. Important issues that need to be considered in further tests and deployments are the prioritization in the optimization strategies, the definition of scaling factors and termination thresholds, or a focus on a diverse set of initial solutions if the strategies are embedded into a population-based method.

All in all, the developed, context-aware optimization approach for BEMSs represents a universally applicable and reusable alternative compared to tailored solutions for specific buildings, BASs, or comfort domains. Instead of directly coding the expert knowledge into a BEMS implementation, machine-readable semantics is modeled in an ontology that can also be used as a basis for other applications in the scope of smart buildings. Thus, the optimization strategies can be applied for different scenarios and settings. On top of the semantic abstraction layer providing the ontological concepts and the communication interface, this work presents an abstract BEMS design. BAS integration into the semantic abstraction layer utilizes WS-based integration solutions. Smart grid interaction, on the other hand, makes use of an IP-centric communication stack and a standardized communication language. In summary, our work targets the three layers of integration, abstraction, and optimization to contribute to the design of intelligent BEMSs.

1.7 Scientific contribution

This work makes several contributions to the current state of the art with respect to the problem statements and the hypotheses introduced in Section 1.2. The following paragraphs give an overview of the scientific contributions.

Interoperable integration of heterogeneous systems. In order to overcome heterogeneity in the BA domain, WSs on top of IP are identified as future-proof technology. We contribute an extension of the RESTful BACnet/WS standard by formalizing the defined object model and introducing basic types necessary for BAS integration on a technology-independent level [37]. A tag-based modeling framework for the description of BASs in a building context is contributed, as well. This modeling framework offers transformations to standardized WS-based integration solutions [38]. Furthermore, uniform communication in the smart grid environment is ensured by the selection of an appropriate protocol stack combined with a suitable communication language on top in order to cover the most relevant DR interaction scenarios between smart buildings and grid operators, energy retailers, or energy aggregators [36].

Generic semantic modeling for building energy management. Explicit modeling of semantics is a key enabler for automatic processing and intelligent applications in the smart building domain. In this context, ontologies are widely used for specific purposes. Thus, a generic ontology is designed that harmonizes the semantic modeling of BASs with respect to BEMSs. The core components are data services and control services linked to the building structure, which are provided by BA resources abstracting from technology-specific characteristics [40]. On the other hand, the ontology addresses the integration of smart buildings into smart grids by means of introducing concepts to describe smart grid interaction scenarios [39]. The generic ontology is able to be linked with other, more specialized ontologies following the idea of the Semantic Web.

Uncoupling of BEMSs from underlying systems and technologies. In order to uncouple BASs and smart grid communication interfaces on the lower level from management applications like BEMSs on the higher level, a semantic abstraction layer is introduced as virtual separation in between. For this purpose, a SOA for M2M communication extending existing approaches is defined that provides services to exchange context information as well as runtime data by means of Semantic Web technologies [41]. The ontology specified in this work provides the necessary basis to model the exchanged message payloads.

Automatic and reusable design process of optimization in BEMSs. One main contribution of this work addresses the simplification of BEMS design, which is often done manually by domain experts resulting in high effort and costs at limited reusability. The interoperable integration of underlying systems and subsystems as well as the semantic modeling of their functionality and their relations to the building context and the building users leads to a suitable basis for automatic and high-level processing. As expert knowledge is already described in the ontology of the semantic abstraction layer, a reusable extraction workflow for optimization problems in the BEMS context is contributed [42]. An adaptive prediction framework for time series based on monitoring data linked to the BA resources and the building zones is defined in order to support the evaluation of solutions in optimization problem solving [43]. The final contribution comprises universally applicable, heuristic optimization strategies to find reasonable solutions for energy-efficient building operation in the form of BAS schedules [44]. The focus is on the exploitation of available semantics with respect to the design of an intelligent optimization mechanism.

1.8 Conclusion

In general, BASs are able to support an energy-efficient building operation, which gains in importance due to the increasing global energy needs of residential and commercial buildings. However, the heterogeneity in existing BA technologies and standards, the continuous integration of buildings into the emerging smart grids and smart cities, or the lack of context information and machine-readable semantics are inhibitors for reusable, universal, and abstract BEMSs. Thus, this work presents the design of a building energy management optimization based on a semantic abstraction layer in order to overcome the mentioned issues. The underlying publications tackle the problem statements from several different perspectives in order to verify the hypotheses introduced in Section 1.2.

Hypothesis 1. *WS-based SOAs provide a suitable basis for interoperable integration of BA technologies into management applications. A technology-independent, tag-based modeling of BASs as well as model-driven transformation rules support automated BAS integration into multiple WS-based integration technologies.*

Regarding a seamless and interoperable integration of BASs into management applications, RESTful BACnet/WS is utilized as described in [37]. The specified object model that supports a detailed description of BASs is converted into a machine-readable representation. Additional type definitions for common building blocks of BASs are specified. Evaluation is based on a proof-of-concept implementation of a BACnet/WS server. With respect to an automated integration of BASs into management applications (e.g. BEMSs) or distributed networks (e.g. IoT) by means of WSs, a modeling and transformation framework is provided in [38]. Besides a technology-independent modeling of BASs based on an extensible tag vocabulary, model-driven transformations are specified to map the defined tag-based models to the WS-based target technologies OBIX, BACnet/WS, and OPC UA as relevant representatives for BAS integration. This shows the suitability of WS-based SOAs for BAS integration and the capability of model-driven principles for an automated integration.

Hypothesis 2. *An IP-based protocol stack enables the utilization of different physical layers in smart grid communication. On top, OASIS EI is able to cover relevant interaction patterns in a DR context between buildings and other smart grid agents.*

A communication architecture is introduced with a focus on DR interaction of smart buildings and other smart grid agents in [36]. Following the MAS paradigm, a communication stack with IP as its central element and an appropriate communication language on top is elaborated that satisfies the requirements of smart grid communication. It is shown that IP ensures the reuse of existing Web infrastructure based on various physical layers while it provides an abstraction for overlying applications. A prototype utilizing XMPP and OASIS EI is presented in order to demonstrate the results and verify the stated hypothesis.

Hypothesis 3. *An ontology as part of a semantic abstraction layer between BEMSs, BASs, and smart grids provides the required basis for structured modeling of semantics* *in the field of building energy management. The resulting representation of the building context is essential for independent and automatic processing.*

Once interoperable communication with BASs and smart grid agents is established, a semantic abstraction layer is described in order to provide a virtual uncoupling between technology-independent management applications (e.g. BEMSs) and technology-specific automation and communication systems (e.g. BASs). For this purpose, an ontology is defined that is able to model the connection of buildings with the smart grid and its agents in [39]. Concepts to describe grid agents, communication technologies, services interfaces, and grid structures enable the semantic modeling of smart grid interaction scenarios. On the other hand, smart control of BASs needs to be supported by the semantic abstraction layer. For this purpose, ontological concepts for building structures, BA resources, data services, and control services are presented in [40]. Thus, the relations of BASs with the building context can be described in a structured and explicit form. The result is a common ontology implemented by means of the OWL standard that is important for subsequent, automatic processing towards generic BEMSs. Moreover, a semantic interface is proposed to support interoperable M2M communication on an abstract, semantic level based on the ontology. Utilizing common Web technologies and existing M2M approaches, a SOA is described on top of the WebSocket protocol in [41]. Services for communication partner identification, publication of BA functions, subscriptions, process data exchange, or semantic querying ease seamless M2M communication. Message contents are based on the specified ontology. Test case-based analysis regarding feasibility and hardware requirements is performed in order to show the applicability of the proposed approach. In summary, the presented work about the semantic abstraction layer with the ontology and the semantic M2M interface confirms the stated hypothesis, which becomes more apparent in the context of the publications built on this abstraction layer [42, 43, 44].

Hypothesis 4. *Monitoring data embedded into the semantic abstraction layer implies knowledge on building process behavior that is required for optimization in BEMSs. Neural networks for time series prediction can be automatically designed and reconfigured utilizing the context information modeled in the ontology.*

An optimization of BAS schedules within a BEMS requires knowledge on future behavior of building processes resulting from changes of device states or external influences. This work presents an approach for an autonomous and adaptive prediction framework published in [43]. In order to forecast relevant time series in the context of BEMSs, monitoring data are used that implicitly contain building process characteristics. This data-driven approach utilizes neural networks as a suitable learning-based technique to create forecast models. The generation and configuration process can be automated based on the context information available in the underlying ontology. Furthermore, evaluation shows that continuous performance assessment, which triggers optional reconfiguration of the instantiated neural networks, leads to promising results without the need of additional human intervention. Therefore, the modeled context information and the corresponding monitoring data provide a sufficient basis for the required time series prediction.

Hypothesis 5. *Context information modeled in the ontology can be exploited to automatically extract optimization problems and to design abstract and generic optimization strategies for universal application in BEMSs.*

The semantic abstraction layer is the essential element for the design of an intelligent optimization in BEMSs. Therefore, this thesis presents an approach for the automatic generation of optimization problems based on the modeled expert knowledge in the common ontology. Semantics about the building context is queried from the ontology, which results in a generic definition of the objective function, the decision variables, the constants, and the corresponding constraints as published in [42]. Minimization of comfort dissatisfaction and minimization of energy costs are considered, which are necessary in a user-oriented but energy-efficient building operation. The derived optimization problem formulation can be further used in a distinct optimization algorithm. Reusability, applicability, and functionality of the approach are discussed by means of an exemplary case study. This automatically generated optimization problem and the previously mentioned forecast models are the basis for universally applicable optimization strategies that are independent of specific buildings, building types, or comfort domains. In contrast to existing approaches that are often limited in their reusability, expert knowledge mapped to the ontology is used to design an intelligent search in order to plan ahead energy-saving and comfort-compliant BAS schedules as defined in [44]. The strategies to divide and conquer the overall optimization problem as well as to derive new knowledge regarding impacts of state changes are embedded into common metaheuristics. Case studies are used to evaluate a proof-of-concept implementation and verify the hypothesis.

1.9 Future work

Although the results of the presented work proposing an energy management optimization based on a semantic abstraction layer show the benefits of a reusable and abstract BEMS design, future work is required towards an elaborate solution with respect to market-readiness. In order to test the performance in a real-world situation, demo buildings and users need to be acquired. This is a tough task as resistance to the use of private data will arise. On the other hand, real-world tests are necessary to improve the quality of the described proof-of-concept implementations and their composition into a consistent system architecture aiming at energy-efficient building operation. Moreover, such tests give information about possible limitations or constraints of the proposed approach in the context of real-world applications. This way, comparisons of the approach with customized BEMS implementations become possible, leading to additional evaluation results. With respect to the smart grid integration, the support of DR programs and interaction patterns in the optimization should be extended to release the full potential of flexibility trading and regional or national balancing of energy demand and supply. Besides the benefit of lower costs for building operators or users, also grid operators and energy retailers will take advantage of this approach due to reduced investments for better infrastructure and compensation of fluctuations. Following the idea of the Semantic Web, the linking of the specified ontology for the support of abstract optimization in BEMSs to other, probably more specialized state-of-the-art ontologies should be promoted in order to establish a broader basis for future work in this field of application.

References

[1] L. Pérez-Lombard, J. Ortiz, and C. Pout, “A review on buildings energy consumption information”, *Energy and Buildings*, vol. 40, no. 3, pp. 394–398, 2008, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2007.03.007.

[2] GEA, *Global Energy Assessment: Toward a Sustainable Future.* Cambridge University Press, Cambridge, UK and New York, NY, USA and the International Institute for Applied Systems Analysis, Laxenburg, Austria, 2012, ISBN: 9780521182935.

[3] International Energy Agency, *Transition to Sustainable Buildings: Strategies and Opportunities to 2050.* OECD/IEA, 2013, ISBN: 9789264202412.

[4] M. Manic, D. Wijayasekara, K. Amarasinghe, and J. J. Rodriguez-Andina, “Building Energy Management Systems: The Age of Intelligent and Adaptive Buildings”, *IEEE Industrial Electronics Magazine*, vol. 10, no. 1, pp. 25–39, Mar. 2016, ISSN: 1932-4529. DOI: 10.1109/MIE.2015.2513749.

[5] R. Yang and L. Wang, “Multi-objective optimization for decision-making of energy and comfort management in building automation and control”, *Sustainable Cities and Society*, vol. 2, no. 1, pp. 1–7, Feb. 2012, ISSN: 2210-6707. DOI: 10.1016/j.scs.2011.09.001.

[6] VDI 3814 Blatt 1, *Gebäudeautomation (GA) – Systemgrundlagen*, Nov. 2009.

[7] VDI 3813 Blatt 1, *Gebäudeautomation (GA) – Grundlagen der Raumautomation*, May 2011.

[8] VDI 3813 Blatt 2, *Gebäudeautomation (GA) – Raumautomationsfunktionen (RA-Funktionen)*, May 2011.

[9] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey”, *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2010.05.010.

[10] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web”, *Scientific American*, vol. 284, no. 5, pp. 29–37, May 2001.

[11] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, “Smart Objects as Building Blocks for the Internet of Things”, *IEEE Internet Computing*, vol. 14, no. 1, pp. 44–51, Jan.-Feb. 2010, ISSN: 1089-7801. DOI: 10.1109/MIC.2009.143.

[12] X. Fang, S. Misra, G. Xue, and D. Yang, “Smart Grid – The New and Improved Power Grid: A Survey”, *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, Fourth Quarter 2012, ISSN: 1553-877X. DOI: 10.1109/SURV.2011.101911.00087.

[13] P. Palensky and D. Dietrich, “Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads”, *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381–388, Aug. 2011, ISSN: 1551-3203. DOI: 10.1109/TII.2011.2158841.

[14] J. Haase, G. Zucker, and M. Alahmad, “Energy efficient building automation: A survey paper on approaches and technologies for optimized building operation”, in *Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2014, pp. 5350–5356. DOI: 10.1109/IECON.2014.7049317.

[15] DIN EN 15232, *Energieeffizienz von Gebäuden – Einfluss von Gebäudeautomation und Gebäudemanagement*, Sep. 2012.

[16] W. Kastner, L. Krammer, and A. Fernbach, “State of the Art in Smart Homes and Buildings”, in *Industrial Communication Technology Handbook*, R. Zurawski, Ed., Second Edition, CRC Press, 2014, ch. 55, pp. 1–20, ISBN: 9781482207323. DOI: 10.1201/b17365-66.

[17] ISO 16484-2, *Building automation and control systems (BACS) – Part 2: Hardware*, Aug. 2004.

[18] H. Dibowski, J. Ploennigs, and K. Kabitzsch, “Automated Design of Building Automation Systems”, *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3606–3613, Nov. 2010, ISSN: 0278-0046. DOI: 10.1109/TIE.2009.2032209.

[19] J. Ploennigs, H. Dibowski, U. Ryssel, and K. Kabitzsch, “Ganzheitlicher, automatischer Entwurf drahtloser Gebäudeautomationssysteme”, *at - Automatisierungstechnik*, vol. 61, no. 6, pp. 393–402, Jun. 2013, ISSN: 0178-2312. DOI: 10.1524/auto.2013.0034.

[20] M. Jung, J. Weidinger, W. Kastner, and A. Olivieri, “Building automation and smart cities: An integration approach based on a service-oriented architecture”, in *IEEE International Conference on Advanced Information Networking and Applications*, Mar. 2013, pp. 1361–1367. DOI: 10.1109/WAINA.2013.200.

[21] M. Jung, J. Weidinger, C. Reinisch, W. Kastner, C. Crettaz, A. Olivieri, and Y. Bocchi, “A transparent IPv6 multi-protocol gateway to integrate Building Automation Systems in the Internet of Things”, in *IEEE International Conference on Green Computing and Communications*, Nov. 2012, pp. 225–233. DOI: 10.1109/GreenCom.2012.42.

[22] H. Farhangi, “The Path of the Smart Grid”, *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, Jan.-Feb. 2010, ISSN: 1540-7977. DOI: 10.1109/MPE.2009.934876.

[23] V. C. Güngör, D. Sahin, T. Kocak, S. Ergüt, C. Buccella, C. Cecati, and G. P. Hancke, “Smart Grid Technologies: Communication Technologies and Standards”, *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 529–539, Nov. 2011, ISSN: 1551-3203. DOI: 10.1109/TII.2011.2166794.

[24] OASIS, *Energy Interoperation Version 1.0, OASIS Standard*, Jun. 2014.

[25] C. Martinez-Cruz, I. J. Blanco, and M. A. Vila, "Ontologies versus relational databases: Are they so different? A comparison", *Artificial Intelligence Review*, vol. 38, no. 4, pp. 271–290, Dec. 2012, ISSN: 0269-2821. DOI: 10.1007/s10462-011-9251-9.

[26] W3C OWL Working Group, *OWL 2 Web Ontology Language Document Overview (Second Edition), W3C Recommendation*, Dec. 2012.

[27] W3C RDF Working Group, *RDF 1.1 Concepts and Abstract Syntax, W3C Recommendation*, Feb. 2014.

[28] H.-x. Zhao and F. Magoulès, "A review on the prediction of building energy consumption", *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 3586–3592, Aug. 2012, ISSN: 1364-0321. DOI: 10.1016/j.rser.2012.02.049.

[29] H. R. Khosravani, M. D. M. Castilla, M. Berenguel, A. E. Ruano, and P. M. Ferreira, "A Comparison of Energy Consumption Prediction Models Based on Neural Networks of a Bioclimatic Building", *Energies*, vol. 9, no. 1, pp. 1–24, Jan. 2016, ISSN: 1996-1073. DOI: 10.3390/en9010057.

[30] A. Foucquier, S. Robert, F. Suard, L. Stéphan, and A. Jay, "State of the art in building modelling and energy performances prediction: A review", *Renewable and Sustainable Energy Reviews*, vol. 23, pp. 272–288, Jul. 2013, ISSN: 1364-0321. DOI: 10.1016/j.rser.2013.03.004.

[31] M. Arikiez, F. Grasso, D. Kowalski, and M. Zito, "Heuristic Algorithm for Minimizing the Electricity Cost of Air Conditioners on a Smart Grid", in *IEEE International Energy Conference*, Apr. 2016, pp. 1–6. DOI: 10.1109/ENERGYCON.2016.7513915.

[32] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, "The NeOn Methodology for Ontology Engineering", in *Ontology Engineering in a Networked World*, M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi, Eds., Springer Berlin Heidelberg, 2012, ch. 2, pp. 9–34, ISBN: 9783642247934. DOI: 10.1007/978-3-642-24794-1_2.

[33] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", Stanford Knowledge Systems Laboratory, Tech. Rep. KSL-01-05, Mar. 2001, pp. 1–25.

[34] M. Uschold and M. Gruninger, "Ontologies: Principles, Methods and Applications", *Knowledge Engineering Review*, vol. 11, no. 2, pp. 93–136, Jun. 1996, ISSN: 0269-8889. DOI: 10.1017/S0269888900007797.

[35] M. Gendreau and J.-Y. Potvin, Eds., *Handbook of Metaheuristics*, Second Edition, ser. International Series in Operations Research & Management Science. Springer US, 2010, ISBN: 9781441916655. DOI: 10.1007/978-1-4419-1665-5.

[36] D. Schachinger, S. Gaida, and W. Kastner, "Smart grid communication at the interface of customer buildings with focus on demand response", in *International Symposium on Smart Electric Distribution Systems and Technologies*, Sep. 2015, pp. 368–373. DOI: 10.1109/SEDST.2015.7315236.

[37] D. Schachinger, C. Stampfel, and W. Kastner, "Interoperable integration of building automation systems using RESTful BACnet Web services", in *Annual Conference of the IEEE Industrial Electronics Society*, Nov. 2015, pp. 3899–3904. DOI: 10.1109/IECON.2015.7392708.

[38] D. Schachinger, A. Fernbach, and W. Kastner, "Modeling framework for IoT integration of building automation systems", *at - Automatisierungstechnik*, vol. 65, no. 9, pp. 630–640, Sep. 2017, ISSN: 0178-2312. DOI: 10.1515/auto-2017-0014.

[39] D. Schachinger, W. Kastner, and S. Gaida, "Ontology-based abstraction layer for smart grid interaction in building energy management systems", in *IEEE International Energy Conference*, Apr. 2016, pp. 1–6. DOI: 10.1109/ENERGYCON.2016.7513991.

[40] D. Schachinger and W. Kastner, "Semantics for smart control of building automation", in *IEEE International Symposium on Industrial Electronics*, Jun. 2016, pp. 1073–1078. DOI: 10.1109/ISIE.2016.7745041.

[41] ——, "Semantic interface for machine-to-machine communication in building automation", in *IEEE International Workshop on Factory Communication Systems*, May 2017, pp. 1–9. DOI: 10.1109/WFCS.2017.7991956.

[42] ——, "Ontology-based generation of optimization problems for building energy management", in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2017, pp. 1–8. DOI: 10.1109/ETFA.2017.8247565.

[43] D. Schachinger, J. Pannosch, and W. Kastner, "Adaptive learning-based time series prediction framework for building energy management", in *IEEE International Conference on Industrial Electronics for Sustainable Energy Systems*, Jan. 2018, pp. 453–458. DOI: 10.1109/IESES.2018.8349919.

[44] D. Schachinger and W. Kastner, "Context-aware optimization strategies for universal application in smart building energy management", in *IEEE International Conference on Industrial Informatics*, Jul. 2018, pp. 478–483.

[45] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges", *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 5–20, First Quarter 2013, ISSN: 1553-877X. DOI: 10.1109/SURV.2012.021312.00034.

[46] M. Meisel, T. Leber, M. Ornetzeder, M. Stachura, A. Schiffleitner, G. Kienesberger, J. Wenninger, and F. Kupzog, "Smart Demand Response Scenarios", in *IEEE Africon*, Sep. 2011, pp. 1–6. DOI: 10.1109/AFRCON.2011.6072138.

[47] W. Wang, Y. Xu, and M. Khanna, “A survey on the communication architectures in smart grid”, *Computer Networks*, vol. 55, no. 15, pp. 3604–3629, Oct. 2011, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2011.07.010.

[48] S. Rohjans, M. Uslar, R. Bleiker, J. González, M. Specht, T. Suding, and T. Weidelt, “Survey of Smart Grid Standardization Studies and Recommendations”, in *IEEE International Conference on Smart Grid Communications*, Oct. 2010, pp. 583–588. DOI: 10.1109/SMARTGRID.2010.5621999.

[49] R. Kyusakov, J. Eliasson, J. van Deventer, J. Delsing, and R. Cragie, “Emerging Energy Management Standards and Technologies - Challenges and Application Prospects”, in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2012, pp. 1–8. DOI: 10.1109/ETFA.2012.6489674.

[50] E. H. Et-Tolba, M. Maaroufi, and M. Ouassaid, “Demand side management in smart grid by multi-agent systems technology”, in *International Conference on Multimedia Computing and Systems*, Apr. 2014, pp. 1042–1045. DOI: 10.1109/ICMCS.2014.6911211.

[51] M. E. Collins, R. W. Silversides, and T. C. Green, “Multi-Agent System Control and Coordination of an Electrical Network”, in *International Universities Power Engineering Conference*, Sep. 2012, pp. 1–5. DOI: 10.1109/UPEC.2012.6398641.

[52] M. Pichler, “Communication Patterns for Demand Side Flexibility”, in *Symposium on Communications for Energy Systems*, Sep. 2014, pp. 19–28.

[53] J. G. Cupp and M. E. Beehler, “Implementing Smart Grid Communications”, *Burns & McDonnell TECHBriefs*, no. 4, pp. 5–8, 2008.

[54] F. Lobo, A. Cabello, A. Lopez, D. Mora, and R. Mora, “Distribution network as communication system”, in *CIRED Seminar SmartGrids for Distribution*, Jun. 2008, pp. 1–4. DOI: 10.1049/ic:20080448.

[55] P. Saint-Andre, K. Smith, and R. Tronçon, *XMPP: The Definitive Guide*, First Edition. O’Reilly Media, 2009, ISBN: 9780596521264.

[56] A. B. Johnston, *SIP: Understanding the Session Initiation Protocol*, Third Edition. Artech House, 2009, ISBN: 9781607839958.

[57] ZigBee Alliance and HomePlug Alliance, *Smart Energy Profile 2 Application Protocol Standard*, Apr. 2013.

[58] OpenADR Alliance, *OpenADR 2.0 Profile Specification, B Profile*, Jan. 2013.

[59] W. Kastner, G. Neugschwandtner, S. Soucek, and H. M. Newmann, “Communication Systems for Building Automation and Control”, *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1178–1203, Jun. 2005, ISSN: 0018-9219. DOI: 10.1109/JPROC.2005.849726.

[60] D. Dietrich, D. Bruckner, G. Zucker, and P. Palensky, "Communication and Computation in Buildings: A Short Introduction and Overview", *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3577–3584, Nov. 2010, ISSN: 0278-0046. DOI: 10.1109/TIE.2010.2046570.

[61] T. Sauter, S. Soucek, W. Kastner, and D. Dietrich, "The Evolution of Factory and Building Automation", *IEEE Industrial Electronics Magazine*, vol. 5, no. 3, pp. 35–48, Sep. 2011, ISSN: 1932-4529. DOI: 10.1109/MIE.2011.942175.

[62] ISO 16484-5, *Building automation and control systems (BACS) – Part 5: Data communication protocol*, May 2017.

[63] ISO/IEC 14543-3-1, *Information technology – Home electronic systems (HES) architecture – Part 3-1: Communication layers – Application layer for network based control of HES Class 1*, Sep. 2006.

[64] ISO/IEC 14908-1, *Information technology – Control network protocol – Part 1: Protocol stack*, Nov. 2012.

[65] G. Lilis, G. Conus, N. Asadi, and M. Kayal, "Integrating Building Automation Technologies with Smart Cities: An Assessment Study of Past, Current and Future Interoperable Technologies", in *International Conference on Smart Cities and Green ICT Systems*, May 2015, pp. 370–375. DOI: 10.5220/0005478803700375.

[66] J. Ploennigs, M. Lehmann, B. Wollschlaeger, T. L. Mai, and K. Kabitzsch, "Multi-objective Device Selection Approach for Component-based Automation Systems", in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2014, pp. 1–8. DOI: 10.1109/ETFA.2014.7005224.

[67] A. C. Oezluek, J. Ploennigs, and K. Kabitzsch, "Designing Building Automation Systems Using Evolutionary Algorithms with Semi-Directed Variations", in *IEEE International Conference on Systems, Man and Cybernetics*, Oct. 2010, pp. 2328–2335. DOI: 10.1109/ICSMC.2010.5641996.

[68] R. Drath, A. Fay, and M. Barth, "Interoperabilität von Engineering-Werkzeugen", *at - Automatisierungstechnik*, vol. 59, no. 7, pp. 451–460, Jul. 2011, ISSN: 0178-2312. DOI: 10.1524/auto.2011.0938.

[69] F. Jammes and H. Smit, "Service-Oriented Paradigms in Industrial Automation", *IEEE Transactions on Industrial Informatics*, vol. 1, no. 1, pp. 62–70, Feb. 2005, ISSN: 1551-3203. DOI: 10.1109/TII.2005.844419.

[70] W. Kastner, M. Kofler, M. Jung, G. Gridling, and J. Weidinger, "Building Automation Systems Integration into the Internet of Things: The IoT6 approach, its realization and validation", in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2014, pp. 1–9. DOI: 10.1109/ETFA.2014.7005197.

[71] D. Guinard, I. Ion, and S. Mayer, “In Search of an Internet of Things Service Architecture: REST or WS-*? A Developers’ Perspective”, in *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, A. Puiatti and T. Gu, Eds., vol. 104, Springer Berlin Heidelberg, 2012, pp. 326–337, ISBN: 9783642309724. DOI: 10.1007/978-3-642-30973-1_32.

[72] OPC Foundation, *OPC Unified Architecture Specification, Part 1: Overview and Concepts, Release 1.03*, Oct. 2015.

[73] OASIS, *OBIX Version 1.1, Committee Specification Draft 02 / Public Review Draft 02*, Dec. 2013.

[74] ASHRAE, *Data Communication Protocol for Building Automation and Control Networks (ANSI/ASHRAE Addendum am to ANSI/ASHRAE Standard 135-2012)*, Apr. 2016.

[75] W. Granzer and W. Kastner, “Information Modeling in Heterogeneous Building Automation Systems”, in *IEEE International Workshop on Factory Communication Systems*, May 2012, pp. 291–300. DOI: 10.1109/WFCS.2012.6242577.

[76] ASHRAE, *Proposed Addendum am to Standard 135-2012, BACnet - A Data Communication Protocol for Building Automation and Control Networks*, Apr. 2014.

[77] ——, *A Data Communication Protocol for Building Automation and Control Networks (ANSI/ASHRAE Addendum c to ANSI/ASHRAE Standard 135-2004)*, Oct. 2006.

[78] D. Schachinger and W. Kastner, “Model-Driven Integration of Building Automation Systems into Web Service Gateways”, in *IEEE World Conference on Factory Communication Systems*, May 2015, pp. 1–8. DOI: 10.1109/WFCS.2015.7160561.

[79] L. Christiansen, M. Hoernicke, and A. Fay, “Modellgestütztes Engineering: Basis für die Automatisierung der Automatisierung”, *atp edition*, vol. 56, no. 3, pp. 18–27, Feb. 2014.

[80] M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in Practice*. Morgan & Claypool, 2012, ISBN: 9781608458837. DOI: 10.2200/S00441ED1V01Y201208SWE001.

[81] Project Haystack Corporation, *Project Haystack*, http://project-haystack.org, accessed: Jan. 16, 2017.

[82] J. Bézivin, “On the unification power of models”, *Software and Systems Modeling*, vol. 4, no. 2, pp. 171–188, May 2005, ISSN: 1619-1374. DOI: 10.1007/s10270-005-0079-0.

[83] X. H. Wang, D. Q. Zhang, T. Gu, and H. K. Pung, “Ontology Based Context Modeling and Reasoning using OWL”, in *IEEE Annual Conference on Pervasive Computing and Communications Workshops*, Mar. 2004, pp. 18–22. DOI: 10.1109/PERCOMW.2004.1276898.

[84] A. Crapo, X. Wang, J. Lizzi, and R. Larson, “The Semantically Enabled Smart Grid”, in *Grid-Inperop*, Nov. 2009, pp. 177–185.

[85] W3C RDF Working Group, *RDF Schema 1.1, W3C Recommendation*, Feb. 2014.

[86] IEC 61970-301, *Energy management system application program interface (EMS-API) – Part 301: Common information model (CIM) base*, Dec. 2016.

[87] IEC 61968-11, *Application integration at electric utilities – System interfaces for distribution management – Part 11: Common information model (CIM) extensions for distribution*, Mar. 2013.

[88] S. Rohjans, M. Uslar, and H.-J. Appelrath, “OPC UA and CIM: Semantics for the Smart Grid”, in *IEEE PES Transmission and Distribution Conference and Exposition*, Apr. 2010, pp. 1–8. DOI: 10.1109/TDC.2010.5484299.

[89] E. Doğdu, A. M. Özbayoğlu, O. Benli, H. E. Akınç, E. Erol, T. Atasoy, O. Güreç, and Ö. Erçin, “Ontology-centric Data Modelling and Decision Support in Smart Grid Applications: A Distribution Service Operator Perspective”, in *IEEE International Conference on Intelligent Energy and Power Systems*, Jun. 2014, pp. 198–204. DOI: 10.1109/IEPS.2014.6874179.

[90] Q. Zhou, W. Guan, and W. Sun, “Impact of Demand Response Contracts on Load Forecasting in a Smart Grid Environment”, in *IEEE Power and Energy Society General Meeting*, Jul. 2012, pp. 1–4. DOI: 10.1109/PESGM.2012.6345079.

[91] L. Daniele, F. den Hartog, and J. Roes, “Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology”, in *Formal Ontologies Meet Industry*, ser. Lecture Notes in Business Information Processing, R. Cuel and R. Young, Eds., vol. 225, Springer International Publishing, 2015, pp. 100–112, ISBN: 9783319215440. DOI: 10.1007/978-3-319-21545-7_9.

[92] D. Bonino and F. Corno, “DogOnt - Ontology Modeling for Intelligent Domotic Environments”, in *The Semantic Web - ISWC 2008*, ser. Lecture Notes in Computer Science, A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, Eds., vol. 5318, Springer Berlin Heidelberg, 2008, pp. 790–803, ISBN: 9783540885634. DOI: 10.1007/978-3-540-88564-1_51.

[93] M. J. Kofler, C. Reinisch, and W. Kastner, “A semantic representation of energy-related information in future smart homes”, *Energy and Buildings*, vol. 47, pp. 169–179, Apr. 2012, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2011.11.044.

[94] T. G. Stavropoulos, D. Vrakas, D. Vlachava, and N. Bassiliades, “BOnSAI: A Smart Building Ontology for Ambient Intelligence”, in *ACM International Conference on Web Intelligence, Mining and Semantics*, Jun. 2012, pp. 1–12. DOI: 10.1145/2254129.2254166.

[95] N. M. Tomašević, M. Č. Batić, L. M. Blanes, M. M. Keane, and S. Vraneš, "Ontology-based facility data model for energy management", *Advanced Engineering Informatics*, vol. 29, no. 4, pp. 971–984, Oct. 2015, ISSN: 1474-0346. DOI: 10.1016/j.aei.2015.09.003.

[96] H. Dibowski and K. Kabitzsch, "Ontology-Based Device Descriptions and Device Repository for Building Automation Devices", *EURASIP Journal on Embedded Systems*, vol. 2011, pp. 1–17, Dec. 2011, ISSN: 1687-3955. DOI: 10.1155/2011/623461.

[97] J. Ploennigs, B. Hensel, H. Dibowski, and K. Kabitzsch, "BASont - A modular, adaptive Building Automation System Ontology", in *Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2012, pp. 4827–4833. DOI: 10.1109/IECON.2012.6389583.

[98] A. M. Gibney, S. Rea, M. Lehmann, S. Thior, S. Lesecq, M. Hendriks, C. Gardeux, L. T. Mai, F. Pacull, J. Ploennigs, T. Basten, and D. Pesch, "A Systematic Engineering Tool Chain Approach for Self-organizing Building Automation Systems", in *Annual Conference of the IEEE Industrial Electronics Society*, Nov. 2013, pp. 7696–7701. DOI: 10.1109/IECON.2013.6700416.

[99] M. Glawe and A. Fay, "Wissensbasiertes Engineering automatisierter Anlagen unter Verwendung von AutomationML und OWL", *at - Automatisierungstechnik*, vol. 64, no. 3, pp. 186–198, Mar. 2016, ISSN: 0178-2312. DOI: 10.1515/auto-2015-0077.

[100] S. Runde, H. Dibowski, A. Fay, and K. Kabitzsch, "A Semantic Requirement Ontology for the Engineering of Building Automation Systems by means of OWL", in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2009, pp. 1–8. DOI: 10.1109/ETFA.2009.5346991.

[101] G. F. Schneider, P. Pauwels, and S. Steiger, "Ontology-Based Modeling of Control Logic in Building Automation Systems", *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3350–3360, Dec. 2017, ISSN: 1551-3203. DOI: 10.1109/TII.2017.2743221.

[102] ISO 16739, *Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries*, Apr. 2013.

[103] J. Beetz, J. van Leeuwen, and B. de Vries, "IfcOWL: A case of transforming EXPRESS schemas into ontologies", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 23, no. 1, pp. 89–101, Feb. 2009, ISSN: 0890-0604. DOI: 10.1017/S0890060409000122.

[104] P. Pauwels and W. Terkaj, "EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology", *Automation in Construction*, vol. 63, pp. 100–133, Mar. 2016, ISSN: 0926-5805. DOI: 10.1016/j.autcon.2015.12.003.

[105] E. Corry, P. Pauwels, S. Hu, M. Keane, and J. O'Donnell, "A performance assessment ontology for the environmental and energy management of buildings", *Automation in Construction*, vol. 57, pp. 249–259, Sep. 2015, ISSN: 0926-5805. DOI: 10.1016/j.autcon.2015.05.002.

[106] M. Asfand-e-yar, A. Kucera, and T. Pitner, "Smart Buildings: Semantic Web Technology for Building Information Model and Building Management System", in *International Conference on Data and Software Engineering*, Nov. 2014, pp. 1–6. DOI: 10.1109/ICODSE.2014.7062671.

[107] G. F. Schneider, Y. Kalantari, G. D. Kontes, S. Steiger, and D. V. Rovas, "An Ontology-Based Tool for Automated Configuration and Deployment of Technical Building Management Services", in *Central European Symposium on Building Physics and BauSIM*, Sep. 2016, pp. 865–872.

[108] J. Ploennigs, A. Schumann, and F. Lécué, "Adapting Semantic Sensor Networks for Smart Building Diagnosis", in *The Semantic Web - ISWC 2014*, ser. Lecture Notes in Computer Science, P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, D. Vrandečić, P. Groth, N. Noy, K. Janowicz, and C. Goble, Eds., vol. 8797, Springer International Publishing, 2014, pp. 308–323, ISBN: 9783319119144. DOI: 10.1007/978-3-319-11915-1_20.

[109] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Bergés, D. Culler, R. K. Gupta, M. B. Kjærgaard, M. Srivastava, and K. Whitehouse, "Brick: Metadata schema for portable smart building applications", *Applied Energy*, vol. 226, pp. 1273–1292, Sep. 2018, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2018.02.091.

[110] J. Han, Y.-K. Jeong, and I. Lee, "A Rule-Based Ontology Reasoning System for Context-Aware Building Energy Management", in *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, Oct. 2015, pp. 2134–2142. DOI: 10.1109/CIT/IUCC/DASC/PICOM.2015.317.

[111] M. Grassi, M. Nucci, and F. Piazza, "Ontologies for Smart Homes and Energy Management: An Implementation-driven Survey", in *IEEE Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*, May 2013, pp. 1–3. DOI: 10.1109/MSCPES.2013.6623319.

[112] G. F. Schneider, "Towards Aligning Domain Ontologies with the Building Topology Ontology", in *Linked Data in Architecture and Construction Workshop*, Nov. 2017, pp. 1–9. DOI: 10.13140/RG.2.2.21802.52169.

[113] E. Curry, J. O'Donnell, E. Corry, S. Hasan, M. Keane, and S. O'Riain, "Linking building data in the cloud: Integrating cross-domain building data using linked data", *Advanced Engineering Informatics*, vol. 27, no. 2, pp. 206–219, Apr. 2013, ISSN: 1474-0346. DOI: 10.1016/j.aei.2012.10.003.

[114] T. Bourgeau, H. Chaouchi, and P. Kirci, "Machine-to-Machine Communications", in *Next-Generation Wireless Technologies: 4G and Beyond*, N. Chilamkurti, S. Zeadally, and H. Chaouchi, Eds., Springer London, 2013, ch. 11, pp. 221–241, ISBN: 9781447151630. DOI: 10.1007/978-1-4471-5164-7_11.

[115] ETSI, *Machine-to-Machine communications (M2M): M2M service requirements (ETSI TS 102 689 V2.1.1)*, Jul. 2013.

[116] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A Survey on Application Layer Protocols for the Internet of Things", *Transaction on IoT and Cloud Computing*, vol. 3, no. 1, pp. 11–17, Jan. 2015.

[117] D. Thangavel, X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan, "Performance Evaluation of MQTT and CoAP via a Common Middleware", in *IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Apr. 2014, pp. 1–6. DOI: 10.1109/ISSNIP.2014.6827678.

[118] W3C SPARQL Working Group, *SPARQL 1.1 Protocol, W3C Recommendation*, Mar. 2013.

[119] V. Pimentel and B. G. Nickerson, "Communicating and Displaying Real-Time Data with WebSocket", *IEEE Internet Computing*, vol. 16, no. 4, pp. 45–53, Jul.-Aug. 2012, ISSN: 1089-7801. DOI: 10.1109/MIC.2012.64.

[120] E. Atam, "New Paths Toward Energy-Efficient Buildings: A Multiaspect Discussion of Advanced Model-Based Control", *IEEE Industrial Electronics Magazine*, vol. 10, no. 4, pp. 50–66, Dec. 2016, ISSN: 1932-4529. DOI: 10.1109/MIE.2016.2615127.

[121] I. Y. Kim and O. L. de Weck, "Adaptive weighted-sum method for bi-objective optimization: Pareto front generation", *Structural and Multidisciplinary Optimization*, vol. 29, no. 2, pp. 149–158, Feb. 2005, ISSN: 1615-147X. DOI: 10.1007/s00158-004-0465-1.

[122] B. B. Ekici and U. T. Aksoy, "Prediction of building energy consumption by using artificial neural networks", *Advances in Engineering Software*, vol. 40, no. 5, pp. 356–362, May 2009, ISSN: 0965-9978. DOI: 10.1016/j.advengsoft.2008.05.003.

[123] E. Asadi, M. G. da Silva, C. H. Antunes, L. Dias, and L. Glicksman, "Multi-objective optimization for building retrofit: A model using genetic algorithm and artificial neural network and an application", *Energy and Buildings*, vol. 81, pp. 444–456, Oct. 2014, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2014.06.009.

[124] R. Yokoyama, T. Wakui, and R. Satake, "Prediction of energy demands using neural network with model identification by global optimization", *Energy Conversion and Management*, vol. 50, no. 2, pp. 319–327, Feb. 2009, ISSN: 0196-8904. DOI: 10.1016/j.enconman.2008.09.017.

[125] S. F. Crone and N. Kourentzes, "Feature selection for time series prediction – A combined filter and wrapper approach for neural networks", *Neurocomputing*, vol. 73, no. 10–12, pp. 1923–1936, Jun. 2010, ISSN: 0925-2312. DOI: `10.1016/j.neucom.2010.01.017`.

[126] M. Benedetti, V. Cesarotti, V. Introna, and J. Serranti, "Energy consumption control automation using Artificial Neural Networks and adaptive algorithms: Proposal of a new methodology and case study", *Applied Energy*, vol. 165, pp. 60–71, Mar. 2016, ISSN: 0306-2619. DOI: `10.1016/j.apenergy.2015.12.066`.

[127] S. Merabti, B. Draoui, and F. Bounaama, "A review of control systems for energy and comfort management in buildings", in *International Conference on Modelling, Identification and Control*, Nov. 2016, pp. 478–486. DOI: `10.1109/ICMIC.2016.7804161`.

[128] P. H. Shaikh, N. B. M. Nor, P. Nallagownden, I. Elamvazuthi, and T. Ibrahim, "A review on optimized control systems for building energy and comfort management of smart sustainable buildings", *Renewable and Sustainable Energy Reviews*, vol. 34, pp. 409–429, Jun. 2014, ISSN: 1364-0321. DOI: `10.1016/j.rser.2014.03.027`.

[129] R. Baños, F. Manzano-Agugliaro, F. G. Montoya, C. Gil, A. Alcayde, and J. Gómez, "Optimization methods applied to renewable and sustainable energy: A review", *Renewable and Sustainable Energy Reviews*, vol. 15, no. 4, pp. 1753–1766, May 2011, ISSN: 1364-0321. DOI: `10.1016/j.rser.2010.12.008`.

[130] D. L. Ha, H. Joumaa, S. Ploix, and M. Jacomino, "An optimal approach for electrical management problem in dwellings", *Energy and Buildings*, vol. 45, pp. 1–14, Feb. 2012, ISSN: 0378-7788. DOI: `10.1016/j.enbuild.2011.11.027`.

[131] F. De Angelis, M. Boaro, D. Fuselli, S. Squartini, F. Piazza, and Q. Wei, "Optimal Home Energy Management Under Dynamic Electrical and Thermal Constraints", *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1518–1527, Aug. 2013, ISSN: 1551-3203. DOI: `10.1109/TII.2012.2230637`.

[132] B. Sun, P. B. Luh, Q.-S. Jia, Z. Jiang, F. Wang, and C. Song, "Building Energy Management: Integrated Control of Active and Passive Heating, Cooling, Lighting, Shading, and Ventilation Systems", *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 588–602, Jul. 2013, ISSN: 1545-5955. DOI: `10.1109/TASE.2012.2205567`.

[133] C. Chen, J. Wang, Y. Heo, and S. Kishore, "MPC-Based Appliance Scheduling for Residential Building Energy Management Controller", *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1401–1410, Sep. 2013, ISSN: 1949-3053. DOI: `10.1109/TSG.2013.2265239`.

[134] I. Mauser, J. Müller, F. Allerding, and H. Schmeck, "Adaptive building energy management with multiple commodities and flexible evolutionary optimization", *Renewable Energy*, vol. 87, pp. 911–921, Mar. 2016, ISSN: 0960-1481. DOI: `10.1016/j.renene.2015.09.003`.

[135] R. Yang and L. Wang, “Multi-zone building energy management using intelligent control and optimization”, *Sustainable Cities and Society*, vol. 6, pp. 16–21, Feb. 2013, ISSN: 2210-6707. DOI: 10.1016/j.scs.2012.07.001.

CHAPTER 2

Smart grid communication at the interface of customer buildings with focus on demand response

Abstract: Traditional power grids lack an appropriate infrastructure to link the involved stakeholders and domains for balancing energy demand and supply. The transmission infrastructure is hierarchically oriented with active bulk generators and passive consumers. Therefore, a bidirectional communication system is needed, which is an essential component of the future smart grid. However, a set of requirements and challenges has to be addressed in order to realize the intended communication infrastructure. In this work, a multi-agent system architecture is presented that tackles these requirements. With a focus on customer buildings and demand response communication patterns, an interoperable and scalable as well as standardized system is defined, which uses the Internet Protocol as central element in the communication stack. OASIS Energy Interoperation standard is used as agent communication language for data exchange between smart grid stakeholders. Furthermore, a proof-of-concept implementation is realized to illustrate the functional capability of the presented approach.

Keywords: Smart grids, communication systems, smart buildings, smart homes, information exchange, demand response, Internet Protocol

The content of this chapter has been published: D. Schachinger, S. Gaida, and W. Kastner, "Smart grid communication at the interface of customer buildings with focus on demand response", in *International Symposium on Smart Electric Distribution Systems and Technologies*, Sep. 2015, pp. 368–373.

2.1 Introduction

Building energy consumption regarding both commercial and residential sector accounts for approximately 40% of final energy consumption in the US and EU while total energy consumption and CO_2 emission are continuously increasing [1]. Thus, buildings provide a vast potential for energy efficiency and optimization measures. However, the electrical infrastructure has not been changed for many decades. Central energy producers provide electricity to consumers via a unidirectional, top-down electric power transmission. There is hardly any coordination between demand and supply to realize energy optimization measures. In order to address this situation, new opportunities emerged in recent years, for example, due to liberalization of energy market and technological progress in energy production and transmission. The traditional infrastructure is going to be replaced by the next generation power grid, the smart grid [2]. Reduction of peak demand, optimization of energy consumption, or monitoring grid stability are only a few issues tackled by the smart grid concept. A major characteristic of this intelligent system is a bidirectional flow of both electricity and information [3]. Additionally, the hierarchical structure with centralized generation is substituted by an adaptive network supporting distributed generation, self-healing mechanisms, or islanding [4]. In parallel, renewable energy based on solar, wind, or hydro power is gradually crowding out non-renewable energy sources as oil, gas, and coal. The smart grid provides the necessary intelligence to handle these trends and to ensure effectiveness and efficiency in energy production and consumption [5].

In the smart grid, many applications can be enabled by reducing the information imbalance based on the introduction of an appropriate information and communication technology (ICT) infrastructure. Decentralized energy resources (DERs) using renewable energy sources can be seamlessly integrated and coordinated in order to enhance reliability and efficiency of power grids [6]. Moreover, smart metering is used by producers as well as consumers to gain insights in energy consumption. Therefore, operational efficiency can be optimized exploiting the exchange of information between smart grid stakeholders [4]. Demand response (DR) systems can be installed to enable a close cooperation between utility companies and customers. Economic incentives like real-time pricing (RTP) induce energy consumers to shift loads and adapt scheduling in order to reduce their costs. On the other hand, an appropriate pricing strategy also supports the utilities trying to flatten consumption during peak periods [5]. To sum up, the smart grid concept facilitates advanced service security, minimization of costs and energy consumption, and protection of resources.

Currently, energy demand is very insensitive to price changes and other constraints. Additionally, DERs need high management effort in the grid to ensure reliability and coordination. Hence, an appropriate communication network is inevitable in order to realize the smart grid vision [7]. Requirements for this communication system are elaborately summarized and discussed in literature [2, 5, 6, 7]. Interoperability, scalability, Quality of Service (QoS), reliability, availability, and security are identified as major requirements. Different wired and wireless network technologies can be used as basis for a smart grid communication system each with advantages and disadvantages depending on the field of application [5, 6, 8, 9]. Based on these technologies, the Internet Protocol (IP) offers a promising network layer technology that is independent of underlying technology and fulfills QoS criteria [8]. In this context, standardization is a key factor to overcome heterogeneity of existing systems and provides a common and reliable infrastructure for future developments. Although organizations and committees have already published several standards and roadmaps, these are mainly concentrated only on individual domains like transmission or distribution [7].

Thus, there is a need for a framework to enable interoperable smart grid communication. In this work, the focus is on DR-related communication at the interface between smart commercial or residential buildings of customers and other smart grid stakeholders like energy retailers. A reliable and interoperable communication infrastructure based on IP is defined using the multi-agent system (MAS) concept. Already available standards are used to specify the information exchange in the form of data signals on the application level. It is discussed, if such a communication system is suitable to meet the introduced requirements and guarantee DR interaction between smart grid agents and customer buildings.

The rest of this work is organized as follows. Section 2.2 describes the common architecture including the smart grid stakeholders. Section 2.3 addresses the communication system based on the introduced architecture covering the protocol stack as well as the language for data exchange. An implementation of the proposed system is presented in Section 2.4. Subsequently, the outcome of this work is discussed in Section 2.5. Finally, Section 2.6 concludes the paper.

2.2 Architectural concept

The next generation power grid covers various domains and stakeholders. First, energy generators are responsible for bulk generation of electricity consumed by customer buildings of residential or commercial sector. Second, transmission and distribution utilities are part of the electric grid. Maintenance and management of transmission and distribution are covered by an operator. Additionally, stakeholders of the energy market, such as energy retailers or energy aggregators, are required in order to coordinate supply and demand. Last but not least, customers participate in the smart grid environment ranging from family homes up to industrial buildings. Besides consuming energy, customers also generate and store electricity [5].

These individual stakeholders need to be integrated into a common smart grid architecture. For this purpose, the concept of MASs is utilized [10, 11, 12]. According to this paradigm, an agent can operate autonomously with certain flexibility in problem solving regarding the particular objectives. Moreover, agents interact with each other and their environment. The described stakeholders can be specified as agents as they match these properties. Figure 2.1 presents this MAS framework for the smart grid including the agents and the necessary infrastructure. Transmission, distribution, and operation are merged to the grid

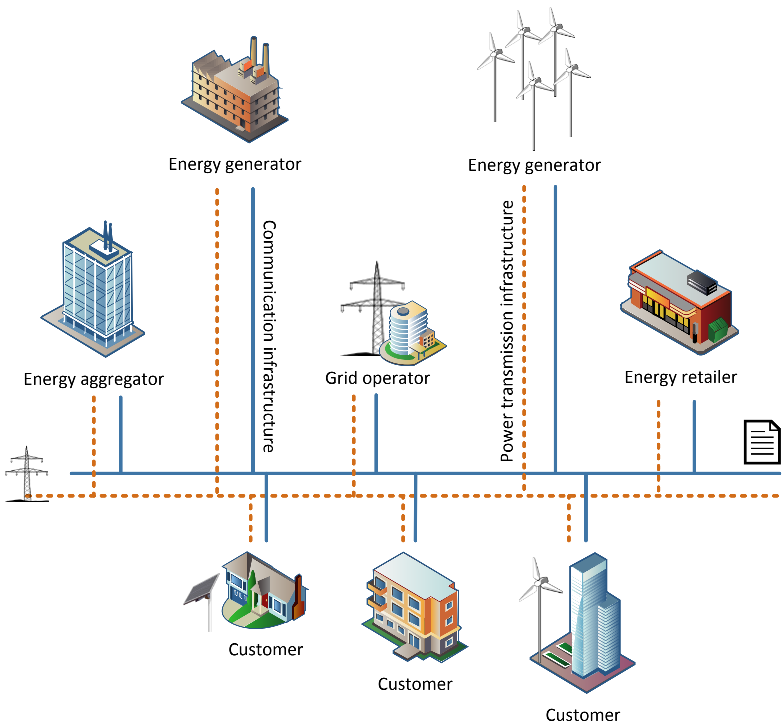

Figure 2.1: MAS framework for smart grid communication

operator agent. Generators of both renewable and non-renewable energy are sketched in the figure. Customers differ in their size as well as energy production capabilities as outlined by local wind turbines or solar panels. Most important for the realization of the smart grid concept is the communication infrastructure besides the existing electric transmission infrastructure. Although these networks might share the same medium, they are addressed separately due to their different requirements [13]. Hence, Figure 2.1 shows two distinct infrastructures.

In order to provide a communication system that is intelligible to all participants of the smart grid, standardized ICT has to be used instead of proprietary technologies. While approved technologies exist for some domains, an interoperable solution covering all domains is necessary to ease the realization of the smart grid. Otherwise, gateways and couplers have to be used to connect heterogeneous grid segments. Furthermore, the agents need to share a common language, a so-called agent communication language (ACL) [10]. Addressing reliability, the proposed MAS should use meshed topologies to overcome failures based on malfunctioning links in the communication system. In the end, agents should be able to cooperate, negotiate, and communicate [14].

In this work, the main focus is on the customer and its interaction with other agents. According to [15], the customer agent is modeled as aggregated system comprising the overall functionality to buy, sell, consume, produce, and store energy. For example, DERs are managed by the customer agent and are not directly integrated into the smart grid. For the sake of simplicity, energy generators are neglected in the remaining work. Hence, observed customer interaction is limited to energy aggregators, grid operators, and energy retailers in the context of DR.

In summary, the introduced smart grid architecture forms a MAS covering several domains that interact to balance energy demand and supply. The potential interaction scenarios can be generalized to a set of abstract communication patterns [16]. Interaction between energy retailer and customer is affected by the exchange of price curves and consumption forecasts. Energy retailers send price information regarding RTP or time-of-use (TOU) pricing. On the contrary, the customer provides an energy consumption forecast for a particular time horizon to support the formation of prices in the energy market domain. In terms of DR, this corresponds to market-oriented mechanisms [17]. Grid operators, on the other hand, try to maintain grid stability by using both market and physical DR. Using market DR, grid operators buy flexibilities from customers to ensure stability. Flexibilities are potential increases or decreases in energy consumption of customers, which can be sold to the grid operator without violating internal constraints like user comfort requirements. Communication between customer and grid operator based on physical DR is implemented with regulatory commands instructing the customer to adapt the amount of energy consumed from the grid. Besides energy retailer and grid operator, the customer interacts with energy aggregators, which operate similar to grid operators trading flexibilities. However, these flexibilities are not used to maintain grid stability, but to bundle small amounts of flexibilities or locally produced energy and resell them on the energy market.

2.3 Communication system

The proposed MAS provides a universal framework for developing a communication system between agents of different domains. The aim is to identify a set of suitable technologies and standards in order to form a reliable and interoperable communication stack as basis for agent communication in the smart grid (see Section 2.3.1). On top of this stack, standardized data exchange mechanisms are introduced to abstract communication from the particular domain (see Section 2.3.2).

2.3.1 Protocol stack

Generally, existing ICT infrastructure should be used as far as possible to ease utilization of the proposed communication system. Otherwise, cost and effort for rearranging existing mechanisms will instantly exceed potential advantages. For example, the Internet infrastructure and its protocols are well suited for this purpose [7]. In order to provide a structured view on the proposed protocol stack, a bottom-up approach is chosen. Therefore, the ISO Open Systems Interconnection (OSI) reference model serves as a template. Although there are seven layers defined in the OSI model, this approach merges physical and data link layer as well as session, presentation, and application layer. The four resulting layers are discussed in the following.

1) Physical layer: The physical media to form a network infrastructure can be basically grouped into wireless and wired systems each with several subtypes [6]. Ethernet and powerline communication (PLC) are popular representatives of wired technologies [7]. Data transmission technology using telephone lines is provided by digital subscriber line (DSL). In the wireless domain, radio-frequency technologies such as IEEE 802.11 or IEEE 802.15.4/ZigBee are utilized. Additionally, communication via cellular networks is a possible solution to connect agents in a smart grid infrastructure. Available technologies are, for example, Global System for Mobile Communications (GSM) or Universal Mobile Telecommunications System (UMTS) [6].

On the other hand, a differentiation regarding spatial dimension can be identified. In this work, a distinction between wide area networks (WANs) and local area networks (LANs) is made to separate wide-area backbone environments and locally concentrated network segments. WANs are used to link LANs, which are intended to cover smaller areas in the smart grid MAS like a neighborhood of a few customer agents. For short distances, wireless technologies like IEEE 802.11 or ZigBee are beneficial as no additional wiring is necessary, and data rates as well as covered range are often sufficient [5, 6]. Likewise, widely available Ethernet installations or existing PLC infrastructure can be used in LAN environments. In WANs, wired DSL or Ethernet are suitable technologies. Furthermore, cellular networks can be used as wireless alternative in WANs [6]. Especially in sparsely populated areas cellular technologies are most promising.

To sum up, some media types and technologies on the physical layer are impracticable while others are more beneficial depending on the geographical location or the area of application. The proposed MAS to realize the future smart grid communication does not need a particular technology on the physical layer, but can operate on many different network types. Thus, interoperability between the agents is ensured irrespective of their location, physical infrastructure, or installed data transmission medium.

2) Network and transport layer: Next, the network layer has to be specified on top of the physical layer. Besides the support of current as well as prospective services, this OSI layer has to be independent of underlying network technologies. Additionally, a scalable architecture combined with appropriate QoS measures is needed [8]. While electric utilities operate their own, incompatible WANs, IP has become a de facto standard for transmitting data due to the ubiquity of the Internet [18]. As IP also meets the mentioned requirements, it is the best choice for the network layer of the smart grid communication stack [8].

Any agent that communicates via IP is able to participate in the smart grid infrastructure regardless of the actual physical medium. Interoperability is ensured, even if different technologies are used in different network segments [8]. Moreover, the upcoming Internet Protocol version 6 (IPv6) further increases the advantages for using IP. For example, IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) based on IEEE 802.15.4 can be used in LANs. In summary, smart grids will benefit from the utilization of IP as it offers the creation of an open, flexible, and secure network. Numerous standards are based on this protocol, and it is widely accepted in industry, business, and society [3].

On top of IP as network layer protocol, Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are identified as proper transport layer protocols [7]. Both alternatives have advantages and disadvantages, and thus the selection depends on the particular application layer. However, as TCP is reliable regarding data transfer and packet ordering as well as connection-oriented, it conforms better to the smart grid requirements.

3) Application layer: In an IP-based protocol stack, several application layer protocols can be used. A set of standardized protocols is identified as appropriate application layer technology. As this is highly application-specific, no universally valid technology is specified in this work. However, a preferably popular technology such as Hypertext Transfer Protocol (HTTP) should be utilized in order to provide a reasonable basis for interoperability. A communication system using instant messaging and presence information can be based on the Extensible Messaging and Presence Protocol (XMPP) [19]. Likewise, the Session Initiation Protocol (SIP), which is commonly used to establish media sessions for Internet telephony or instant messaging, can be used as application layer protocol for smart grid communication [20]. On the other hand, representation of data is important for interoperability. Extensible Markup Language (XML), JavaScript Object Notation (JSON), or Efficient XML Interchange (EXI) is often used for data serialization [7].

2.3.2 Common language

The protocol stack in combination with the grid architecture provides a solid basis for agent communication. Subsequently, these agents need a common and standardized language for data exchange, also known as ACL [10]. A common agreement has to be made to support interoperable end-to-end communication between the agents [4]. Currently, there exist numerous standards addressing different areas in the smart grid. For example, substation automation is targeted by IEC 61850 while IEEE 1547 addresses the power system for DERs [5]. Regarding application-level energy management, IEC standards 61968 and 61970 describe a Common Information Model (CIM) for data exchange with focus on the power distribution and transmission domain [6]. Nevertheless, a common language for data exchange is needed which covers all relevant smart grid domains to enable smart grid mechanisms:

- In the smart grid, the utilized ACL has to be able to handle distribution of price information. In this context, energy retailers or grid operators use mechanisms like day-ahead pricing or TOU pricing to control demand of customers.

- Information concerning energy consumption and energy usage has to be sent between agents. This includes both real-time metering information and energy consumption forecasts for varying time horizons. Also, signals to limit consumption of customers need to be exchanged via the communication system.
- Connections between agents have to be established in order to enable end-to-end communication between them. Thus, a kind of enrollment or registration is necessary.
- The possibility to trade locally produced or stored energy is essential in a smart grid environment. Hence, a tendering mechanism to collect offers and order or rather buy selected offers is required. Energy aggregators use these signals to buy energy for reselling on energy markets. Likewise, grid operators use tendering signals when communicating with customers in order to ensure grid stability by trading flexibilities of energy usage.
- Moreover, agents should be able to notify other smart grid partners about events such as imminent blackouts or power fluctuations.

Smart Energy Profile (SEP) version 2.0 covers most of these requirements such as pricing, DR, load management, and smart metering, but it is focused on communication within residential and commercial customer buildings [7]. Although it has been proposed by ZigBee Alliance, SEP 2.0 is independent of the physical layer and relies on TCP and IP. On the other hand, Open Automated Demand Response (OpenADR) version 2.0 can be used in the proposed MAS as it defines DR and DER signals for communication between energy markets, utilities, operators, or customers [21]. This common language is based on multiple sources like OpenADR 1.0, and it can be used with several application layer technologies like HTTP, SIP, or XMPP. However, OpenADR is a subset of Energy Interoperation (EI) version 1.0 published by the Organization for the Advancement of Structured Information Standards (OASIS) [22]. This standard provides services covering dynamic pricing, reliability, emergency, transactions, and reporting. Thus, EI is identified as most promising and diversified standard for interoperable communication between smart grid agents. Price distribution is realized with the EI *quote service*. The *transactive service* is used for publishing energy consumption information as well as tendering of flexibilities and locally produced energy. The registration of agents is provided by the *enroll service*, and the *event service* offers the possibility to send notifications via the grid. Therefore, EI with its data and communication model is defined as common language for data exchange between smart grid agents.

2.4 Implementation

In this section, an implementation of the proposed smart grid communication system is presented. The work is focused on the interaction at the smart grid interface of customer buildings. The relevant agents, which are realized, are energy retailers, energy

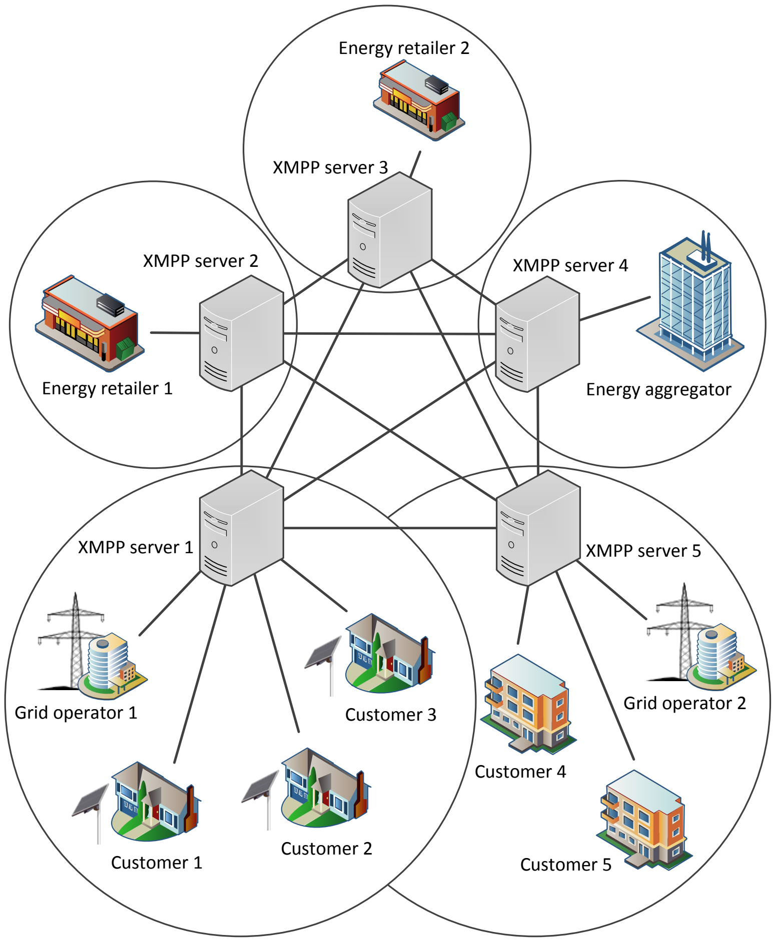

Figure 2.2: Implemented smart grid setting

aggregators, grid operators, and customers. Bulk generators are omitted as they do not communicate directly with the customers. IEEE 802.11 is chosen as network technology in order to reduce wiring effort in the setup. Based on IP and the overlying TCP, the standardized XMPP is utilized as application layer protocol, which is also suggested in [16]. A server-client architecture similar to the World Wide Web is established. Each agent represents a client, which is connected to a server. XMPP is characterized by its decentralized and extensible approach [19]. Communication traffic is spread over various servers, and new servers and agents can be easily integrated into the network. Smart grid operational domains can be covered by distinct XMPP servers to achieve a logical encapsulation and separation.

Figure 2.2 illustrates the implemented setting. The domains are visualized as circles consisting of their associated agents. For example, the domain of XMPP server 1 symbolizes a smart grid segment operated by grid operator 1 comparable to a small neighborhood with a substation. If customer 1 obtains energy from retailer 1, communication between

Listing 2.1: Serialized signal for ordering an energy flexibility offer


```
<pyld:eiCreateTransaction>
 <pyld:requestID>R_GO1_382</pyld:requestID>
 <ei:partyID>GO1</ei:partyID>
 <ei:counterPartyID>C3</ei:counterPartyID>
 <!-- ... -->
 <ei:eiTransaction>
  <ei:transactionID>A_GO1_83</ei:transactionID>
  <ei:tenderID>O_C3_831</ei:tenderID>
  <!-- ... -->
  <emix:product>
   <emix:transactiveState>
    transaction
   </emix:transactiveState>
   <emix:side>buy</emix:side>
   <!-- ... -->
  </emix:product>
 </ei:eiTransaction>
</pyld:eiCreateTransaction>
```


these agents is running via XMPP server 1 and XMPP server 2. Each domain is operated by one distinct agent, e.g. the domain of XMPP server 5 is managed by grid operator 2.

OASIS EI is used as common language for exchange and interpretation of data, which are serialized using XML. In this implementation, all signals are defined and realized that are necessary to meet the previously introduced communication requirements. An example for a transaction signal to order a previously received flexibility offer is shown in Listing 2.1. The transaction signal defines a unique request identifier (`requestID`) as well as sender (`partyID`) and receiver (`counterPartyID`) of the signal. Additionally, this signal refers to the obtained flexibility offer (`tenderID`). In this example, grid operator 1 (`GO1`) wants to buy (`buy`) the previously offered flexibility (`O_C3_831`) from customer 3 (`C3`) in order to maintain grid stability. Therefore, the listed signal is sent from grid operator 1 to XMPP server 1. As both agents are in the domain of XMPP server 1 (see Figure 2.2), the transaction signal is directly forwarded to customer 3 without an intermediate step via another XMPP server. Afterwards, customer 3 will control its local appliances to save the just sold flexibility.

All clients and servers are implemented on separate Raspberry PIs (model B+) resulting in an overall number of 15 Raspberry PI boards. During registration of a client at a server, a unique Jabber Identifier (JID) is defined. In order to enhance reliability, redundant servers may be used, or disconnected clients can use XMPP servers from other domains to maintain connectivity. Based on XMPP, clients are able to communicate via unicast or in chat rooms, comparable to multicast messaging. On the XMPP servers, the free Openfire version 3.9.3[1] is used as server implementation. The agents run on Debian based

[1] www.igniterealtime.org/projects/openfire

Raspbian[2], and the application software uses Java JRE 1.8. XMPP communication on the clients is handled by Smack API[3]. The agents have different application programs in accordance with their objectives and tasks. As the focus of this work is on communication issues, mockup implementations are used. Thus, the agents operate in a simplified way to enhance traceability of smart grid interaction.

2.5 Discussion

The smart grid communication infrastructure assumes several requirements and is faced with many challenges. Thus, this work proposes an approach to meet these requirements in context of DR interaction between customer buildings and other smart grid agents. In a proof-of-concept implementation, the functionality of this approach is demonstrated. This section will discuss the advantages and disadvantages of the design decisions made in the introduced approach.

Availability and reliability are important aspects as power infrastructure is a highly prioritized sector. In addition to adequate protocols and mechanisms, these requirements yearn for appropriate technologies on the hardware level. While wireless systems might have lower installation costs compared to wired alternatives, they offer limited security and performance concerning parameters like bandwidth or data rate [9]. In the presented implementation, wireless technology is used for convenience, but dependable network technologies have to be used for crucial agents and grid segments. Otherwise, these links are not robust enough in case of emergency. Concerning less important grid members, also less reliable wireless technologies can be used resulting in a mixed solution.

Concerning interoperability, the system should be able to operate in various domains from energy generators to customers. There exist a lot of standards addressing individual domains. In the proposed MAS, the internal communication within a substation or a customer building is enclosed within the agent. Independent of the internally used technology, e.g. to exchange metering data between a smart meter and an energy management system, the agents are able to communicate on a higher level. Thus, the heterogeneity of domain-specific standards and technologies is abstracted to provide a homogeneous agent-based communication system. In the context of this work, this is realized by the de facto standards IP and TCP/UDP, which can use the already existing Internet infrastructure.

In the future smart grid, a potentially high number of agents has to be handled by the communication system [9]. Additionally, new services might emerge which have to be integrated. IP-based networks provide the necessary flexibility to be utilized in the smart grid [8]. While deployment and maintenance costs can be reduced, the protocol is independent of the overlying services and applications [2]. Services have to be addressed by the common ACL where communication patterns are defined. OASIS EI already covers a variety of potential smart grid interactions, which is presented in the proof-of-concept

[2] www.raspbian.org
[3] www.igniterealtime.org/projects/smack

implementation using the example of DR communication at the smart grid interface of customer buildings. With increased communication traffic, a compact data representation is required, as well. The signals in the presented implementation are encoded with XML. However, a compressed format like EXI will boost performance [7]. Scalability issues are also tackled by XMPP due to its push approach and the ability of servers to source out message processing [19].

Although the proposed system is capable to be used in the smart grid communication infrastructure, QoS and security must not be disregarded. Security aspects need to be addressed on different levels from the physical medium and the network technology up to end-to-end security in the smart grid applications. Both data security and authorization have to be ensured by applying state-of-the-art security technologies. In [23], a comprehensive approach based on public key infrastructure (PKI) technology is presented in order to tackle smart grid security requirements. With ongoing interconnection, the potential of attacks and incidents against individual agents or the entire grid is increasing [2]. Moreover, security needs to be guaranteed for a long time as power installations have a long life cycle. In addition, QoS measures have to be defined. While security issues are out of scope in this work, QoS is implicitly addressed in the standardized technologies of the proposed approach like IP, TCP, UDP, or XMPP as well as Ethernet, IEEE 802.11, or ZigBee.

2.6 Conclusion

In this work, a communication architecture is presented to enable DR interaction of smart buildings with other smart grid agents like energy retailers or grid operators. The proposed architecture follows a MAS concept. Additionally, an IP-based communication stack is defined to support multiple network technologies on the physical layer as well as different application layer protocols. A common language for data exchange between the smart grid agents covering a set of requirements is evaluated. In a proof-of-concept implementation, the approach is realized on the basis of a smart grid setting using XMPP as application layer protocol and OASIS Energy Interoperation for data exchange. Furthermore, the design decisions of this system are discussed concerning the requirements for smart grid communication presented in literature.

Further steps are the definition of additional communication patterns apart from the customers and a simulation-based evaluation of network performance. For example, the interaction between energy retailers and energy aggregators should be analyzed. Parameters of network technologies and the effect of grid size regarding both number of agents and spatial dimension should be evaluated based on simulations in order to gain detailed knowledge on communication behavior. Outcomes of this simulation should be verified in real-world installations to prove applicability of the proposed approach. Moreover, security has to be integrated to form an overall communication system for critical and sensitive applications. Finally, the compliance of the system with general as well as domain-specific QoS requirements needs to be evaluated.

Acknowledgment

This work was funded and supported by FFG (Austrian Research Promotion Agency) under the *BProfEM* project (P845627) and by Siemens AG Austria under the *XMPP Demonstrator* project.

References

[1] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information", *Energy and Buildings*, vol. 40, no. 3, pp. 394–398, 2008, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2007.03.007.

[2] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges", *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 5–20, First Quarter 2013, ISSN: 1553-877X. DOI: 10.1109/SURV.2012.021312.00034.

[3] V. K. Sood, D. Fischer, J. M. Eklund, and T. Brown, "Developing a Communication Infrastructure for the Smart Grid", in *IEEE Electrical Power & Energy Conference*, Oct. 2009, pp. 1–7. DOI: 10.1109/EPEC.2009.5420809.

[4] H. Farhangi, "The Path of the Smart Grid", *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, Jan.-Feb. 2010, ISSN: 1540-7977. DOI: 10.1109/MPE.2009.934876.

[5] W. Wang, Y. Xu, and M. Khanna, "A survey on the communication architectures in smart grid", *Computer Networks*, vol. 55, no. 15, pp. 3604–3629, Oct. 2011, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2011.07.010.

[6] V. C. Güngör, D. Sahin, T. Kocak, S. Ergüt, C. Buccella, C. Cecati, and G. P. Hancke, "Smart Grid Technologies: Communication Technologies and Standards", *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 529–539, Nov. 2011, ISSN: 1551-3203. DOI: 10.1109/TII.2011.2166794.

[7] R. Kyusakov, J. Eliasson, J. van Deventer, J. Delsing, and R. Cragie, "Emerging Energy Management Standards and Technologies - Challenges and Application Prospects", in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2012, pp. 1–8. DOI: 10.1109/ETFA.2012.6489674.

[8] F. Lobo, A. Cabello, A. Lopez, D. Mora, and R. Mora, "Distribution network as communication system", in *CIRED Seminar SmartGrids for Distribution*, Jun. 2008, pp. 1–4. DOI: 10.1049/ic:20080448.

[9] V. C. Güngör and F. C. Lambert, "A survey on communication networks for electric system automation", *Computer Networks*, vol. 50, no. 7, pp. 877–897, May 2006, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2006.01.005.

[10] M. E. Collins, R. W. Silversides, and T. C. Green, "Multi-Agent System Control and Coordination of an Electrical Network", in *International Universities Power Engineering Conference*, Sep. 2012, pp. 1–5. DOI: 10.1109/UPEC.2012.6398641.

[11] Z. Wang, R. Paranjape, A. Sadanand, and Z. Chen, "Residential demand response: An overview of recent simulation and modeling applications", in *IEEE Canadian Conference on Electrical and Computer Engineering*, May 2013, pp. 1–6. DOI: 10.1109/CCECE.2013.6567828.

[12] E. H. Et-Tolba, M. Maaroufi, and M. Ouassaid, "Demand side management in smart grid by multi-agent systems technology", in *International Conference on Multimedia Computing and Systems*, Apr. 2014, pp. 1042–1045. DOI: 10.1109/ICMCS.2014.6911211.

[13] Z. M. Fadlullah, M. M. Fouda, N. Kato, A. Takeuchi, N. Iwasaki, and Y. Nozaki, "Toward Intelligent Machine-to-Machine Communications in Smart Grid", *IEEE Communications Magazine*, vol. 49, no. 4, pp. 60–65, Apr. 2011, ISSN: 0163-6804. DOI: 10.1109/MCOM.2011.5741147.

[14] C. Yilmaz, S. Albayrak, and M. Lützenberger, "Smart Grid Architectures and the Multi-Agent System Paradigm", in *International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies*, Apr. 2014, pp. 90–95.

[15] S. Kahrobaee, R. A. Rajabzadeh, L.-K. Soh, and S. Asgarpoor, "A Multiagent Modeling and Investigation of Smart Homes With Power Generation, Storage, and Trading Features", *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 659–668, Jun. 2013, ISSN: 1949-3053. DOI: 10.1109/TSG.2012.2215349.

[16] M. Pichler, "Communication Patterns for Demand Side Flexibility", in *Symposium on Communications for Energy Systems*, Sep. 2014, pp. 19–28.

[17] P. Palensky and D. Dietrich, "Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads", *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381–388, Aug. 2011, ISSN: 1551-3203. DOI: 10.1109/TII.2011.2158841.

[18] J. G. Cupp and M. E. Beehler, "Implementing Smart Grid Communications", *Burns & McDonnell TECHBriefs*, no. 4, pp. 5–8, 2008.

[19] P. Saint-Andre, K. Smith, and R. Tronçon, *XMPP: The Definitive Guide*, First Edition. O'Reilly Media, 2009, ISBN: 9780596521264.

[20] A. B. Johnston, *SIP: Understanding the Session Initiation Protocol*, Third Edition. Artech House, 2009, ISBN: 9781607839958.

[21] OpenADR Alliance, *OpenADR 2.0 Profile Specification, B Profile*, Jan. 2013.

[22] OASIS, *Energy Interoperation Version 1.0, OASIS Standard*, Jun. 2014.

[23] A. R. Metke and R. L. Ekl, "Security Technology for Smart Grid Networks", *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 99–107, Jun. 2010, ISSN: 1949-3053. DOI: 10.1109/TSG.2010.2046347.

CHAPTER 3

Interoperable integration of building automation systems using RESTful BACnet Web services

Abstract: Building automation domain is affected by a diversity of standards and technologies. With the upcoming Internet of Things heading for a pervasive network of interconnected smart things and the need for efficient and intelligent energy management systems, the necessity of integrating these heterogeneous building automation environments soars. Thus, standardized, interoperable, secure, and scalable solutions are required. Well-established Web service technologies based on the Internet Protocol act as key enabler to realize this vision. In this work, an approach for the seamless and interoperable integration of building automation systems based on RESTful BACnet/WS is presented. In order to ease the integration process, the BACnet/WS specification is described as formal, machine-readable object model. Additionally, most common building blocks of building automation systems including logical as well as physical resources are specified in the form of type definitions to unify integration. Furthermore, a proof-of-concept implementation of a BACnet/WS server is realized in order to illustrate the functional capability of the presented approach.

The content of this chapter has been published: D. Schachinger, C. Stampfel, and W. Kastner, "Interoperable integration of building automation systems using RESTful BACnet Web services", in *Annual Conference of the IEEE Industrial Electronics Society*, Nov. 2015, pp. 3899–3904.

3.1 Introduction

During the past decades, a wide range of standards and technologies has been evolved in the domain of building automation (BA). Cost optimization, energy reduction, increased comfort, and security as well as safety are main goals that are addressed by these developments [1]. Traditionally, building automation systems (BASs) are faced with these requirements in an enclosed environment such as a single residential or commercial building. With the upcoming paradigm of the Internet of Things (IoT), the borders of these BA islands are more and more softened resulting in the vision of a pervasive, global network of interconnected and seamlessly integrated things. Moreover, a high degree of BAS integration is needed in order to support efficient energy management systems (EMSs) matching statutory regulations regarding energy consumption and emission [2]. However, a set of basic problems has to be tackled in order to realize the idea of integration. Besides security and privacy concerning user data, interoperability and scalability are identified as major challenges. The prevailing heterogeneity in BA technologies and standards acts as an inhibitor with respect to service integration. Thus, seamless integration of manifold BASs into a common environment as an EMS or the IoT needs an interoperable approach enabling uniform communication and homogeneous representation of information [3].

Service-oriented architecture (SOA) is identified as promising concept to solve the integration problem by combining both autonomy and interoperability [4]. Technologies based on the Internet Protocol (IP) provide a standardized basis for SOA in order to establish an integration of BASs into information technology (IT) systems and enterprise applications as they are already well-established at the BA management level [3]. In SOA, services are focused on business processes with a loose and stateless coupling of the communicating entities. Therefore, SOA represents a flexible and adaptable concept which introduces a certain level of abstraction while reducing development costs [4]. Web services (WSs) follow this SOA approach offering platform-agnostic interfaces to interconnect heterogeneous systems [4]. Seamless and real-time integration of different BA technologies as well as interoperability between enterprise applications and BASs can be realized by these self-contained, self-describing, and modular WSs [5]. In general, it can be distinguished between a Representational State Transfer (REST) and a WS-* architecture [6]. According to WS-*, services usually use the Simple Object Access Protocol (SOAP) in combination with the Hypertext Transfer Protocol (HTTP), and additional mechanisms for addressing, security, or discovery are provided. On the other hand, REST is primarily focused on resources which are identified using uniform resource identifiers (URIs). Only a small set of verbs is used to interact with these resources via the WS interface. WS-* offers advanced security concepts and is better suited for business integration due to the higher level of abstraction. Nonetheless, the advantages of the lightweight REST compared to WS-* are a better scalability, ease of use, and the more intuitive resource-orientation [6].

General requirements for integration technologies are platform independence, easy extension, open standardization, and utilization of IP at the field level [7]. As there are still non-IP devices in BASs, gateways are a reasonable choice to avoid high costs for replacing legacy devices. Several technologies have emerged that enable BAS integration using application layer gateways with WS interfaces, e.g. OPC Unified Architecture (OPC UA) [8], Open Building Information Exchange (OBIX) [9], or BACnet Web services (BACnet/WS) [10]. Currently, OPC UA uses WSs based on SOAP as well as a binary protocol based on the Transmission Control Protocol (TCP). Its information model enables the definition of comprehensive BA information. OBIX, on the other hand, defines a concise and generic information model and specifies protocol bindings for SOAP and REST. BACnet/WS in its first version standardizes a SOAP-based WS interface, which is not limited to BACnet but can be used with other technologies, as well [11]. Related work already addresses approaches for OPC UA [12], BACnet/WS [11], and OBIX [7, 13]. This paper is focused on seamless and interoperable BAS integration into IT systems based on REST as this concept is identified as most compliant to the resource-orientation of BASs. Therefore, the proposed extension of BACnet/WS defining a RESTful WS interface and an abstract object model tailored to the BA domain is utilized [14]. Automatic processing and interpretation of information is enabled by introducing a machine-readable representation of the BACnet/WS object model, which is further enhanced by type definitions for common BAS building blocks. The potential of this novel BACnet/WS interface as an alternative to existing WS-* as well as REST architectures is highlighted in a proof-of-concept implementation.

3.2 RESTful BACnet Web services

In 2006, the American Society of Heating, Refrigerating, and Air-Conditioning Engineers (ASHRAE) published BACnet/WS as Addendum c to ANSI/ASHRAE standard 135-2004 specifying a WS interface for integrating BASs into enterprise applications [10]. The defined SOA uses SOAP with HTTP and provides a simple but non-extensible object model for the definition of nodes and attributes. In the proposed Addendum am to ANSI/ASHRAE standard 135-2012, this interface is replaced by RESTful services [14]. In 2014, the first public review of RESTful BACnet/WS was started.

3.2.1 Web service interface

A WS interface for integrating BASs requires a set of system functions. Besides basic services for remote reading and writing of resources, advanced services are needed for error handling or alarming. In order to provide an interoperable integration, existing BA standards and technologies have to be supported by the WS-based integration platform. Additionally, standardized IT security mechanisms have to be supported, and the connectivity to the BAS has to be ensured [5].

In contrast to the initial BACnet/WS publication, the new service interface follows the REST approach with an HTTP binding that uses only a small set of verbs to communicate with resources of the underlying BAS. Therefore, a migration for matured SOAP-based services is provided in the proposed extension of BACnet/WS. The fixed set of service operations for accessing attributes is replaced by standardized operations to read and write any kind of structured data, invoke operations, or create and delete resources. These basic services are complemented by services for filtering items, history logs, and subscriptions. Thus, more complex interaction with the BAS is provided. Both past and future data changes can be addressed by either analyzing the data history of resources or subscribing for notification of data changes. An intelligent search is enabled using the filtering mechanism where data and metadata information can be used to restrict result sets. Moreover, different data representations based on Extensible Markup Language (XML), JavaScript Object Notation (JSON), JSON with padding (JSONP), or plain text are supported. Regarding security of user data, RESTful BACnet/WS defines mechanisms for secure communication using Transport Layer Security (TLS). The specified security measures enable the composition of secure application domains. Authorization based on OAuth permits access restrictions for different security scopes. Thus, protecting sensitive user data can be ensured by means of this extended WS interface.

In summary, the REST extension of BACnet/WS is less restrictive than the previous SOAP-based interface. Intuitive handling of resources known from traditional Web interaction eases the seamless integration of BASs. As BACnet/WS can be combined with any BA technology, it provides a platform for interoperability between heterogeneous BASs.

3.2.2 Object model

Although a well-defined WS interface is necessary for a standardized communication of BASs with enterprise applications or the IoT, interoperability in such a heterogeneous domain has to be addressed at the information level. The structure and representation of data as well as the semantics need to be specified in a uniform way to avoid the overhead of many-to-many technology mappings [12].

Such a technology-independent object model to define data and metadata of BASs is provided by BACnet/WS. While the object models of OBIX and OPC UA are extensible, BACnet/WS (Addendum c, 135-2004) provides only a simple and fixed object model with limited capabilities to add semantical information [11]. However, this object model has been enhanced to enable the description of comprehensive semantics in the proposed WS interface. Moreover, a machine-readable representation of the BACnet/WS object model is essential for automatic and computer-aided integration of BASs. As the BACnet/WS extension describes the abstract object model only textually, this paper aims at defining an explicit, machine-readable model comprising concepts for designing data and metadata. This is the first step for modeling BASs in order to integrate them into an interoperable, WS-based gateway.

In general, BACnet/WS distinguishes between *data* and *metadata*. The basic building block for data, which is intended to hold the value information, is called data item (*DataItem*). Metadata, on the other hand, are used to model non-value information. A list of base types for primitive as well as constructed data is provided. Furthermore, a common set of metadata for all data items and specific metadata for particular base types are defined. In the following, the core elements for data and metadata are described, and their composition into a consistent, machine-readable object model is presented.

1) Data: For the purpose of modeling data, all base types become separate classes in the object model. While primitive data items such as *Real*, *Time*, or *String* directly hold the particular value, there exist constructed data items such as *Collection*, *Object*, or *Sequence* that are containers for subordinated items. These primitive and constructed data items provide a suitable set of modeling alternatives for the description of BAS resources, their internal structure, and their relationships. If the type of a data item is not known at design time, *Any* can be used instead. Besides individually defined data items based on this object model, BACnet/WS provides a set of standard data items. These standard entry points are used to publish type definitions (*.definitions*), manage subscriptions (*.subscriptions*), or access logical BAS structures (*.trees*). In addition, information about the server (*.sysinfo*) and the authorization (*.auth*) is provided.

2) Metadata: The data items and their values have to be enhanced by additional semantic information. Thus, a client accessing integrated BASs is able to interpret values and understand the logical structure of resources. This non-value information is primarily modeled as attributes of the abstract *DataItem* and its inherited classes. For identification of data items, the metadata *name* and *id* are used. Moreover, the name of a data item is utilized to create its URI as unique path within the BACnet/WS server. On the other hand, metadata to express dependencies and relationships between data items are modeled as references, e.g. *extends*, *children*, or *units*. While the standard defines these cross reference metadata as simple string values, the developed object model enables a more accurate regulation of relationships using associations between the data item classes. The metadata *links* of a data item can host a list of links to other data items or even remote resources on other servers defined as *Link*. For a detailed description of data items, the *tags* metadata can be used to assign predefined *Tag* instances. In order to invoke operations on data items, BACnet/WS provides the concept of a *Function*, which is added as child element to other data items and defines a set of input *parameters* as well as the data it *returns*. Moreover, multilingual texts can be modeled using the concept of *LocalizableMetadataItem*.

Figure 3.1 illustrates the machine-readable interpretation of the BACnet/WS object model (Addendum am, 135-2012). It has to be noted that only the core elements are shown in the figure in order to keep a certain level of clarity and readability. Furthermore, this model is intended for internal representation of the integrated BAS than for serialization purposes. However, most concepts can be directly converted to an output format in XML or JSON. In summary, RESTful BACnet/WS is a good alternative compared to other WS-based integration technologies for BASs as it combines both a lightweight REST interface, which is well-known in Web communication, and a generic yet detailed object model for semantically enriched modeling of BAS resources. Different data representations and

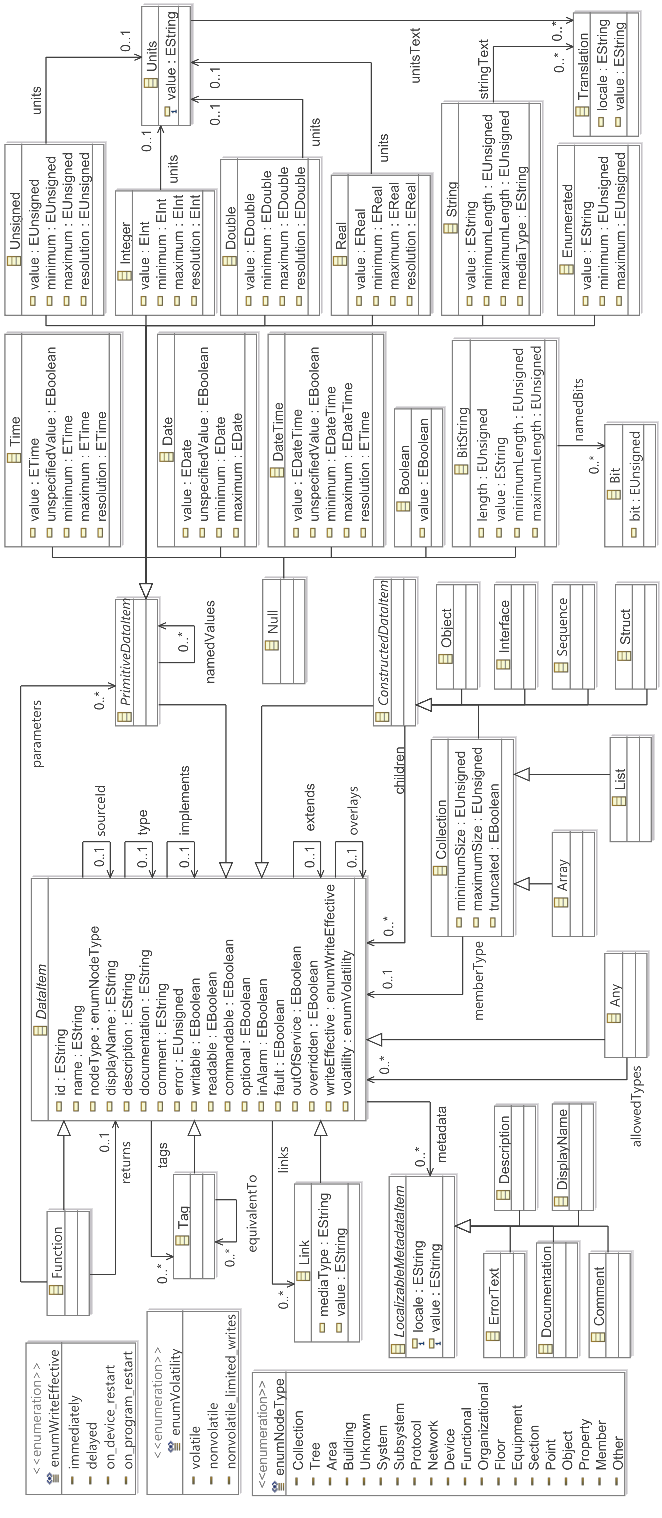

Figure 3.1: BACnet/WS core object model

the ability to extend the object model by own types ease a large-scale application of the proposed BACnet/WS standard addendum in the domain of BAS integration.

3.3 Building automation system integration

Interoperability between different BA technologies can be achieved by means of standardized data representation that is based on the BACnet/WS object model. Protocol-specific communication details can be hidden by the BACnet/WS server, and thus IT applications are able to access the underlying BAS via the common and well-known RESTful WS interface. Although BAS integration can be based immediately on the object model, an additional set of global type definitions serves as intermediate layer for describing advanced building blocks of BASs. Such standard building blocks take advantage of already available modeling concepts in order to describe more complex components of the BA domain.

As depicted in Figure 3.2, BAS type definitions are defined in the context of the general BACnet/WS object model. Subsequently, instantiation of BAS resources in the course of integration is mainly based on the domain of these predefined types. However, individual modeling using the basic BACnet/WS object model is still possible, which is graphically illustrated by the overlapping areas. An advantage of BACnet/WS is the machine-readable publication of type definitions in the so-called definition context. Thus, creation of BAS resources as well as access operations via the WS interface can rely on these types, which eases automatic interpretation and processing of information.

The definition of a universal information model for the integration of BASs is a challenging task due to the heterogeneity of existing BA technologies and standards. However, a set of core components can be identified, which are found in most common protocols. First, aggregation is used to incorporate information from different technologies into these components. Second, abstraction from technology-specific details leads to a universally applicable representation of the core elements. Thus, an independent and general description of BASs for the purpose of interoperable integration is enabled. In general, at least three main concepts for the construction of BASs can be found in prevailing technologies [15]. In the following, BAS type definitions are derived from these concepts.

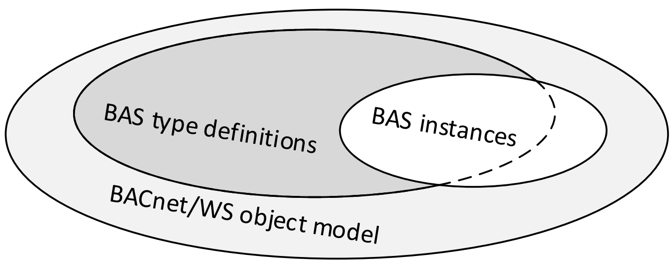

Figure 3.2: Modeling domains

1) Devices are the physical entities of BASs. Information about the manufacturer, product-specific metadata (e.g. order number), and installation-specific data (e.g. device address) are encapsulated within a device instance. Thus, a device is described as complex BACnet/WS *Object* containing the additional information in the form of child elements of node type *Property*. Furthermore, a device acts as aggregation of functional endpoints of the BAS interworking that are known as datapoints. Therefore, the type definition of a device also consists of an unordered *List* of related datapoints. Members of this list are limited to elements of the datapoint type.

2) Datapoints host the actual functionality of the BAS. BACnet/WS provides the special node type *Point* to model these communication endpoints. However, datapoints are not only atomic data items but comprise extra information like priority. Hence, a datapoint type should be modeled as constructed *Object* holding the corresponding set of values as well as the extra information in subordinated child elements. The actual value data items are of node type *Point* in order to indicate their special characteristics.

3) Views integrate devices or datapoints into logical structures. Depending on the intended integration use case, different hierarchies and lists can be built with the generic view concept. The view type is derived from the base type *Object* and has lists for subordinated elements of type view, device, and datapoint. Thus, any kind of topological, functional, or geographical structure can be defined based on this view type concept. For example, a building structure consisting of several building parts, floors, and rooms can be designed. Links between building elements and particular devices or datapoints indicate the area of influence (e.g. a light switching actuator operating in meeting room on the 5th floor).

In the scope of this work, the presented concepts are formulated as machine-readable types in conformance with the BACnet/WS object model to provide a solid basis for subsequent BAS integration. Additionally, supportive information such as enumerations (e.g. priority, building part types) is published in the definition context. Thus, a reasonable and dynamic type set is formed. Anyone familiar with the language of BACnet/WS, which is defined in the corresponding object model, is able to interpret these definitions. In order to give a brief overview of the definition context, Figure 3.3 presents some simplified type declarations. The types, which can be instantiated, are colored orange. First, the *device* type is shown including the previously mentioned children. Then, the *datapoint* type is illustrated specifying the generic datapoint structure. As an example, the *temperature* type extends this generic datapoint type by adding a data item of base type *Real* representing the current temperature value of this datapoint. Finally, an exemplary enumeration is modeled to clarify the creation and usage of supportive information within the definition context. When integrating existing BASs into a BACnet/WS server, the resources and their logical as well as physical relationships are instantiated using the defined types. If necessary, also technology-specific details can be added by either extending these types or using default BACnet/WS modeling concepts.

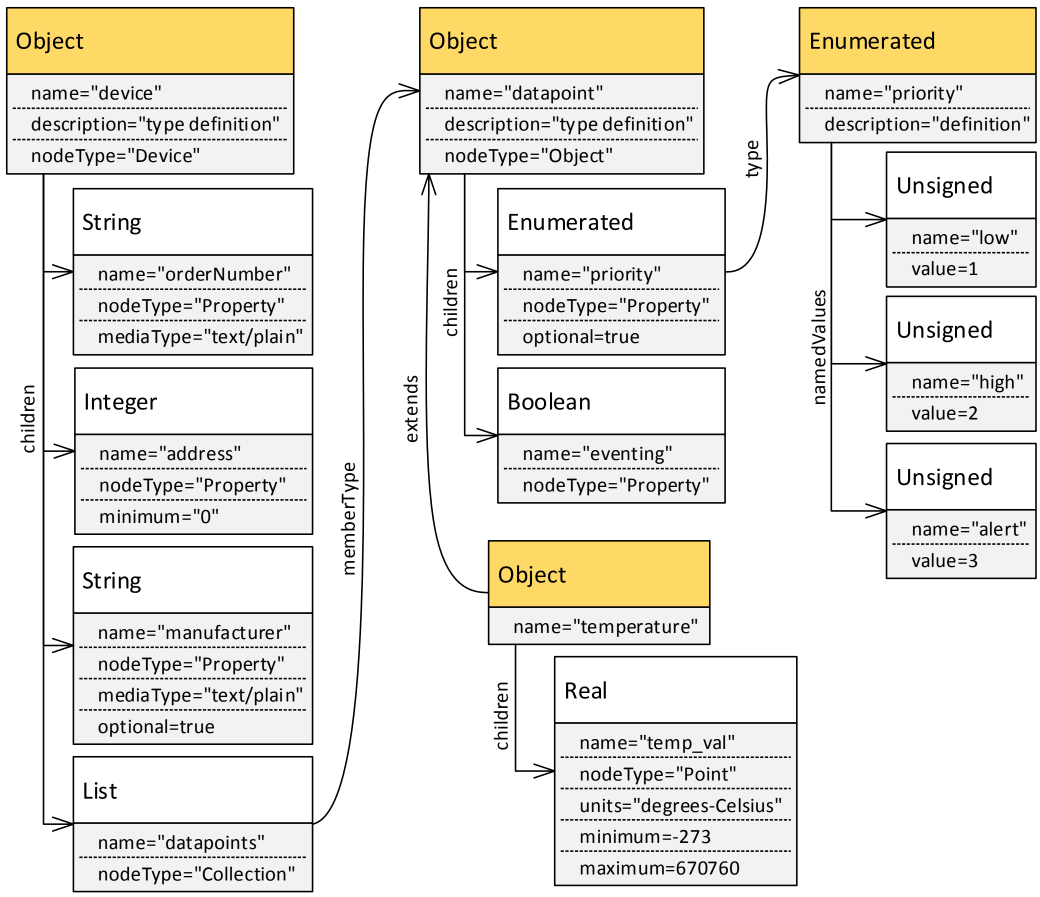

Figure 3.3: Datapoint and device in the BACnet/WS definition context

3.4 Server implementation

In this work, the RESTful BACnet/WS concept is realized in terms of a proof-of-concept implementation. The open source Web server Apache Tomcat is used as servlet container for the Java-based server implementation. Beyond Tomcat and the Java Servlet API, the Spring framework is utilized to handle dependencies between the various software packages.

Similar to other gateway solutions, the BACnet/WS server is located between BASs and remote clients. Depending on the dimension of the BAS, the BACnet/WS server is able to handle residential buildings with only a few datapoints as well as commercial buildings with vast amount of devices and datapoints. Various communication media can be used to establish a link between the BACnet/WS server and the BAS. For each BA technology, a corresponding technology adapter is required to provide interoperability at the server level. The remote clients interact with the server via synchronous or asynchronous request-response services.

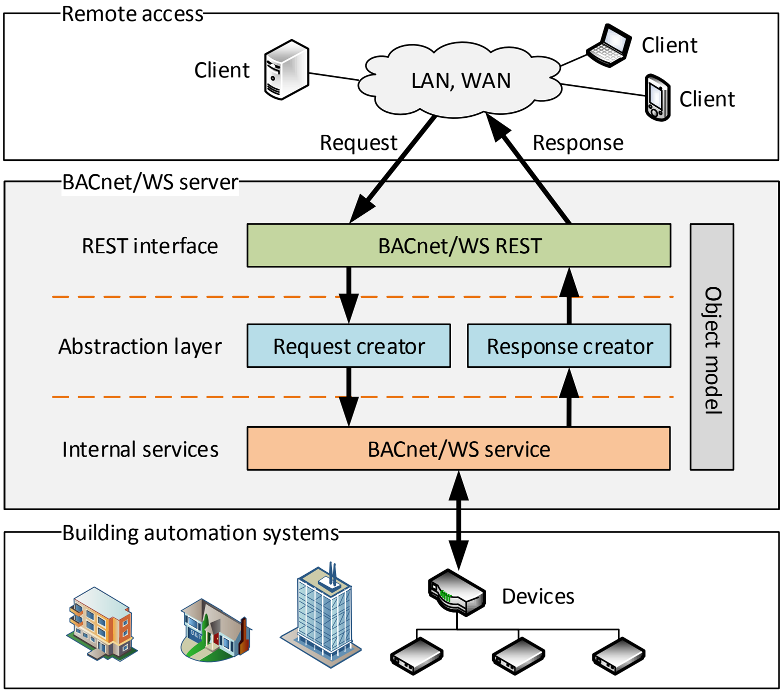

Figure 3.4: BACnet/WS server architecture

In order to process requests (e.g. read requests, service subscriptions, write attempts), the BACnet/WS server is based on a three-layer architecture. The REST interface is responsible for receiving messages from clients and sending messages to clients. The REST operations are implemented in this server layer. As different data representations (e.g. XML) are allowed in BACnet/WS, an intermediate abstraction layer is defined to transform content data. The request creator translates incoming data into an internal representation (i.e. unmarshaling). On the other hand, the response creator constructs the output message for transmission regarding the desired data representation (i.e. marshaling). The internal services host the actual BACnet/WS functionality. Additionally, interaction with the integrated BAS is realized in this lowermost layer. A framework for the BACnet/WS object model supporting the three operative layers completes the server architecture (see Figure 3.4).

At runtime, the server dynamically loads the type definitions that are available in BACnet/WS object model syntax. In addition to predefined types described in the proposed standard, these type definitions form the syntactical and semantical basis for subsequent integration of BASs. Remote clients can access the definition context of the server via the standard data item *.definitions*. Afterwards, configuration information of the BASs is imported describing their structure as well as the physical and logical resources. This information is made available via the standard data item *.trees*.

The developed proof-of-concept implementation supports basic interaction with the integrated BASs via the standardized HTTP WS interface. For data representation, XML serialization is used, and the presented BACnet/WS core object model is fully supported. Logical views and physical devices can be browsed, and values of datapoints are readable and writable. Furthermore, the definition context is published to enable automatic type processing. Rudimentary filtering on simple metadata is implemented in order to provide enhanced search capabilities. Finally, subscription to change of values for primitive data items is realized. Thus, clients can register on particular resources to receive update information.

3.5 Discussion and evaluation

This section will discuss the characteristics of this approach regarding major issues for BAS integration, and the functionality of the proof-of-concept implementation is evaluated.

First, interoperability is of utmost importance in order to enable interconnection of various BA technologies and IT systems. WSs on top of well-established, IP-based communication are identified as most promising solution to build a technology-independent infrastructure of loosely coupled communication partners. Additionally, interoperability on the information level has to be ensured to achieve real technology-independence [12]. BACnet/WS defines a comprehensive as well as extensible object model and uses a well-suited RESTful WS interface for BAS integration. Hence, BACnet/WS perfectly matches the required interoperability demand.

Regarding efficiency and performance, it can be distinguished between one-time configuration effort and continuous runtime service capability. In the scope of this work, a KNX test bed consisting of several sensors and actuators is used to evaluate the developed approach. Data from the Engineering Tool Software (ETS) of KNX are used in the integration process, and thus configuration of the server can be done in an efficient way. Although the complexity of the integrated BA technology influences the total configuration time, this step is basically linear in the size of the integrated BAS. Runtime performance, on the other hand, heavily depends on the communication protocol and the message representation. While performance analysis for SOAP-based BACnet/WS exist [16], the RESTful interface is not addressed in literature, yet. Evaluated with an OBIX server, the Constrained Application Protocol (CoAP) leads to smaller message sizes while HTTP supports a better throughput [13]. However, ASHRAE only provides an HTTP binding for BACnet/WS. Thus, transmission performance comparisons are limited to different message encodings. As XML and JSON have a high overhead of additional structuring data, a binary or compressed data representation like Efficient XML Interchange (EXI) would lead to significantly smaller message sizes [13].

In addition to the performance bottleneck of WS-based communication, resource demand of BACnet/WS needs to be addressed especially for constrained devices. The presented proof-of-concept implementation runs on a standard Raspberry PI (model B+). Response times during evaluation with the KNX test bed are in the range of common Web communication. It is even possible to equip a field device, which is characterized by

Listing 3.1: Server response to a read request of a temperature datapoint

```xml
<Object displayName="Temperature, Channel A" id="DP-034E"
        name="temp_channel_a" type="temperature">
  <Enumerated name="priority" value="low" type="priority"/>
  <Boolean name="eventing" value="false"/>
  <Real name="temp_val" value="25.92" nodeType="Point"/>
</Object>
```

limited memory and computational power, with a BACnet/WS implementation by downgrading the supported features to pure device functionality. This would lead to a direct integration of single BA devices realizing the vision of the IoT. However, gateway solutions are still necessary for heavily constrained devices and legacy systems.

The functional capability of the proposed integration approach is evaluated with a KNX test bed using Calimero as KNX technology adapter. A series of HTTP requests is sent to the server to analyze its functionality. Amongst others, reading of a temperature datapoint is tested with an HTTP *GET* call including the URI of the particular datapoint resource. In the KNX test bed, the temperature datapoint *temp_channel_a* is part of a temperature sensor *temp*, which is located in the view *sensing*. Listing 3.1 shows the content of the server response after sending a read request to the relative path *views/sensing/devices/temp/datapoints/temp_channel_a*. As illustrated, the temperature channel is of datapoint type *temperature* that is specified in the definition context (see Figure 3.3). The current value is located in the child element named *temp_val* indicating an ambient temperature of $25.92°C$. For convenience, unimportant attributes are omitted in this listing.

Similar to OPC UA, RESTful BACnet/WS defines security mechanisms while OBIX and SOAP-based BACnet/WS only refer to TLS [11]. However, security features are not yet realized in the proof-of-concept implementation. An interoperable BAS integration requires also a high degree of scalability. The introduced WS interface based on standardized Web technologies satisfies this need. Furthermore, improved scalability can be realized by a hierarchical topology of servers [7]. All in all, the presented approach based on RESTful BACnet/WS describes a promising solution for the interoperable integration of BASs.

3.6 Conclusion

Seamless and interoperable integration of BASs into the IoT or enterprise applications such as EMSs is faced with many challenges [5, 7, 13]. Thus, this work proposes an approach to meet these requirements by utilizing RESTful BACnet/WS [14]. A lightweight WS interface supporting basic read and write operations as well as advanced services for history logs, notification subscription, or authorization is defined by BACnet/WS. Moreover, the object model supporting a fine-grained description of BASs is converted into a machine-readable form. For the purpose of an interoperable integration of BASs, additional type definitions for advanced building blocks of BASs are specified in this work. In a proof-of-concept implementation, a BACnet/WS server is developed and evaluated by integrating a KNX test bed. The presented approach is discussed regarding requirements listed in literature concerning interoperable BAS integration.

Further steps are the extension of the proof-of-concept implementation and an extensive evaluation of runtime performance. Security measures and history logging have to be embedded, and subscription and filtering mechanisms need to be enhanced. Performance of the integration approach should be analyzed using different Web protocols, BA technologies, data representations, and hardware platforms. Finally, this approach should be incorporated into the model-driven integration workflow presented in [15] in order to fully automate BAS integration using RESTful BACnet/WS.

Acknowledgment

This work was funded by FFG (Austrian Research Promotion Agency) under the *BProfEM* project (P845627).

References

[1] D. Dietrich, D. Bruckner, G. Zucker, and P. Palensky, "Communication and Computation in Buildings: A Short Introduction and Overview", *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3577–3584, Nov. 2010, ISSN: 0278-0046. DOI: `10.1109/TIE.2010.2046570`.

[2] S. Soucek and G. Zucker, "Current developments and challenges in building automation", *Elektrotechnik und Informationstechnik*, vol. 129, no. 4, pp. 278–285, Jun. 2012, ISSN: 0932-383X. DOI: `10.1007/s00502-012-0013-4`.

[3] W. Kastner, M. Kofler, M. Jung, G. Gridling, and J. Weidinger, "Building Automation Systems Integration into the Internet of Things: The IoT6 approach, its realization and validation", in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2014, pp. 1–9. DOI: `10.1109/ETFA.2014.7005197`.

[4] F. Jammes and H. Smit, "Service-Oriented Paradigms in Industrial Automation", *IEEE Transactions on Industrial Informatics*, vol. 1, no. 1, pp. 62–70, Feb. 2005, ISSN: 1551-3203. DOI: `10.1109/TII.2005.844419`.

[5] J. Bai, Y. Hao, and G. Miao, "Integrating Building Automation Systems based on Web Services", *Journal of Software*, vol. 6, no. 11, pp. 2209–2216, Nov. 2011, ISSN: 1796-217X. DOI: `10.4304/jsw.6.11.2209-2216`.

[6] D. Guinard, I. Ion, and S. Mayer, "In Search of an Internet of Things Service Architecture: REST or WS-*? A Developers' Perspective", in *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, A. Puiatti and T. Gu, Eds., vol. 104, Springer Berlin Heidelberg, 2012, pp. 326–337, ISBN: 9783642309724. DOI: 10.1007/978-3-642-30973-1_32.

[7] H. Järvinen, A. Litvinov, and P. Vuorimaa, "Integration Platform for Home and Building Automation Systems", in *IEEE Consumer Communications and Networking Conference*, Jan. 2011, pp. 292–296. DOI: 10.1109/CCNC.2011.5766475.

[8] OPC Foundation, *OPC Unified Architecture Specification*, Feb. 2009.

[9] OASIS, *OBIX Version 1.1, Committee Specification Draft 02 / Public Review Draft 02*, Dec. 2013.

[10] ASHRAE, *A Data Communication Protocol for Building Automation and Control Networks (ANSI/ASHRAE Addendum c to ANSI/ASHRAE Standard 135-2004)*, Oct. 2006.

[11] W. Kastner and S. Szucsich, "Accessing KNX networks via BACnet/WS", in *IEEE International Symposium on Industrial Electronics*, Jun. 2011, pp. 1315–1320. DOI: 10.1109/ISIE.2011.5984349.

[12] W. Granzer and W. Kastner, "Information Modeling in Heterogeneous Building Automation Systems", in *IEEE International Workshop on Factory Communication Systems*, May 2012, pp. 291–300. DOI: 10.1109/WFCS.2012.6242577.

[13] M. Jung, J. Weidinger, W. Kastner, and A. Olivieri, "Building automation and smart cities: An integration approach based on a service-oriented architecture", in *IEEE International Conference on Advanced Information Networking and Applications*, Mar. 2013, pp. 1361–1367. DOI: 10.1109/WAINA.2013.200.

[14] ASHRAE, *Proposed Addendum am to Standard 135-2012, BACnet - A Data Communication Protocol for Building Automation and Control Networks*, Apr. 2014.

[15] D. Schachinger and W. Kastner, "Model-Driven Integration of Building Automation Systems into Web Service Gateways", in *IEEE World Conference on Factory Communication Systems*, May 2015, pp. 1–8. DOI: 10.1109/WFCS.2015.7160561.

[16] C. Ninagawa, T. Sato, and Y. Kawakita, "Communication Performance Simulation for Object Access of BACnet Web Service Building Facility Monitoring Systems", in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2008, pp. 701–704. DOI: 10.1109/ETFA.2008.4638473.

CHAPTER 4

Modeling framework for IoT integration of building automation systems

Abstract: Advancements within the Internet of Things are leading to a pervasive integration of different domains including also building automation systems. As a result, device functionality becomes available to a wide range of applications and users outside of the building automation domain. In this context, Web services are identified as suitable solution for machine-to-machine communication. However, a major requirement to provide necessary interoperability is the consideration of underlying semantics. Thus, this work presents a universal framework for tag-based semantic modeling and seamless integration of building automation systems via Web service-based technologies. Using the example of the KNX Web services specification, the applicability of this approach is pointed out.

Keywords: Building automation systems, Internet of Things, system integration, semantic modeling, model transformation

The content of this chapter has been published: D. Schachinger, A. Fernbach, and W. Kastner, "Modeling framework for IoT integration of building automation systems", *at - Automatisierungstechnik*, vol. 65, no. 9, pp. 630–640, Sep. 2017.

4.1 Introduction

Application domains in the Internet of Things (IoT) are manifold, including areas like cyber-physical production systems, traffic management, smart infrastructure, supply chains, logistics, healthcare, or social applications [1]. As part of the smart infrastructure, a trend towards intelligent homes and buildings presumes a high degree of device integration and connection in order to cope with more complex application scenarios and enhanced user needs. The aim is a homogeneous communication of building automation systems (BASs) with services and applications residing inside the IoT, which is independent of the underlying technologies.

Atzori et al. describe the IoT as intersection of three visions, which can be recognized also when integrating BASs [2]. First, the overlap with the things-oriented vision is obvious as the focus is on sensors, actuators, and controllers of building service equipment that need to be connected and should communicate with each other. Second, the Internet-oriented vision symbolizes the bridge between traditionally enclosed BASs and their couplings to a wide area network like the Internet. In this context, Web services (WSs) are identified as promising technology to abstract from heterogeneous protocols in the building automation (BA) domain and force IoT integration of BASs [3]. Third, the semantic-oriented vision needs to be considered as it is a key factor for interoperability due to the heterogeneity in the field of BA as well as in IT systems. Therefore, a common information representation needs to be established [4]. Based on this, concepts to model the underlying semantics are required in order to allow the automatic interpretation of this information for machine-to-machine communication. IoT integration demands machine-readable semantics of the built environment, the device functionality, or the technology characteristics.

In this article, a framework is presented that supports a seamless integration of BASs into the IoT in accordance with these three visions. Starting point of the framework is a set of requirements (Section 4.2). A universal modeling concept is the basis for a model-driven integration process (Section 4.3). BAS models use a vocabulary of tags for semantic description of the represented system. In this context, tags represent keywords for semantic annotation. Transformations to state-of-the-art integration solutions based on WSs are defined in order to provide repeatable, automatic mappings from the tag-based model (Section 4.4). The proposed approach found its way into the recent KNX Web services specification, which shows its applicability (Section 4.5).

4.2 Basics and requirements

Literature already refers to diverse approaches for the integration of BASs into IT systems aiming at process optimization, energy management, or superior comfort control. In this context, WSs are a suitable method offering an interoperable communication interface. Bai et al. use WSs as a middleware technology that tackles the problems resulting from incompatible protocols [5]. Standardized integration technologies that utilize WSs emerged, such as Open Building Information Exchange (OBIX), BACnet Web services (BACnet/WS), Devices Profile for Web Services (DPWS), or OPC Unified Architecture (OPC UA). An approach based on OBIX to connect BA devices with a dedicated server is shown in [6]. Another approach using OBIX at the application layer in combination with IPv6 for IoT integration of BASs is presented in [7]. In [8], a mapping from KNX to BACnet/WS is discussed while the integration of BASs in general by means of BACnet/WS is addressed in [9]. Mappings from different BA technologies to OPC UA are presented in [4]. Han et al. use DPWS as the basis for their service-oriented architecture [10]. Utilization of OPC UA and DPWS in industrial automation is, for example, examined by Candido et al. [11].

Most of these approaches have in common that they manually model a BAS directly in the WS interface [7, 8]. They do not define an automatic workflow from the engineering phase of the BAS to the point of integration into the IoT. Thus, an efficient and universal reuse is not supported. Moreover, the integration might need more information than it is modeled in the engineering phase. In addition, each BA technology has its own engineering tool and information representation. Here, Model-Driven Engineering (MDE) offers a solution in terms of a homogeneous modeling environment supporting automatic transformation between different modeling languages [12]. An example is given in [13], where a basic modeling language for the description of BASs is defined.

On the other hand, modeling of semantics is an important aspect with regard to interoperability in the heterogeneous environment of IoT applications and BA technologies. Recently, ontologies became a powerful method to model semantics. A pioneering example is the work of Ploennigs et al. introducing the BASont ontology for BASs that supports different use cases over the system life cycle [14]. The DogOnt ontology focusing on BA devices, states, and functionality in domotic systems is presented in [15]. Kofler et al. enhance this ontology and add energy-related information for smart homes summarized in the ThinkHome ontology [16]. The BOnSAI ontology based on CoDAMoS and OWL-S provides concepts for describing services, functionality, hardware, and context awareness for ambient intelligence [17]. A knowledge-based system for requirements engineering in BA from a function-oriented perspective is presented in [18]. Ontology utilization in order to detect abnormal building behavior is shown in [19]. Another form of semantic modeling suitable for the proposed framework is provided by tag-based annotations. An example is Project Haystack that offers a vocabulary of tags to describe BASs and integrates the resulting models into a WS interface [20]. However, Project Haystack does not support an explicit, machine-readable formalization of tag relations and tag compositions. The modeling framework should be usable by different user groups with different modeling know-how, like BA manufacturers, application developers, facility managers, or building owners. Thus, an intuitive and simple way of semantic modeling is necessary. As ontologies require more background knowledge in application and handling, a tag-based approach is favored in this work. This is also underpinned by feedback from BA manufacturers that prefer tags-based descriptions for basic IoT integration of BASs in practice.

Considering these issues, the proposed modeling framework combines the idea of a model-driven integration workflow known from MDE and a tag-based modeling approach extending the concept of Project Haystack. Results are an easy to use approach for IoT integration of BASs via WS interfaces, the description of semantics based on an extensible and structured tag vocabulary, and a homogeneous modeling independent of BA technologies. Moreover, automatic transformations imply higher reusability of the integration workflow. Information of the engineering phase is fed into an abstract BAS model, which is finally transformed to a target model of the WS interface.

4.3 Tag-based modeling concept

The modeling concept aims at supporting the integration process of BASs into the IoT. Information about a BAS needs to be made available at the WS interface in order to allow management access from the outside. As there are diverse BA technologies and WS-based integration solutions, a generic modeling concept that abstracts from technology-specifics is required. Moreover, this modeling needs to cover relevant semantics in order to enable machine-to-machine communication, i.e. the provided information has to be interpretable without additional knowledge. Thus, this work presents a model-driven approach supporting semantic modeling by means of an extensible list of tags that are summarized in a vocabulary. In the following, the meta-model as basis for both the vocabulary and the BAS models is explained (Section 4.3.1) before the vocabulary is highlighted (Section 4.3.2).

4.3.1 Common meta-model

In MDE, a system is represented by a model that is an instance of a meta-model describing a digital subscriber line (DSL) [12]. This hierarchy can be further extended by a common meta-meta-model on top in order to declare a shared notion of meta-models, which makes them comparable. Figure 4.1 illustrates these modeling levels. In this work, the system is a particular BAS that should be described by a model using the tag vocabulary. A meta-model is needed that specifies the DSL for the definition of tags and their compositions to class-like structures. Moreover, the meta-model has to include concepts that allow for describing actual entities of a BAS by means of the vocabulary. In MDE, this principle is called meta-modeling [21]. In other words, there is a common meta-model for the instantiation of system models (i.e. BAS models) and tag vocabularies. On the contrary, the vocabulary and the actual system models are in an orthogonal relationship compared to the meta-modeling hierarchy, known as meta-programming [12].

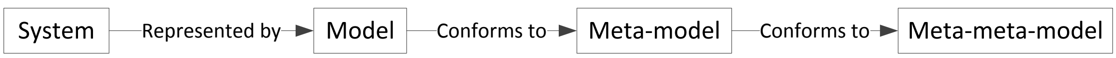

Figure 4.1: Model hierarchy [12]

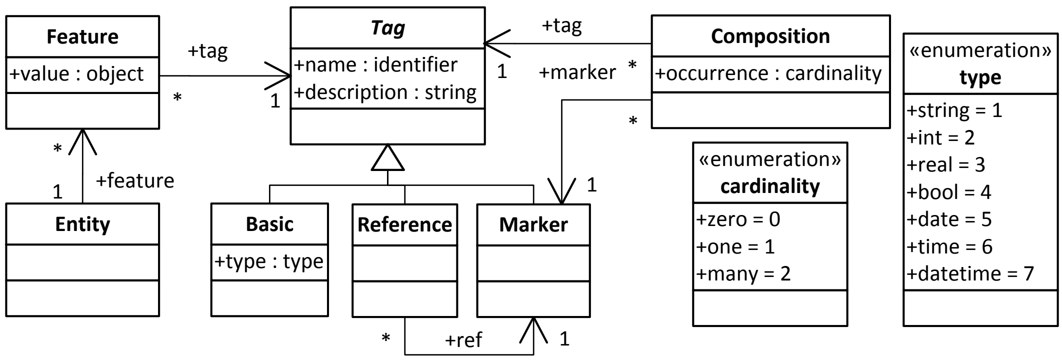

Figure 4.2: Common meta-model for vocabularies and system models

The number of concepts in this common meta-model is limited, but the described DSL enables the creation of powerful vocabularies and comprehensive system models. Figure 4.2 visualizes the meta-model as UML class diagram, which was developed in this work. The specified tags (`Tag`) maintain the description of entities (`Entity`) in the form of features (`Feature`). Thus, an entity is a container for arbitrary features that combine a tag with an optional *value* to express an actual characteristic. Each entity needs at least one feature, which is linked to the tag *id* resulting in a unique identifier for this entity. Basically, tags have a unique *name* and an optional *description*. There are three types of tags. A `Basic` tag has a simple data type corresponding to the `type` enumeration, in particular *string*, *int*, *real*, *bool*, *date*, *time*, and *datetime*. If such a tag is used in a feature, the feature's value needs to be an instance of this data type. A `Marker` is used to express an *is-a* relationship, i.e. a membership to a concept. For example, the marker tag *device* declares an entity as an instance of a BA device. Marker features do not have a value. `Reference` tags are used to describe relationships between entities. The accepted set of entities that can be referenced is defined by the association *ref* between the reference tag and a marker tag. For example, the reference tag *deviceRef* is related to the marker tag *device*. Therefore, an entity can use *deviceRef* to refer to another entity tagged with the *device* marker. Here, the feature's value is set to the unique identifier of the referenced entity.

Additionally, the meta-model defines a mechanism to create tag compositions. This leads to virtual classes forming nested structures of tags that belong together. The class `Composition` is utilized in order to express this kind of tag relation that links marker tags with other tags. The attribute *occurrence* determines the number of allowed appearances of the associated tag in combination with the specified marker tag within a single entity. For this purpose, the enumeration `cardinality` lists three possible values. While *one* and *many* are used as their names indicate, *zero* helps to explicitly model that two tags should never occur together. If the associated tag is also a marker tag, this composition relation is interpreted as inheritance. For example, the marker tag *room* is linked to the marker tag *building part*. Hence, a room entity is also an instance of a building part. In summary, this meta-model provides a succinct basis for the proposed modeling framework. As it is encoded in an XML schema, a wide range of generic visual and textual editors for modeling BASs and vocabularies is available. Moreover, specialized tools like the Web Service Exporter App of KNX can be used (see Section 4.5).

4.3.2 Vocabulary

A core set of the most important tags has to be predefined although a vocabulary needs to be extensible in order to consider future needs regarding technology-specific properties or user and manufacturer preferences. Otherwise, defined vocabularies can degrade to incompatible tag lists that do not guarantee the required interoperability for integration. Therefore, this work identifies the intersection of relevant concepts and terms that all major BA technologies have in common. The idea is to differentiate between (1) the physical elements of the BAS and (2) their arrangements in the context of building, functionality, or topology. Moreover, semantic modeling requires (3) tags to describe basic meta-information like units and enumerations as well as (4) tags to define types of datapoints or function blocks. As information about the latter is often not available in machine-readable form, this modeling framework can be used to define technology specifics in an abstract model. This fundamentally supports automatic processing and increases interoperability for IoT integration of BASs.

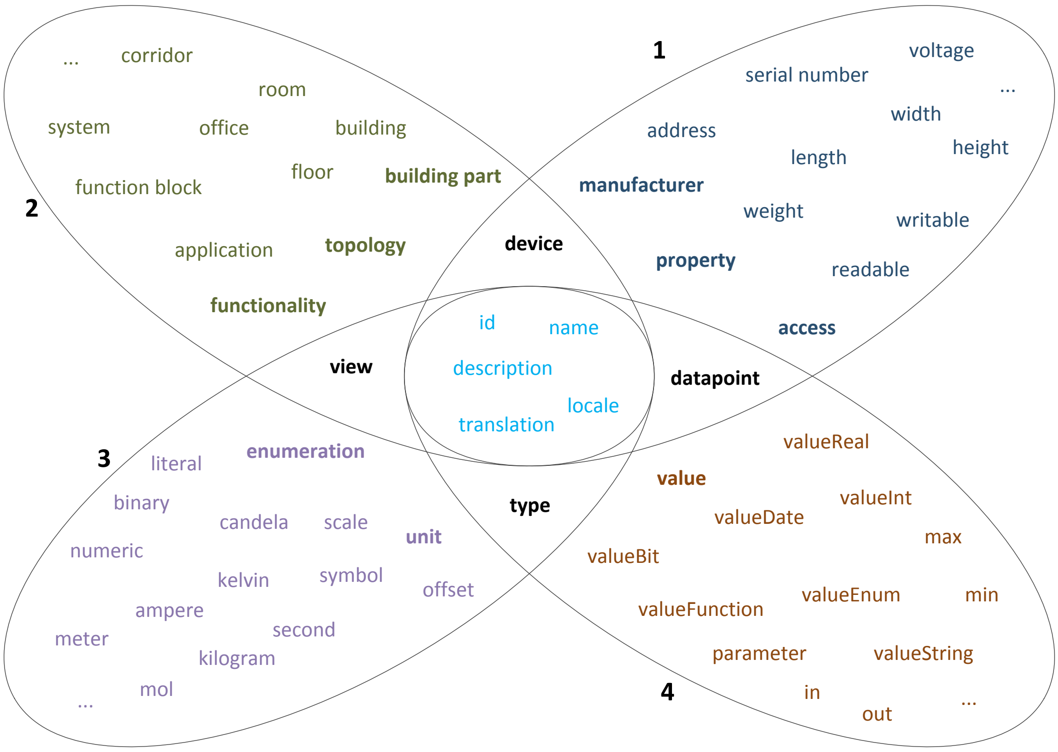

Figure 4.3: Core vocabulary with key tags

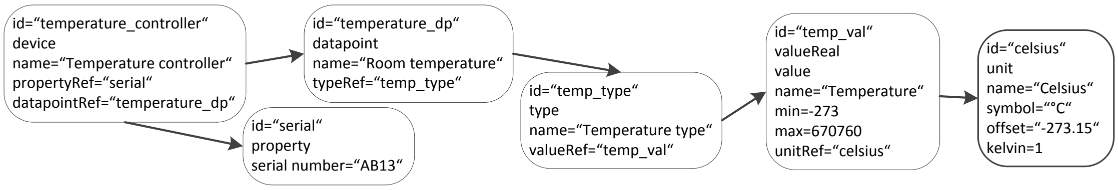

Figure 4.4: Modeling example

In Figure 4.3, these four tag domains are sketched. In order to keep the figure readable, only tag names are listed without their data types or compositions. Reference tags are omitted, as well. The tags at the intersection of all four domains are the most basic tags for identification (*id*, *name*) and textual *description*. Multilingual texts are supported by *translation* and *locale*. In general, tags near the center are more basic than those at the outer edge. In block 1, tags to describe devices are shown. A *device* has a *manufacturer*, some properties like *serial number* or *weight*, and *datapoints* that represent its inputs and outputs. A datapoint can be *readable* or *writable*. Arrangements, which are summarized in block 2, are based on the general marker tag *view* with its inherited markers *functionality*, *topology*, and *building part*. Specialized tags enable a more detailed characterization (e.g. *room*). Tags for *unit* and *enumeration* are concentrated in block 3. Enumerations consist of a set of *literals* with additional *binary* or *numeric* keys. Units are expressed as combination of basic SI units. Finally, block 4 contains the main tags for modeling a datapoint *type* that consists of a set of *values*. Figure 4.4 gives an example for using this vocabulary to model a device with a property, a datapoint, a datapoint type with one value, and a unit. This example also shows the use of reference tags to link entities. BAS models based on this vocabulary and the introduced meta-model represent only static information, i.e. no runtime information in the form of process data is modeled. For instance, facility managers are able to model their BASs based on this framework to enable technology-independent remote monitoring. Also BA manufacturers can provide predefined models of their components in the form of product libraries for reuse by building users or engineers in order to broaden the application area of their products.

4.4 Integration by transformation

The last step towards the IoT integration of BASs by means of a WS interface is realized as an automatic transformation process. Here, technology-independent BAS models are mapped to the information model of a WS-based integration technology. This forms the second part of the proposed modeling framework. Advantages are a better reusability and a more efficient integration with respect to time and effort. For legacy systems, a gateway device will be used to map the communication. If BA devices with IP-based interfaces already support one of the WS interfaces, they can be directly configured in this integration process. The modeling framework tries to support established, open, and standardized WS-based integration technologies. Thus, OBIX [22], RESTful BACnet/WS [23], and OPC UA [24] are selected as target technologies for the transformation.

4.4.1 Generic transformation schema

A dynamic transformation is required as the system models are based on a variable set of tags. Rules formally define the mapping process, which analyze the list of tags, the tag compositions, and the system model. Basically, there are three sets of rules, which are illustrated by the dashed arrows in Figure 4.5. System models that conform to the vocabulary and the meta-model are transformed into objects of the target WS interface, which again represent resources of the BAS. Afterwards, applications access this WS interface to interact with the BAS. While the meta-model specifies the basic structure of system models, actual semantics is provided by the tags and the virtual classes composed of these tags. Similarly, the information models of the WS-based integration technologies offer concepts to define types (e.g. OBIX contracts).

Rule set 1 handles the mapping from the tag vocabulary to the type concepts of the WS interface. If this step is not performed, client applications do not have the full information for interpreting objects. In general, marker tags become distinct types. All other tags that are linked via compositions to a marker tag become attributes of the resulting type. Exceptions are, for example, types that are already defined in the information model of the WS interface.

Second, rules of set 2 are responsible for the extraction of types (e.g. datapoint types) from the system model to types of the WS interface. This rule set is necessary because types are modeled as common entities in the system model using the vocabulary, but result in dedicated types at the WS interface that can be instantiated by objects like an actual datapoint.

Finally, rule set 3 is focused on the transformation of entities that are not yet covered by one of the previous rules. The entities are transferred from the domain of the modeling framework to objects in the domain of the WS interface. Here, the previously created types and the already available, standardized elements of the WS-based integration technology are utilized. More basically, this rule set leads to the translation of all entities that describe the devices, the structure, or the logical views of a BAS. In addition, units

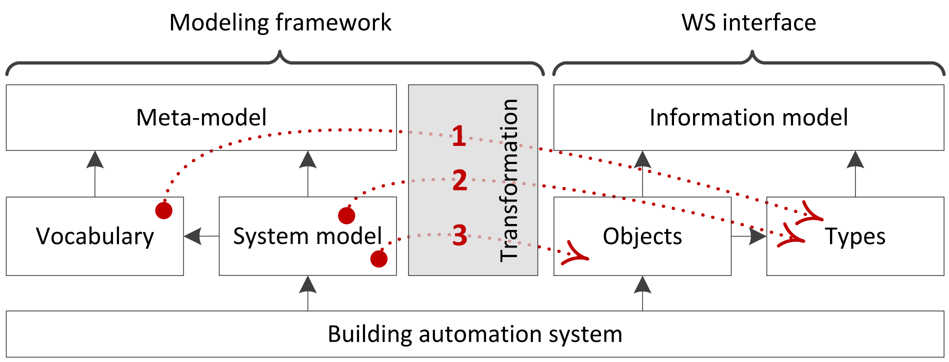

Figure 4.5: Transformation process within the modeling framework

and enumerations are transformed with respect to already existing concepts of the target technology.

4.4.2 Rules for target technologies

The three rule sets are formulated for each of the selected WS-based integration technologies, i.e. OBIX, BACnet/WS, and OPC UA. The transformation results are always the same with respect to the mapped information. However, the specifics of the target technologies need to be considered. Table 4.1 lists the transformation rules for the target technologies in a nutshell. The rules are arranged into the three generic rule sets of Section 4.4.1. In column 2, the source elements of the modeling framework are given. The abbreviation *comp.* means that this rule is only applied to those tags that are in a composition relation with a superior marker tag. Terms like *Type entity* show that the corresponding rule is used with entities marked by the mentioned tag (e.g. *type*). On the other hand, *Entity* refers to all common entities that are not covered by other rules. The features are distinguished by means of their linked tag class (e.g. *Basic feature*). The columns 3 to 5 show the respective target elements in the domain of the specific WS interfaces. Terms written in italic letters indicate that attributes or metadata are used to map the source element to the WS interface (e.g. *is*). All other terms stand for basic types of the WS interfaces (e.g. `Obj`).

OBIX and BACnet/WS have generic information models that can be extended by individual contracts and types in the definition context, respectively. Although the rules are very similar in general, the mapping of units is an exception. While OBIX provides the contract *obix:Unit* to model units in detail, BACnet/WS makes use of the predefined BACnet Engineering Units. Regarding reference tags and reference features, OBIX and BACnet/WS rules distinguish between cardinality *one* and *many*. Cardinality *zero* is not transformed, but it can be used for consistency checks of tag-based models. In order to exemplify a transformation to OBIX, Listing 4.1 shows the result of applying rule set 2 on the entity *temp_type* illustrated in Figure 4.4. The generated contract is an instance of the more general *type* contract. Like in the modeling example, the OBIX *Obj* has a *Real* value that represents a temperature and refers to a unit.

In contrast, OPC UA comes with a quite complex information model hierarchy including various standardized types. These types already have specialized semantics, which should be reused as far as possible. In rule set 1, the *DeviceType* is used for the marker *device* while all other markers are mapped to types inheriting *FolderType*. Regarding rule set

Listing 4.1: OBIX transformation example

```xml
<obj href="temp_type" is="/contracts/type" displayName="Temperature type">
  <real name="temp_val" href="temp_val" displayName="Temperature"
       unit="/units/celsius" min="-273" max="670760" is="obix:Point"/>
</obj>
```

Table 4.1: Technology-specific transformation rules

Set	Framework	OBIX	BACnet/WS	OPC UA
1	Marker tag	Obj (contract)	Object (type)	DeviceType
				FolderType
	Marker tag comp.	*is*	*extends*	*SubtypeOf*
	Reference tag comp.	Ref	Object	*HasComponent*
		List	Collection	*HasProperty*
				Organizes
	Basic tag comp.	Real, Int, ...	Real, Integer, ...	PropertyType
2	Type entity	Obj (contract)	Object (type)	BaseDataVariableType
	Value entity	Real, Int, ...	Real, Integer, ...	DataItemType
3	Unit entity	Obj (obix:Unit)	BACnet Unit	EngineeringUnits
	Enumeration entity	Enum	Enumerated	Property
			Boolean	
	Entity	Obj	Object	Device
				Folder
				BaseDataVariable
				DataItem
	Marker feature	*is*	*type*	*HasTypeDefinition*
	Reference feature	Ref	Object	*HasComponent*
		List	Collection	*HasProperty*
				Organizes
	Basic feature	Real, Int, ...	Real, Integer, ...	Property

2, type entities with only one value entity are combined to a single *DataItemType*. On the contrary, a complex type entity is mapped to a *BaseDataVariableType* as container for *DataItemTypes*. Similar to BACnet/WS, units use the available *EngineeringUnits*. The mapping of reference tags and reference features depends on the type of the referred

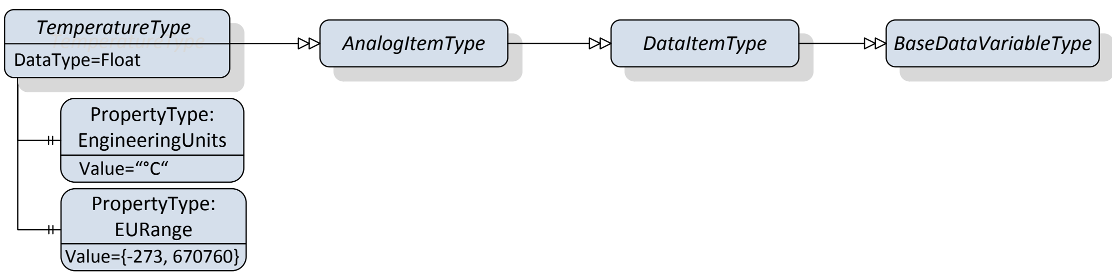

Figure 4.6: OPC UA transformation example

element. For example, *Organizes* is used for *Folder*, but *HasComponent* links *DataItem*. Similar to the example above, Figure 4.6 shows the result of applying rule set 2 for OPC UA. The type *TemperatureType* is a subtype of the built-in *DataItemType*. Two attached properties model unit and range of this type.

4.5 Applicability and feasibility

The modeling framework with its tag-based modeling and the transformation rules towards WS interfaces influenced the development of the KNX Web services (KNX WS) specification. Hence, KNX WS is used to highlight the applicability and feasibility of the proposed approach. The KNX standard (ISO/IEC 14543-3) is a notable representative in home and building automation [25]. While current integration of KNX networks into IP-based systems requires specific knowledge about the KNX protocol (KNXnet/IP Tunneling, KNXnet/IP Routing), the KNX WS specification provides external applications an access interface to KNX networks that is independent of the KNX protocol. Application development is simplified, and BAS communication is hidden behind a gateway device.

An overview of KNX WS is depicted in Figure 4.7. The KNX Network is the BAS that should be integrated into the IoT. The KNX WS Gateway bridges the communication in order to enable KNX-independent access for external clients via the KNX Web interface. The Engineering Tool Software (ETS) forms the primary source for a technology-independent, tag-based model known as KNX Information model. The ETS export is based on the Web Service Exporter App. Using individual tags, additional information can be added to this model. KNX WS is intended to support OBIX, BACnet/WS, and OPC UA as basis for the KNX WS Gateway. The KNX Information model is integrated into the gateway using the transformation rules of this modeling framework. After initialization, communication with the clients (KNX Web interface) and the KNX Network (KNX Network access) can be performed. Depending on the implemented gateway profile, certain communication protocols (e.g. CoAP, HTTP) and information encodings (e.g. XML, JSON) need to be provided.

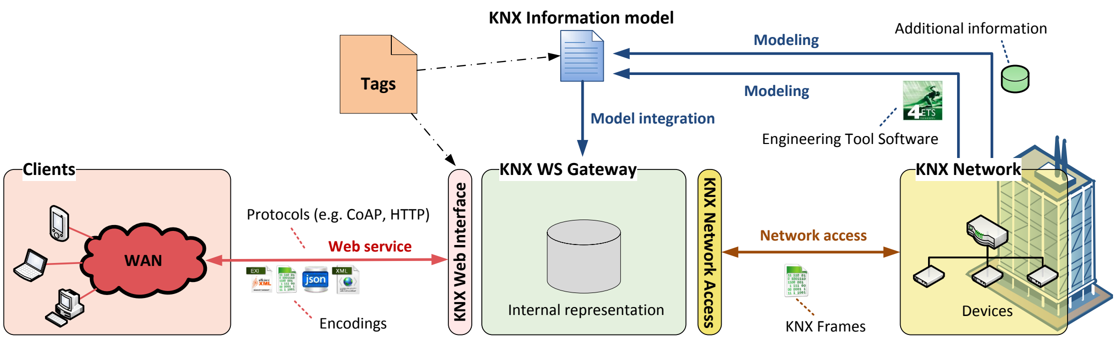

Figure 4.7: Overview of KNX Web services [26]

Compared to the meta-model introduced in Section 4.3.1, KNX WS relies on a slightly simplified version. Marker and reference tags are declared using additional literals in the *type* enumeration, i.e. there are no distinct sub classes for these special tags. The class *feature* is called *tag/value pair*, and the cardinality *zero* is not supported. Similarly, the vocabulary of core tags as discussed in Section 4.3.2 is supplemented by some tags to model channels and functional blocks as well as additional device properties. An essential task of the KNX WS Gateway is the mapping of requests from external applications to the standardized KNX communication. Thus, semantics about access methods (e.g. group communication objects) needs to be captured by the tag vocabulary. Besides specific tags for the different access types, the general KNX flags (e.g. updatable, transmittable) are added to the vocabulary. Tags for least significant bit (LSB) and most significant bit (MSB) for the values of a type help the KNX WS Gateway to send and receive KNX frames. However, most of the already defined tags are reused in the KNX WS specification showing the applicability of this core vocabulary. Once the KNX Network is defined using the tag-based modeling concept, the model can be transformed to one of the supported WS-based integration technologies. There is no necessity to modify the specified transformation rules.

According to the example in Figure 4.4, the modeled type *temp_type* representing a temperature is specified as DPST 9.001 in the KNX standard. The corresponding KNX group communication object is modeled as separate entity that is linked to the datapoint *temperature_dp*. Additional device properties (e.g. order number, individual address) are added. The final model is integrated into an OBIX-based gateway that supports HTTP communication and XML encoding. Based on this, a request and the corresponding response for reading information of the temperature controller are given in Listing 4.2. Here, the modeled device is part of the logical view *all* of the KNX Network *demo*. The *is* attributes of the returned objects refer to types that are created by applying transformation rules of set 1 on the KNX tag vocabulary. Although some information is omitted in this listing, it can be seen that the objects contain all the modeled information. The meaning can be determined by analyzing the types of the requested objects or the values and attributes of the child elements as well as the relationships to other objects by following the links. All in all, the KNX WS specification is a good example that the proposed approach can already be used in practice. By means of the tag-based modeling and the integration by transformation, interoperability between KNX and IoT applications or other IT systems is established. Moreover, application development is eased using common standards instead of struggling with specifics of BA technologies.

4.6 Conclusion and outlook

In summary, this work presents a modeling framework for an automatic and seamless integration of BASs into the IoT. WS-based integration technologies are used to bridge these two worlds. The framework defines, on the one hand, a tag-based modeling concept for semantic description of a BAS on an abstract, technology-independent level. Besides a set of core tags, the vocabulary can be individually extended offering enough flexibility

Listing 4.2: Interaction example

```
GET http://localhost:8080/installations/demo/views/all/devices/
    temperature_controller

-- response --
<obj name="temperature_controller" href="temperature_controller/"
    displayName="Temperature controller" is="/knx/device">
  <int name="individualAddress" href="individualAddress" val="1.1.10"/>
  <str name="orderNumber" href="orderNumber" val="00110011"/>
  <str name="serialNumber" href="serialNumber" val="AB13"/>
  <list href="datapoints" of="obix:ref /knx/datapoint">
    <ref href="datapoints/temperature_dp" is="/knx/DPST_9_001"/>
  </list>
</obj>
```

for future needs. Components of the BAS, their logical and physical arrangements, the inherent functionality, or general elements such as units are described by means of these tags. The modeling language to design the tag vocabulary and the BAS models is specified in a meta-model. On the other hand, the modeling approach consists of transformation rules in order to map the tag-based BAS models to the information models of the WS-based target technologies. Rules are defined for OBIX, BACnet/WS, and OPC UA as relevant representatives for IoT integration of BASs. Hence, the presented framework supports homogeneous, technology-independent system modeling that is easily applicable, expressive, and extensible. Semantics is represented in a machine-readable form as basis for an automatic integration process.

One of the next steps is the formulation of transformation rules for the Web Ontology Language (OWL). Mapping a tag vocabulary and a BAS model to OWL is a rather simple task. Marker tags are defined as OWL classes. Reference tags become object properties with range restrictions while simple tags are described as data properties of a certain type. The compositions lead to a taxonomy of classes with property constraints. Thus, integration into SPARQL endpoints becomes possible, which enables Semantic Web applications to access BASs. Interaction with BA devices by means of SPARQL queries leads to an even higher level of abstraction than with the currently used WS-based integration solutions. Furthermore, a focus is on the integration of the proposed modeling framework into the Model-Driven Architecture (MDA) approach. Modeling tools, like the Eclipse Modeling Framework (EMF), and transformation standards, like Query/View/Transformation (QVT), can be utilized as illustrated in [13]. Moreover, KNX WS will be used to request feedback from the BA industry, application developers, or users in order to improve the modeling framework.

Acknowledgment

The authors want to thank the KNX Association for supporting the development of the KNX WS specification.

References

[1] A. Whitmore, A. Agarwal, and L. Da Xu, "The Internet of Things—A survey of topics and trends", *Information Systems Frontiers*, vol. 17, no. 2, pp. 261–274, Apr. 2015, ISSN: 1387-3326. DOI: 10.1007/s10796-014-9489-2.

[2] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey", *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2010.05.010.

[3] W. Kastner, M. Kofler, M. Jung, G. Gridling, and J. Weidinger, "Building Automation Systems Integration into the Internet of Things: The IoT6 approach, its realization and validation", in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2014, pp. 1–9. DOI: 10.1109/ETFA.2014.7005197.

[4] W. Granzer and W. Kastner, "Information Modeling in Heterogeneous Building Automation Systems", in *IEEE International Workshop on Factory Communication Systems*, May 2012, pp. 291–300. DOI: 10.1109/WFCS.2012.6242577.

[5] J. Bai, Y. Hao, and G. Miao, "Integrating Building Automation Systems based on Web Services", *Journal of Software*, vol. 6, no. 11, pp. 2209–2216, Nov. 2011, ISSN: 1796-217X. DOI: 10.4304/jsw.6.11.2209-2216.

[6] H. Järvinen, A. Litvinov, and P. Vuorimaa, "Integration Platform for Home and Building Automation Systems", in *IEEE Consumer Communications and Networking Conference*, Jan. 2011, pp. 292–296. DOI: 10.1109/CCNC.2011.5766475.

[7] M. Jung, J. Weidinger, W. Kastner, and A. Olivieri, "Building automation and smart cities: An integration approach based on a service-oriented architecture", in *IEEE International Conference on Advanced Information Networking and Applications*, Mar. 2013, pp. 1361–1367. DOI: 10.1109/WAINA.2013.200.

[8] W. Kastner and S. Szucsich, "Accessing KNX networks via BACnet/WS", in *IEEE International Symposium on Industrial Electronics*, Jun. 2011, pp. 1315–1320. DOI: 10.1109/ISIE.2011.5984349.

[9] D. Schachinger, C. Stampfel, and W. Kastner, "Interoperable integration of building automation systems using RESTful BACnet Web services", in *Annual Conference of the IEEE Industrial Electronics Society*, Nov. 2015, pp. 3899–3904. DOI: 10.1109/IECON.2015.7392708.

[10] S. N. Han, G. M. Lee, and N. Crespi, "Semantic Context-Aware Service Composition for Building Automation System", *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 752–761, Feb. 2014, ISSN: 1551-3203. DOI: 10.1109/TII.2013.2252356.

[11] G. Cândido, F. Jammes, J. B. de Oliveira, and A. W. Colombo, “SOA at Device level in the Industrial domain: Assessment of OPC UA and DPWS specifications”, in *IEEE International Conference on Industrial Informatics*, Jul. 2010, pp. 598–603. DOI: 10.1109/INDIN.2010.5549676.

[12] J. Bézivin, “On the unification power of models”, *Software and Systems Modeling*, vol. 4, no. 2, pp. 171–188, May 2005, ISSN: 1619-1374. DOI: 10.1007/s10270-005-0079-0.

[13] D. Schachinger and W. Kastner, “Model-Driven Integration of Building Automation Systems into Web Service Gateways”, in *IEEE World Conference on Factory Communication Systems*, May 2015, pp. 1–8. DOI: 10.1109/WFCS.2015.7160561.

[14] J. Ploennigs, B. Hensel, H. Dibowski, and K. Kabitzsch, “BASont - A modular, adaptive Building Automation System Ontology”, in *Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2012, pp. 4827–4833. DOI: 10.1109/IECON.2012.6389583.

[15] D. Bonino and F. Corno, “DogOnt - Ontology Modeling for Intelligent Domotic Environments”, in *The Semantic Web - ISWC 2008*, ser. Lecture Notes in Computer Science, A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, Eds., vol. 5318, Springer Berlin Heidelberg, 2008, pp. 790–803, ISBN: 9783540885634. DOI: 10.1007/978-3-540-88564-1_51.

[16] M. J. Kofler, C. Reinisch, and W. Kastner, “A semantic representation of energy-related information in future smart homes”, *Energy and Buildings*, vol. 47, pp. 169–179, Apr. 2012, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2011.11.044.

[17] T. G. Stavropoulos, D. Vrakas, D. Vlachava, and N. Bassiliades, “BOnSAI: A Smart Building Ontology for Ambient Intelligence”, in *ACM International Conference on Web Intelligence, Mining and Semantics*, Jun. 2012, pp. 1–12. DOI: 10.1145/2254129.2254166.

[18] S. Runde and A. Fay, “Software Support for Building Automation Requirements Engineering—An Application of Semantic Web Technologies in Automation”, *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 723–730, Nov. 2011, ISSN: 1551-3203. DOI: 10.1109/TII.2011.2166784.

[19] J. Ploennigs, A. Schumann, and F. Lécué, “Adapting Semantic Sensor Networks for Smart Building Diagnosis”, in *The Semantic Web - ISWC 2014*, ser. Lecture Notes in Computer Science, P. Mika, T. Tudorache, A. Bernstein, C. Welty, C. Knoblock, D. Vrandečić, P. Groth, N. Noy, K. Janowicz, and C. Goble, Eds., vol. 8797, Springer International Publishing, 2014, pp. 308–323, ISBN: 9783319119144. DOI: 10.1007/978-3-319-11915-1_20.

[20] Project Haystack Corporation, *Project Haystack*, http://project-haystack.org, accessed: Jan. 16, 2017.

[21] M. Brambilla, J. Cabot, and M. Wimmer, *Model-Driven Software Engineering in Practice*. Morgan & Claypool, 2012, ISBN: 9781608458837. DOI: 10.2200/S00441ED1V01Y201208SWE001.

[22] OASIS, *OBIX Version 1.1, Committee Specification Draft 02 / Public Review Draft 02*, Dec. 2013.

[23] ASHRAE, *Data Communication Protocol for Building Automation and Control Networks (ANSI/ASHRAE Addendum am to ANSI/ASHRAE Standard 135-2012)*, Apr. 2016.

[24] OPC Foundation, *OPC Unified Architecture Specification, Part 1: Overview and Concepts, Release 1.03*, Oct. 2015.

[25] KNX Association, *KNX Standard Version 2.1*, Oct. 2013.

[26] D. Schachinger and W. Kastner, "Integration von KNX Netzwerken in das Internet der Dinge: Die KNX Web Services Spezifikation", in *Jahreskolloquium Kommunikation in der Automation*, Nov. 2016, pp. 1–8.

CHAPTER 5

Ontology-based abstraction layer for smart grid interaction in building energy management systems

Abstract: Replacing traditional power grids by future smart grids opens manifold opportunities for energy-efficient operation of buildings and cities as well as improved coordination of energy demand and supply. Current information and communication technology provides a suitable basis for the bidirectional flow of information between buildings and other smart grid stakeholders. However, a common notion of shared knowledge is essential in order to unify heterogeneous grid environments, incorporate information of smart grid participants, and process this information in building energy management. In this work, an abstraction layer based on an OWL ontology is presented that enables semantic representation of knowledge for interaction between building energy management systems and smart grids. A well-proven methodology is used to develop this ontology. Furthermore, the ontology application into building energy management systems and smart grid environments is illustrated, and the functional capabilities of this approach are shown.

Keywords: Energy management, information technology, knowledge representation, smart grids

---

The content of this chapter has been published: D. Schachinger, W. Kastner, and S. Gaida, "Ontology-based abstraction layer for smart grid interaction in building energy management systems", in *IEEE International Energy Conference*, Apr. 2016, pp. 1–6.

5.1 Introduction

A substantial part of world-wide energy needs is demanded by residential and commercial buildings with nearly 40% of final energy consumption [1]. Thus, energy efficiency and optimization of energy consumption become increasingly important in order to lower costs and reduce detrimental effects on the environment. For this purpose, coordination of energy supply and demand and increased use of decentralized energy resources are essential measures. As traditional power grids do not provide the required infrastructure, they are going to be replaced by smart grids, which are characterized by a bidirectional flow of information and energy, self-healing capabilities, adaptiveness, and support of islanding [2]. Therefore, preservation of grid stability, protection of infrastructure, and balancing of energy production and consumption can be realized. Especially demand response (DR) mechanisms, which can be divided into market DR (e.g. real-time pricing) and physical DR (e.g. emergency management), are well-suited in order to achieve smart grid objectives [3]. In building energy management systems (BEMSs), participation in such DR programs is used to interact with other grid stakeholders, such as energy generators, grid operators, or energy retailers. Basic use cases are the adoption of day-ahead pricing strategies, selling of flexibilities, or the realization of priority functions to maintain grid stability [4]. Local production and consumption of energy can be balanced in accordance with grid information to finally reduce building's energy needs or minimize energy purchasing costs.

In order to realize an efficient interplay of grid stakeholders, a communication stack with the Internet Protocol (IP) as central element and a common communication language on top can be used as basis [5]. Thus, existing information and communication technology (ICT) infrastructure can be reused. Nonetheless, certain requirements need to be addressed in smart grid communication [6]. On the information level, a BEMS has to combine multiple domains, such as usage profiles, environmental influences, smart grid interactions, or building automation systems, in order to optimize local energy consumption. This set of information and knowledge has to be managed in a structured way to aid BEMS functionality.

Both databases and ontologies are appropriate solutions for such an information and knowledge representation supporting BEMS interaction with smart grid stakeholders. While especially relational databases and the relational data model have become de facto standards for information storage and querying in the last decades, ontologies are relatively new in the field of computer science. Databases are primarily focused on meeting requirements of a specific application or organization. Efficient storage and processing are further characteristics of relational databases. Stimulated by the Semantic Web, ontologies are most suitable for semantic modeling and inference of new knowledge by means of reasoners. Furthermore, ontologies are more beneficial for sharing knowledge among different domains and linking distributed knowledge [7].

Therefore, ontologies are favored over databases for describing interaction between BEMSs and smart grids. Although databases provide efficient data management, advantages of ontologies, like a higher level of abstraction, implementation independence, or semantic modeling of information, are prevailing regarding application in an open, flexible, and evolving environment as the smart grid. Additionally, ontologies are identified as promising data modeling techniques for the smart grid enabling formal semantics in combination with a shared understanding [8]. In the domain of smart buildings, literature already describes different ontologies. BOnSAI ontology, for example, covers building functionality (e.g. operations), hardware (e.g. devices, appliances), users (e.g. profiles), and context information (e.g. location, rooms) [9]. Besides processes, resources, user comfort, or external influences, energy-related information (e.g. energy types, tariffs, providers) is part of ThinkHome ontology [10]. In [11], an ontology with a focus on home appliances and household information for efficient energy management is defined. An approach combining various ontologies in the field of smart buildings is described in [12]. Moreover, a semantic smart grid information model in order to support DR is proposed in [13]. Although static information about power grid infrastructure (e.g. distribution network) is covered in this ontology, it lacks in concepts to define the context of smart grid services and interaction.

Hence, this work aims at developing an ontology-based abstraction layer that integrates all relevant concepts for smart grid interaction in order to provide homogeneous knowledge representation for BEMSs. Based on the ontology, a BEMS is able to gain knowledge of both dynamic and static characteristics of the ambient smart grid. Furthermore, reasoning functionality of the ontology is used to infer new knowledge maintaining BEMS operation. Additionally, this ontology is combined with other smart building ontologies forming a highly applicable and interoperable abstraction layer for energy management of buildings in a smart grid context. Moreover, application of the proposed approach is illustrated. Evaluation is divided into two parts. First, the modeling capability of the ontology is analyzed, and the reasoning potential is studied. Second, a DR test case is processed in a smart grid test bed in order to show the potential for BEMSs.

5.2 Ontology development

Ontologies comprise definitions for objects and types of objects as well as their semantics and relations in a formal, machine-readable way forcing a shared understanding of some domain [14]. In contrast to a mere taxonomy, which represents a hierarchically organized vocabulary of generic and specialized concepts, an ontology extends this idea by means of concept relations or constraints in order to enhance semantic interpretation. Main components of ontologies are a hierarchy of concepts representing types of entities, relations between concepts, restrictions on relations, and instances.

Literature refers to several ontology development methodologies. For example, the NeOn methodology describes a set of nine scenarios for ontology engineering [15]. Depending on availability and type of resources (e.g. non-ontological resources, ontology design patterns) as well as handling of these resources, the scenarios describe different development processes. If no knowledge resource is reused, the desired ontology is built from scratch consisting of the steps specification, scheduling, conceptualization, formalization, and implementation. Another methodology is presented as ontology creation guide called Ontology Development 101, which describes a process containing seven phases [16]. First, the scope and domain of the ontology need to be clarified. Then, reusability of existing resources is considered, and most important terms are enumerated. These terms result in a taxonomy of classes that form the concepts of the examined domain. Subsequently, properties and constraints, which describe characteristics and relations of classes, are defined before instances of classes are created. In addition to this process, guidelines for consistent creation of ontologies are specified. In [14], a methodology including identification of purpose and scope, ontology creation, evaluation, documentation, and guidelines is presented.

In general, development of ontologies is an iterative process with different ways of modeling a particular domain [16]. Although there are several convenient methodologies for developing the smart grid ontology, Ontology Development 101 is chosen due to its simplicity and clear structure. In the following, this methodology is used to develop the proposed ontology for smart grid interaction in BEMSs.

1) Scope: In order to support BEMSs to optimize local energy consumption and minimize costs by consideration of smart grid conditions, the ontology covers four main blocks. Although BEMSs are primary users of the ontology, other smart grid stakeholders, such as energy retailers or grid operators, maintain the stored knowledge. Thus, smart buildings always have up-to-date information available. In Figure 5.1, the ontology domain with its four primary blocks is illustrated:

- Agents or stakeholders of the smart grid, such as energy retailers, grid operators, or energy aggregators, are addressed by defining concepts to model their identity (e.g. name, location) and functional capability (e.g. service line-up, operating area). Figure 5.1 shows different types of agents ranging from central energy generators to customers and their buildings. The location of agents is visualized by the underlying map.
- Interaction scenarios form an important part of this ontology. Besides definition of processes, like trading of flexibilities or exchange of pricing information, involved agents and their roles are taken into consideration. Semantically enriched modeling facilitates automatic processing of smart grid interaction at the BEMS level. In Figure 5.1, the interaction capabilities of agents are illustrated in the form of offered and required service interfaces in UML notation. Interaction processes are composed of these service interfaces.
- Communication technologies are incorporated including information about the used protocols or the communication parameters, like security features. This information is relevant for particular services offered by smart grid agents. As an example, Figure 5.1 defines communication protocol and language for some service interfaces.

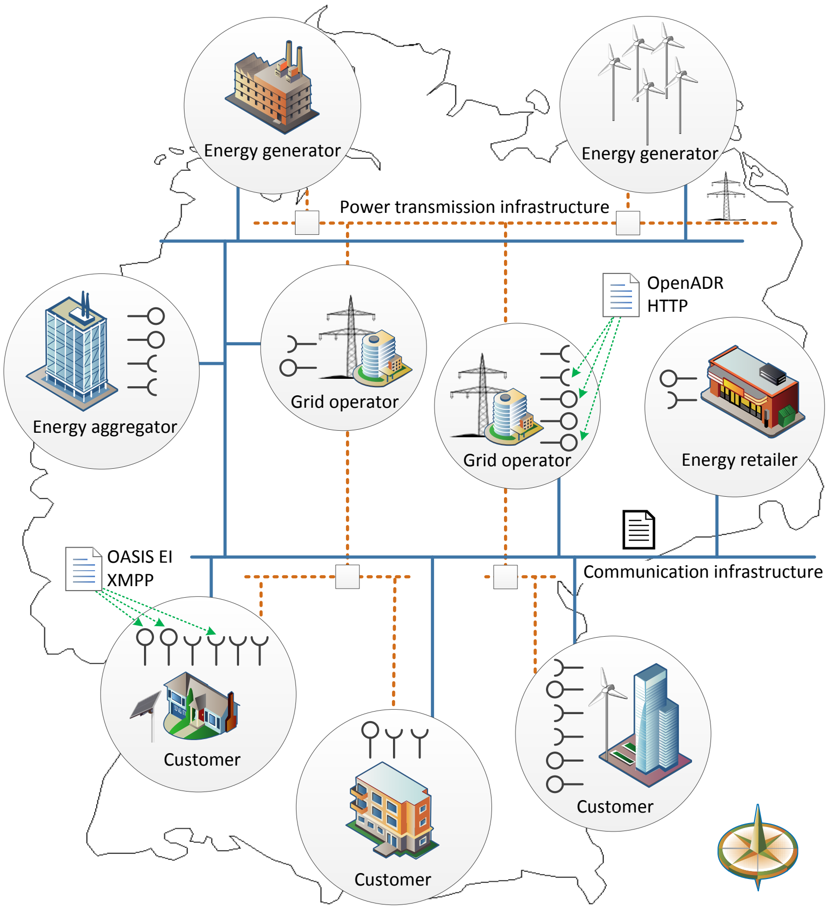

Figure 5.1: Ontology domain

- Grid structure concepts enable the modeling of spatial arrangement of smart grid agents as well as hierarchical routing paths via various intermediate nodes (e.g. substations). Exemplarily, Figure 5.1 shows some grid agents as well as ordinary grid nodes. While the smart grid describes two infrastructures, the concepts of this ontology block are intended to model the power transmission infrastructure. Communication infrastructure is assumed to use already existing ICT infrastructure.

2) Reuse: Suitable, existing ontologies should be reused as far as possible to avoid reinventing the wheel. However, no ontologies have been found covering all blocks of the introduced domain. At least, concepts for the static grid structure can be reused [13]. Generic ontologies, like FOAF or Linked Geo Data, describing basic concepts are reused to develop the smart grid ontology. Moreover, concepts of the developed ontology are linked with other ontologies of building automation or energy management domain in order to increase interoperability of knowledge bases. Examples are BOnSAI [9], ThinkHome [10], or SeWoA [17].

3) Terms: The most important terms in the context of this work are again split into four main blocks. First, *agent* and the more specialized *energy retailer*, *grid operator* (used for both distribution and transmission system operator), *energy aggregator*, *energy generator*, and *customer* are identified. All agents have a *location*, and energy retailers, for example, sell energy of a certain *energy type* (e.g. *gas*, *electricity*, *oil*) in at least one *energy tariff*. Second, grid structure is built using the terms *grid segment* and *grid node*. Agents are connected to grid nodes. Third, lists of communication protocols (e.g. *HTTP*, *XMPP*, *CoAP*) and message exchange languages (e.g. *OpenADR*, *OASIS EI*) are elaborated. In addition, each agent has an *identifier* and is connected via at least one physical *medium* to the communication infrastructure. Finally, terms for interaction scenarios are enumerated. Each agent offers a set of *services* that are either *ingoing* or *outgoing*. A service is used to send or receive data, which are nested sets of *parameter* values (e.g. *price*, *time*, *flexibility offer*, *grid regulation*). In order to give an overview on the terms of this ontology, only a few examples are listed in this paragraph although the complete list contains a lot more concepts.

4) Classes: According to [14], a class hierarchy can be built using a top-down, bottom-up, or middle-out approach. As the ontology is mainly used by automatically operating BEMSs, there is no need for a highly fine-grained taxonomy of classes in a first step. Thus, the top-down approach is chosen starting with the most general concepts and adding necessary, specialized classes afterwards. The most important top-level concepts of the ontology are *Agent*, *Service*, *Node*, *Segment*, *Technology*, *Protocol*, *Language*, and *Parameter*.

Agents are further divided into *EnergyRetailer*, *GridOperator*, *EnergyGenerator*, *EnergyAggregator*, and *Customer*. These classes enable the modeling of most common communication patterns in smart grid applications [4]. A grid operator can be classified into *TransmissionSystemOperator* or *DistributionSystemOperator*. Depending on their offered services, a grid operator can adopt different roles, like *FlexibilityOperator* or *LowVoltageGridController*. There is also a categorization for energy generators (e.g. *CoalFiredPowerPlant*, *WindFarm*) or customers (e.g. *IndustrialCustomer*, *ResidentialCustomer*).

The main categories of services are *OfferedService* and *RequiredService*. Moreover, there are specialized services, like *FlexibilityRequest*, *EnergyForecast*, *FlexibilityOffer*, *GridRegulation*, *PriceDistribution*, or *FlexibilityOrder*. Services are able to transmit data, i.e. a set of values, and parameter types are used to describe the semantics of these values. Therefore, the parameter class is divided into *Price*, *Energy*, *Power*, *Time*, *Request*, *Order*, *Offer*, *Info*, and *Instruction*. For example, energy tariffs provided by energy retailers are published by means of an offered price distribution service, and the data values are prices per time slots.

Regarding grid nodes, no further classification is made as the BEMS does not directly interact with grid nodes but with agents connected to grid nodes. The grid segments are subdivided according to geographical (e.g. *LocalSegment*, *RegionalSegment*, *National-*

Segment), physical (e.g. *LowVoltageSegment*, *HighVoltageSegment*), and functional characteristics (e.g. *TransmissionSegment*, *DistributionSegment*).

Communication technologies are not split into subclasses, but they are described by means of the associated communication protocol and language. Both protocol and language are closed classes defining an enumeration of available instances. Security parameters (e.g. encryption algorithm) are also defined in a particular communication technology. *Medium* describes an enumeration of instances, as well. Class hierarchies for energy type (e.g. renewable and nonrenewable energy sources) and location (e.g. cities and countries) are imported from other ontologies like ThinkHome.

5) Properties: The properties of classes are defined in parallel to the development of the class hierarchy as they are tightly coupled. In this step, all properties that are relevant for smart grid interaction in BEMSs are specified. For identification, the data properties *name* and *identifier* are created, which can be used for agents, services, grid segments, and grid nodes. Regarding the location of spatial things like agents and nodes, two concepts are developed. Geographic coordinates can be set by using data properties *longitude*, *latitude*, and *altitude*. On the other hand, each spatial thing can have an address comprising of *street* and *streetNumber* in combination with the object property *hasCity*. As a grid segment is the combination of various grid nodes, the property *comprises* and its inverse *belongsTo* are needed. The link between grid nodes is realized by means of the symmetric property *adjacentTo*.

The energy retailer supplies the customer with different types of energy that are produced by means of different energy sources. Thus, an energy retailer has the properties *hasEnergySource* and *providesEnergyType*. In order to model an energy mix, each energy source has the data property *percentage*. Some of these properties are adopted from the ThinkHome ontology to make use of already available ontological concepts [10]. In general, each agent is connected to the grid via a grid node (*connectedTo*) and operates on a certain area (*covers* and its inverse *isCoveredBy*). Moreover, an agent offers (*offersService*) and requires services (*requiresService*).

These services use nested parameter configurations to describe the exchanged data. Each configuration has parameters (*hasParameter*) and is linked to a service (*hasConfiguration*). Services can have dependencies to other services, which leads to interaction sequences. This is realized by means of the properties *isFollowedBy* and *isPrecededBy*. Furthermore, the technological requirements for service execution need to be defined. Therefore, the property *hasTechnology* is specified, and a communication technology has the properties *hasProtocol* and *hasLanguage*. Additionally, a service endpoint is identified by means of the data property *url*.

6) Facets: The used value types for data properties are integer (e.g. street number), float (e.g. latitude, longitude), and string (e.g. street, name, identifier). Most of these data properties are declared as functional. For object properties, domain and range have to be specified, and the cardinality of the properties in the classes needs to be defined. As an example, the property *adjacentTo* is used in the domain of grid nodes and can range over all grid nodes. It can occur multiple times within the class *Node*, and furthermore it is a symmetric property. The properties *offersService* and *requiresService* are in the agent domain, and they can link services to an agent. Similarly, all other properties are described. In addition, primitive and defined classes are specified in order to determine the scope of these classes and their disjunction to other classes. Reasoners can use this information to infer class membership of instances or detect inconsistencies in the ontology.

7) Instances: Based on the specified classes, their properties, and the relations between the classes, instances can be modeled. In this step, all instances needed for enumerated classes are created. For example, communication protocols (e.g. HTTP) and communication languages (e.g. OpenADR) are defined. The definition of instances describing a smart grid environment is addressed in Section 5.3.

In summary, an ontology is developed which affords BEMSs the acquisition of meaningful information of the surrounding smart grid. Thus, a BEMS is able to assess available agents and their services as well as the grid infrastructure. Additionally, access mechanisms to these services using communication technology can be modeled in this ontology.

5.3 Ontology application

The developed ontology, first, needs to be implemented using standard Semantic Web technologies (see Section 5.3.1). Afterwards, integration and usage of the ontology in the context of a BEMS are described (see Section 5.3.2).

5.3.1 Implementation

In this work, Web Ontology Language (OWL) 2 is used for implementing the proposed ontology as there are several expressiveness issues and syntax problems in OWL [18]. Class hierarchy, object and data properties, and constraints on classes and properties are created by means of the open source ontology editor Protégé.

Figure 5.2 visualizes an excerpt of the developed ontology. Besides classes, the figure contains instances, data values, and property relations in order to show ontology utilization. Classes are marked with circles, defined classes have three additional lines, and individuals are tagged with a diamond. Data values show the used data type. The illustrated example contains an energy retailer `ER1` that offers the service `EnergyPrice`. Besides the parameter configuration `Config1`, the service is related to a technology adapter `Adapter1` specifying the communication technology for accessing the service. The energy retailer is connected to one of two available, adjacent substations. In order to distinguish between standardized properties and own properties, different namespaces are used (e.g. `sg`, `owl`, `rdf`). Solid lines represent asserted relations, and dashed lines are used for relations that are inferred by the reasoner. The figure is intended to give an overview

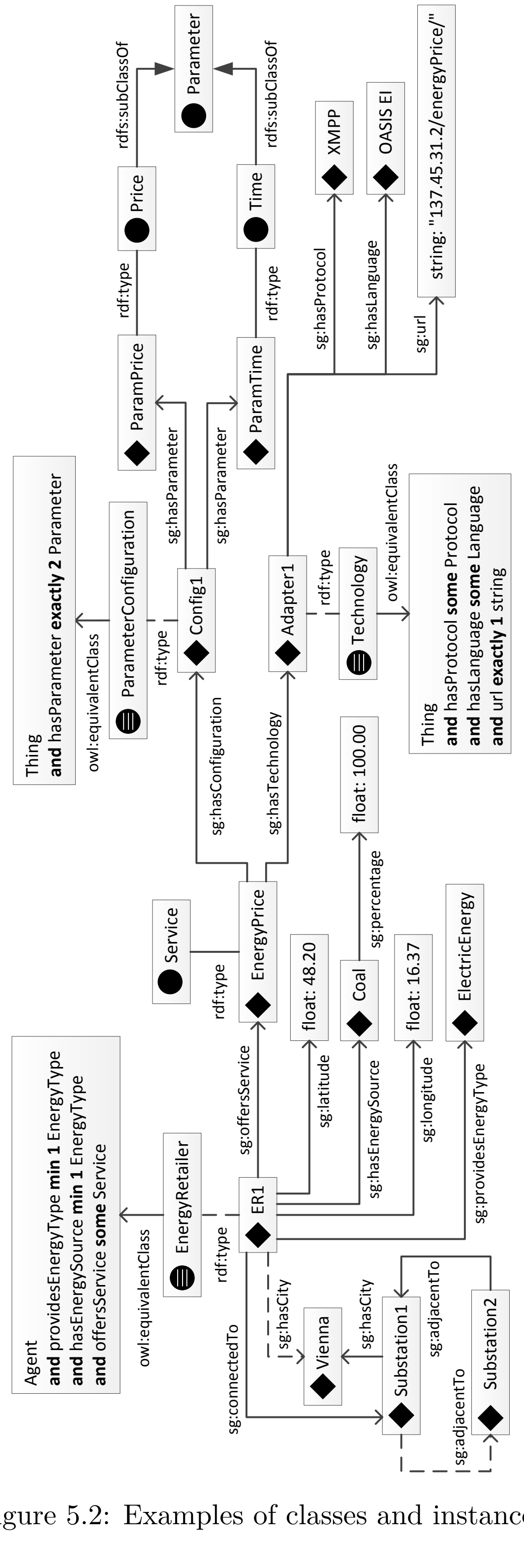

Figure 5.2: Examples of classes and instances

on the modeling of instances based on the defined ontology. Thus, it is not complete, and many classes and properties are omitted. For example, location of one substation is left out although this information can be used to infer further knowledge, such as the distance between grid nodes.

5.3.2 Integration

A BEMS uses the described ontology for identification and assessment of grid services, grid agents, and grid structure. Thus, more advanced applications that exploit this additional knowledge can be built. For this purpose, the BEMS needs to obtain knowledge from external sources in order to create its own notion of the surrounding smart grid. This can be realized by using one or more central knowledge bases that can be accessed by all agents. These knowledge bases concentrate all grid-relevant information and provide it in an abstract form to a BEMS. For example, central servers per country can be operated by the national energy authority, and all publishable information, like agents, grid structure, and provided services, is modeled in this knowledge base. Additionally, information is linked with other knowledge sources realizing the idea of the Semantic Web and the linked open data initiatives. A BEMS can navigate ad hoc through all available knowledge bases without further access information as resources have unique identifiers in the Semantic Web. However, it is also possible to model everything in a local knowledge base of the BEMS without any links to other ontologies. The functionality of the BEMS remains the same as the ontology defines an abstraction layer to underlying technologies as well as to distribution of information. In this work, the popular open source framework Jena is used to store ontology information for integration into a BEMS. The provided APIs are used to query and modify the ontology.

5.4 Evaluation and testing

Depending on the energy management strategy of a BEMS on top of the ontology-based abstraction layer, different knowledge about the smart grid can be gathered to improve operation quality. For example, the nearest energy retailer can be selected, or the various services of a customer's grid operator are analyzed regarding classification into market or physical DR services. When a service needs to be executed by a BEMS, access information within the ontology is used to contact involved agents. Data gained from a service or sent to a service are not stored in the ontology. Here, a hybrid approach is used by defining a relational database schema and linking stored information with semantic information of the ontology. Thus, efficient processing of vast amount of data is ensured, and the ontology is used to enhance semantics of these runtime data. To sum up, the ontology is used to model more static information while runtime data from service execution are separately stored.

In order to evaluate the developed ontology that is embedded into an abstraction layer, a smart grid test bed is used, which is based on the grid structure visualized in Figure 5.1. Two grid operators, one energy retailer, one energy aggregator, two central energy generators, and three customers are implemented on separate Raspberry PI boards. As the focus of this work is on BEMSs of customer buildings, mockup implementations are used for all other agents to simulate service execution. For evaluation, the BEMS has a user interface for accessing the knowledge base and executing smart grid service interaction sequences. The ontology is stored on a distinct Raspberry PI representing the energy authority, and all agents have direct access to this central storage. In addition, agents are equipped with a MySQL database for storing runtime data. In the test bed, the Extensible Messaging and Presence Protocol (XMPP) and OASIS Energy Interoperation (EI) are used for service communication via the communication infrastructure enabling pairwise interaction between agents [5]. In order to test ontology reasoning regarding service compatibility, some services of one grid operator are configured to use Open Automated Demand Response (OpenADR) instead of OASIS EI. Offered and required services are defined on the basis of DR communication patterns introduced in [4]. Power transmission infrastructure is not actually created and is only virtually present in the ontology.

First, modeling capabilities are analyzed by the definition of the smart grid test bed using the developed ontology. Exemplarily, this is shown in Figure 5.2. The location of agents corresponds to their position in Figure 5.1. The power transmission infrastructure is described by some substations, the connected agents, and one high as well as two low voltage grid segments. One grid operator combines both a low voltage grid controller and a flexibility operator while the other grid operator is a pure low voltage grid controller. The former offers services for grid regulation, sending flexibility requests, and sending flexibility orders. On the other hand, this multifunctional grid operator has incoming services for customers' energy consumption forecasts and flexibility offers. Energy aggregator services are similar to flexibility operator services, and the energy retailer provides a service for sending energy price information and requires a service for energy consumption forecasts. Customers define appropriate counterparts of these services. Interaction sequences are specified by defining service chains (e.g. energy consumption forecast is sent after reception of energy price information). Once, all information of the grid is modeled using the ontology concepts, the reasoner is able to infer new knowledge and uncover implicitly available knowledge. Some rudimentary examples are already visualized in Figure 5.2, like class membership or inverse properties. However, also more sophisticated information can be inferred by means of rules. An example is the geographical distance between the location of agents or grid nodes enabling appropriate selection of agents and perception of grid dimensions. Also compatibility rules that are defined during ontology development are applied by the reasoner. Thus, services and their communication technologies can be automatically matched. In the test bed, the reasoner recognizes that some grid operator services use OpenADR as communication language, which does not comply with any customer service based on OASIS EI.

Second, this support of smart grid interaction is pointed out by running through an ordinary DR test case. Energy prices are distributed by the energy retailer, and subsequently each customer sends an energy consumption forecast in response. In order to ease testing, one customer has been selected for interaction with the energy retailer. Based on the information available in the ontology, the BEMS of the customer knows that there is one energy retailer in the grid, which offers a price distribution service (cf. Figure 5.2). Furthermore, the BEMS knows that its own customer has an incoming service for energy price information that is linked with an outgoing service for publication of energy consumption forecasts. The latter uses the parameters energy and time to describe the sent data. Similarly, the energy retailer is defined to receive these forecasts. The reasoner infers that all these services are pairwise compatible as communication technology and language match. Based on this information that can be automatically interpreted by a BEMS, the interaction is started resulting in the execution of the described services. The exchanged information is stored in the local database, and the BEMS links the information of the ontology with this received and sent data. In real-world situations, authentication and authorization for service execution and access to service interfaces are necessary in order to avoid malicious access. For this purpose, standardized and well-proven IT security mechanisms provide protection.

5.5 Conclusion

Interaction between BEMSs and smart grids becomes increasingly important to balance energy demand and supply or to incorporate distributed energy resources. Thus, this work presents an ontology as part of an abstraction layer that enables semantically enriched description of smart grids and supports operation of BEMSs. The development focuses on four main parts covering grid agents, communication technologies, service interfaces, and grid structure. The proposed approach is implemented in the form of an OWL ontology, and its application and integration into BEMSs are described. Furthermore, modeling capabilities, ontology reasoning, and functionality are evaluated using a smart grid test bed.

Further steps are the consideration of additional agent interaction apart from customers and BEMSs and the expansion of rules in order to enhance reasoning. For example, interaction between grid operators and energy retailers can use the ontology-based abstraction layer. Additional evaluation needs to be done including both simulation and real-world applications. Thus, statements regarding scalability, security, or interoperability will be available. Concerning interoperability, additional technology adapters for common smart grid communication languages and protocols (e.g. HTTP, OpenADR) need to be implemented as the current implementation only supports XMPP and OASIS EI. Finally, the developed ontology needs to be linked with further ontologies improving semantics and stimulating progression of the Semantic Web.

Acknowledgment

This work was funded by FFG (Austrian Research Promotion Agency) under the *BProfEM* project (845627).

References

[1] L. Pérez-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information", *Energy and Buildings*, vol. 40, no. 3, pp. 394–398, 2008, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2007.03.007.

[2] H. Farhangi, "The Path of the Smart Grid", *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, Jan.-Feb. 2010, ISSN: 1540-7977. DOI: 10.1109/MPE.2009.934876.

[3] P. Palensky and D. Dietrich, "Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads", *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381–388, Aug. 2011, ISSN: 1551-3203. DOI: 10.1109/TII.2011.2158841.

[4] M. Pichler, "Communication Patterns for Demand Side Flexibility", in *Symposium on Communications for Energy Systems*, Sep. 2014, pp. 19–28.

[5] D. Schachinger, S. Gaida, and W. Kastner, "Smart grid communication at the interface of customer buildings with focus on demand response", in *International Symposium on Smart Electric Distribution Systems and Technologies*, Sep. 2015, pp. 368–373. DOI: 10.1109/SEDST.2015.7315236.

[6] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges", *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 5–20, First Quarter 2013, ISSN: 1553-877X. DOI: 10.1109/SURV.2012.021312.00034.

[7] C. Martinez-Cruz, I. J. Blanco, and M. A. Vila, "Ontologies versus relational databases: Are they so different? A comparison", *Artificial Intelligence Review*, vol. 38, no. 4, pp. 271–290, Dec. 2012, ISSN: 0269-2821. DOI: 10.1007/s10462-011-9251-9.

[8] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart Grid – The New and Improved Power Grid: A Survey", *IEEE Communications Surveys & Tutorials*, vol. 14, no. 4, pp. 944–980, Fourth Quarter 2012, ISSN: 1553-877X. DOI: 10.1109/SURV.2011.101911.00087.

[9] T. G. Stavropoulos, D. Vrakas, D. Vlachava, and N. Bassiliades, "BOnSAI: A Smart Building Ontology for Ambient Intelligence", in *ACM International Conference on Web Intelligence, Mining and Semantics*, Jun. 2012, pp. 1–12. DOI: 10.1145/2254129.2254166.

[10] M. J. Kofler, C. Reinisch, and W. Kastner, "A semantic representation of energy-related information in future smart homes", *Energy and Buildings*, vol. 47, pp. 169–179, Apr. 2012, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2011.11.044.

[11] N. Shah, K.-M. Chao, T. Zlamaniec, and A. Matei, “Ontology for Home Energy Management Domain”, in *Digital Information and Communication Technology and Its Applications*, ser. Communications in Computer and Information Science, H. Cherifi, J. M. Zain, and E. El-Qawasmeh, Eds., vol. 167, Springer Berlin Heidelberg, 2011, pp. 337–347, ISBN: 9783642220265. DOI: 10.1007/978-3-642-22027-2_28.

[12] M. Grassi, M. Nucci, and F. Piazza, “Ontologies for Smart Homes and Energy Management: An Implementation-driven Survey”, in *IEEE Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*, May 2013, pp. 1–3. DOI: 10.1109/MSCPES.2013.6623319.

[13] Q. Zhou, S. Natarajan, Y. Simmhan, and V. Prasanna, “Semantic Information Modeling for Emerging Applications in Smart Grid”, in *International Conference on Information Technology: New Generations*, Apr. 2012, pp. 775–782. DOI: 10.1109/ITNG.2012.150.

[14] M. Uschold and M. Gruninger, “Ontologies: Principles, Methods and Applications”, *Knowledge Engineering Review*, vol. 11, no. 2, pp. 93–136, Jun. 1996, ISSN: 0269-8889. DOI: 10.1017/S0269888900007797.

[15] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, “The NeOn Methodology for Ontology Engineering”, in *Ontology Engineering in a Networked World*, M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi, Eds., Springer Berlin Heidelberg, 2012, ch. 2, pp. 9–34, ISBN: 9783642247934. DOI: 10.1007/978-3-642-24794-1_2.

[16] N. F. Noy and D. L. McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology”, Stanford Knowledge Systems Laboratory, Tech. Rep. KSL-01-05, Mar. 2001, pp. 1–25.

[17] W. Kastner, A. Fernbach, W. Granzer, and M. Jung, “KNX and the Semantic Web of Automation”, in *KNX Scientific Conference*, Nov. 2014, pp. 1–17.

[18] B. C. Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler, “OWL 2: The next step for OWL”, *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6, no. 4, pp. 309–322, Nov. 2008, ISSN: 1570-8268. DOI: 10.1016/j.websem.2008.05.001.

CHAPTER 6

Semantics for smart control of building automation

Abstract: Building automation is an important part of state-of-the-art building management in order to attain most efficient operation in accordance with comfort requirements, energy consumption, or budget allowance. For this purpose, current building management systems enable communication with subjacent systems at the field and automation level by definition of mostly syntactical technology mappings. However, integration of building automation systems for management and control purposes also needs to address the semantics of these subsystems, their cooperation, and their interference. In this work, such an integration approach is presented that enables smart control of building automation resources by the use of semantic technologies. An OWL ontology is developed in order to represent and link knowledge of all relevant domains. Furthermore, an interface concept for seamless and interoperable cross-border communication in the heterogeneous building automation environment is introduced. Finally, an application scenario illustrates the functional capabilities of this approach for smart control in building management.

Keywords: Building automation, building management, control systems, ontology, semantics, system integration

---

The content of this chapter has been published: D. Schachinger and W. Kastner, "Semantics for smart control of building automation", in *IEEE International Symposium on Industrial Electronics*, Jun. 2016, pp. 1073–1078.

6.1 Introduction

A building automation system (BAS) aims at establishing and maintaining safety, security, energy reduction, cost optimization, and comfort in commercial and residential buildings [1]. First and foremost, compliance with comfort constraints of building users should be ensured. However, also energy reduction becomes increasingly important as buildings are responsible for a large portion of total energy consumption [2]. Realization of these functions and compliance with the requirements need integrated, interoperable, and technically mature BASs. Over the last decades, a three-level architecture, which consists of field, automation, and management level, has evolved in building automation (BA). Various BA technologies cover different domains, like heating, ventilation, and air conditioning (HVAC), lighting, or security, on different levels of this hierarchical architecture. Thus, interoperable integration is a key factor in this heterogeneous environment in order to implement efficient monitoring and control in terms of building management systems (BMSs).

In general, technology mappings are used to link BMSs at the management level with field and automation level systems in order to enable communication between disparate technologies for the purpose of satisfying specified requirements. As an example, gateways can be used to integrate isolated subsystems into an overall BAS. While this integration methodology incorporates syntactic interoperability, it often lacks in concepts to model semantics associated with functionality or arrangement of BASs. However, this knowledge forms the necessary prerequisite for smarter control and management of BA. Thus, user interaction in BMSs can be minimized, and the power of semantic technologies is utilized to support highly automated, operational management of buildings.

Initiated by the Semantic Web, ontologies are relatively new in the field of computer science, but they are identified as most promising technology for semantic modeling and reasoning [3]. This accepted fact in combination with other advantages, like the high abstraction level, the implementation independence, or the ability to link and share distributed knowledge among different domains, underpin the utilization of ontologies in the proposed approach for semantics-based control of BA. The semantic information is incorporated into an intermediary layer between BMSs and the subjacent BA resources similar to [4], where integration of smart grids and building energy management systems is described. Hence, the ontology acts as point of concentration for all relevant information while it decouples BASs from BMSs. Consequently, the prevailing heterogeneity in BA can be bypassed besides the ability of semantic modeling of BASs, their services, sensed data, and interactions with the ambient building.

Literature already refers to various, ontology-based modeling approaches for BA. BOnSAI ontology combines existing ontologies, such as CoDAMoS or OWL-S, in order to form a basis for ambient intelligence in buildings [5]. In [6], DogOnt ontology is presented, where controllable and uncontrollable building things can be linked with the surrounding building environment. Moreover, BA resources have associated states and functionality. SOUPA ontology provides concepts for modeling of agents and persons as well as their profiles to support pervasive applications [7]. Integration of heterogeneous building data sources into an ontology for the purpose of performance analysis is presented in [8]. Similarly, [9] shows an ontology-based facility modeling approach aiming at energy management. Here, monitoring signals are enriched with additional semantics. ThinkHome ontology is used to combine information about building, resources, process information, and external influences with energy-related information [10]. A framework to merge ontologies for smart homes and energy management is proposed in [11]. Semantically enriched description of devices in order to ease BAS design and commissioning is addressed in [12]. An automatic design process should be supported by formal and unified device definitions using the specified ontology. These device descriptions are also used in BASont ontology that combines modeling concepts in order to support variable use cases in the BAS life cycle [13].

This work aims at defining a semantic framework to support operational building management. An ontology in combination with a unified communication and management interface provides a common basis for control of BA. Thus, semantic integration of both BA technologies and high-level management applications is realized in an abstract and universal way. The ontology enables a generic but expressive, semantic description of the BAS and the building structure. Energy-related information, a link to the smart grid, or connections to external services, such as weather service providers, are out of scope. However, the framework can be easily extended by these features due to its generic and interoperable structure.

6.2 Architectural concept

Efficient and effective control in modern building management is dependent on high-quality data and information. Maintaining interoperability of heterogeneous BA technologies in order to ensure compatible data transportation is a necessary basis. Moreover, interoperability needs to be addressed on top of the protocol stacks at the information level. Solely, the integration of enhanced semantics regarding resources, functionality, or context information is able to realize a smart, dynamic, and flexible control of BASs. In this context, ontologies provide an adequate methodology to model the semantics and infer new knowledge by the use of reasoners.

In contrast to a traditional, hierarchically oriented integration approach, where the BAS is directly linked to the top-level BMS, this work proposes a semantic layer that is inserted in between. This semantic layer with its ontology and a control interface becomes the central element. On the one hand, this intermediate layer leads to higher abstraction between the connected technologies. On the other hand, the ontology acts as concentration point for information and knowledge exchange. Thus, cross-border links over different domains and technologies can be established, and also knowledge of remote ontologies can be incorporated in this central component.

This concept is illustrated in Figure 6.1. The ontology is part of the semantic layer that can be accessed via a common and uniform interface. Thus, all system components

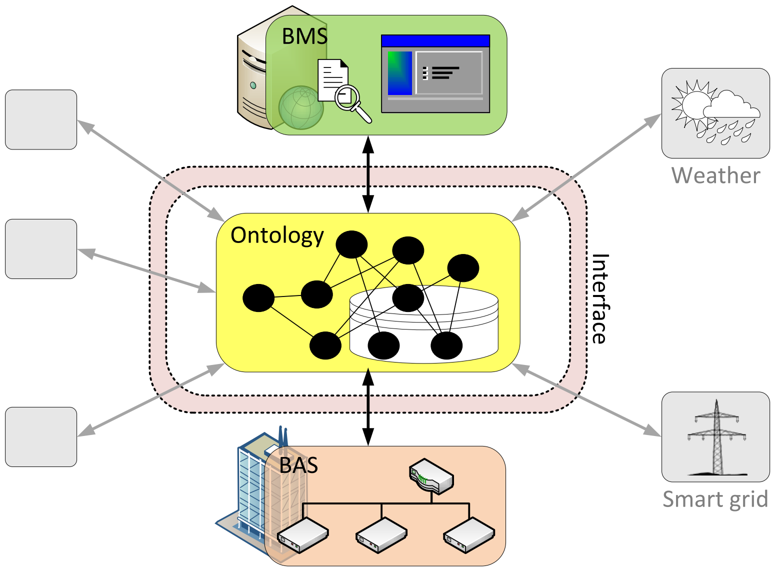

Figure 6.1: Overview

are able to communicate with the knowledge base. Only a technology connector that is familiar with the ontology and its concepts needs to be available to map a certain technology to the proposed interface. Depending on the BAS, control networks at the field and automation level, like KNX or EnOcean, can be linked to the semantic layer, but also management integration solutions, such as Open Building Information Exchange (OBIX), BACnet Web services (BACnet/WS), or OPC Unified Architecture (OPC UA), can be connected. Information from the BAS is integrated into the ontology, and the BMS is able to exploit this information in order to decide on ideal control strategies. Although direct communication between BAS and BMS is prevented, control commands are set in the semantic layer, and the BAS is informed about new settings. This is possible as the control interface provides active mechanisms while the internal ontology is only a passive component.

As this work is focused on smart control of BA by means of semantic technologies, BMS and BAS are the essential parts besides the semantic layer with its ontology and control interface. However, this approach can be easily extended in order to incorporate other components, such as smart grid agents and their communication services or weather service providers, which use standardized Web service technologies. In Figure 6.1, this can be seen by the gray components. Furthermore, it is possible to directly link BA subsystems with the semantic layer's interface. Thus, the challenge of interoperable integration of these BAS parts is shifted to the semantic layer and needs not be handled on a lower level. However, gateway technologies can still be used. Functional bindings of components from different technologies are realized in these gateways, and the overall semantic information is integrated into the ontology. In summary, this setup ensures flexibility in system design and combines all relevant, semantically enriched information in a centrally accessible knowledge base.

6.3 Ontology development

The ontology aims at combining information of the building, the BA resources, and the data and control services on an abstract but semantically enriched level. Well-defined ontology engineering methodologies are analyzed in order to select a suitable procedure for implementing the proposed ontology. In [14], Ontology Development 101 guide is specified. At first, the scope is determined, and reuse of existing ontologies is considered. Based on an enumeration of important terms, the classes, properties, and constraints are defined before individuals are created. A skeletal methodology for ontology creation is presented in [15]. Disregarding different naming of process steps and slightly different ordering of tasks, both methodologies possess similar workflows. On the contrary, NeOn methodology describes a set of different engineering processes depending on the availability of existing resources and their reuse [16]. The development of the proposed ontology is mainly based on Ontology Development 101, but principles of [15] are also taken into account.

As noted in the introduction, there exist ontologies to describe BA resources as well as building structures. Examples are DogOnt [6], BASont [13], or ThinkHome [10]. These ontologies are used as basis for development of the proposed smart control ontology, which enables the representation of relevant elements on an abstract level in order to support smart and automated building management for meeting comfort or operational constraints. The control tasks in building management can be generalized to a variation of parameters, such as temperature, humidity, or brightness, by the execution of available BA services. This variation can be triggered by analysis of sensed data. In order to create a shared understanding of concepts, the ontology also needs to be aligned with other ontologies by defining links between corresponding concepts. In the following, the four main parts of the resulting ontology are described in more detail.

1) Building structure: In the proposed ontology, the class *Zone* forms the main element to describe building structures. Zones are arranged to each other in a relative fashion and can be composed of smaller zones. Well-known concepts, like *Site*, *Building-Part*, *Floor*, or *Room*, are special zones with certain characteristics. For example, a zone of type site forms the topmost element in a particular building structure and consists of building parts. Site, building part, floor, and room are divided into several subclasses (e.g. *Building*, *OfficeBuilding*, *Bathroom*, *Corridor*, *Attic*). The elements of this taxonomy can be linked in order to form hierarchical building structures. The used properties are *hasZone* and its inverse *isZoneOf*. Moreover, there are several object and data properties used to describe instances of these concepts in more detail. Examples are *name*, *latitude*, *longitude*, *address*, *hasCity*, or *hasOwner*. In order to enhance compatibility, the building structure classes are aligned with similar classes of other ontologies. For example, site is equivalent to *IfcSite* of IFC ontology, or room is equivalent to *Room* of ThinkHome ontology.

The building arrangement is primarily based on zones (*Zone*) and zone delimiters (*ZoneDelimiter*). A delimiter (e.g. a *Wall*) has data properties for *material*, *transparency*, and *thermalPermeability*. The properties *isRightOf*, *isLeftOf*, *isInFrontOf*, *isBehind*, *isAbove*, and *isUnder* are used to order all available zones and delimiters into an alternating, relative structure. Two zones are adjoined if they share a zone delimiter. Outer walls of a building are explicitly modeled and can be linked to an orientation (*hasOrientation*). Hence, the orientation of other outer walls can be inferred leading to the absolute orientation of the building. BA resources and other devices can be assigned either directly to a zone (*isLocatedIn*) or to a zone delimiter by using the introduced ordering properties. Then, the zone membership of a device can be easily inferred.

2) Devices and appliances: Devices (e.g. sensors, actuators) within a building need to be modeled as they host data and control services. In this ontology, the class *BuildingResource* is introduced as top-level concept. As devices are merely bridging the gap between the building structure and the offered services, a precise description is not needed here. However, concepts of existing ontologies can be used if modeling of additional device and appliance characteristics is desired. As an example, the class *Controllable* of the DogOnt ontology is linked with the building resource class. Moreover, *Controllable* is already aligned with *Device* from SSN ontology. As a result, DogOnt and SSN concepts can be used to describe controllable devices if necessary.

3) Data services: A *DataService* represents data that is made available by a service provider (*provides*). In general, providers are BA devices, and data services mark output datapoints of these devices. For data description, one or multiple elements of type *ParameterConfiguration* are used (*hasConfiguration*). Each parameter configuration has exactly two parameter types (*hasParameter*) describing the actual data values. Examples for parameter types are *Temperature*, *Time*, *Humidity*, *CO2Level*, or *Precipitation*. Figure 6.2 shows a temperature data service (`TempData1`) provided by a temperature sensor device. Its parameter configuration refers to a `Temperature` and a `Time` parameter. Thus, data values of this service are temperature values with corresponding instants of time. The location of a service gives further information about the measurement context. The history of data provided by these services can be used in BMSs to calculate forecasts of value progressions. Thus, comfort requirements are implicitly available, and explicit description of user preferences or comfort profiles can be omitted in this ontology.

In order to describe the access mechanism to the BAS, a *TechnologyConnector* is used, which is of a particular technology type (e.g. *KNXConnector*, *OBIXConnector*). For identification purposes, it includes the address of the connector component, for example its IP address (*connectorAddress*). Each data service can have at most one technology connector. Moreover, a data service is characterized by a *serviceAddress* that is unique within the associated technology connector. The KNX group address or a relative OBIX path can be specified using this property. Other technology characteristics, such as protocol or message encoding, are not modeled as this communication is hidden by the used technology connector.

4) Control services: A *ControlService* represents an active element that can influence the mentioned parameters. Like data services, control services are offered by service providers (e.g. devices), and they represent input datapoints of these providers. Switching actuators, HVAC controllers, or shutter actuators are examples for such providers. A control service can trigger another resource or equipment (*triggers*). Again, technology connectors are used to model access information. Each control service is connected to a hard or soft data service, which can be used to access the history of control values and to set new control values (*hasControlData*).

Generally, a control service can influence parameters by changing the state value of BA resources. Possible parameter variations are modeled as *ParameterVariation* that describes a distinct parameter (*hasParameter*), a variation trend (*hasTrend*), and a relative state value change (*hasOrder*). Variation trends indicate whether the value of the specified parameter will be increased (*UpTrend*) or decreased (*DownTrend*) in accordance with the state value change (*HigherValue* or *LowerValue*). Parameter variations are linked with conditions (*VariationCondition*) that specify comparisons of constants, values of data services, or state values of control services. Finally, the control services have state values (*StateValue*) that are subdivided into *DiscreteStateValue* and *IntervalStateValue*. Moreover, a state value defines absolute (*AbsoluteStateValue*) or relative values (*RelativeStateValue*). It is important to provide an ordering of all state values to decide on value changes considering the parameter variations (*isHigherThan*, *isLowerThan*).

The ontology is based on the Web Ontology Language (OWL) 2, and Protégé ontology editor is used for implementation of the developed classes, properties, constraints, rules, and instances. Only a subset of all classes and properties is discussed in this section. To sum up, Figure 6.2 shows a modeling example. Classes are marked with circles, defined classes have three additional lines, and individuals are tagged with a diamond. A building consisting of two rooms (`RoomA`, `RoomB`) provides the basis. One temperature sensor (`TempOut`) is placed on an outer wall, and a second temperature sensor (`TempIn`) is located in `RoomA`. Moreover, an electric radiator (`Radiator`) is available in `RoomA`. While the temperature sensors provide data services for the temperature values, the radiator offers a control service `Heating` to influence the temperature within the room. Dashed lines are used for relations that are inferred by the reasoner while solid lines represent asserted relations. The namespace `sc` is used for all properties of this smart control ontology. For the sake of clarity, modeling of some classes, properties, and instances is omitted in order to keep the figure as clear as possible.

6.4 Control interface

As already described in the architecture (see Section 6.2), the ontology is surrounded by a control interface that manages ontology access of all connected components, such as a BAS or a BMS. Technology connectors bridge the gap between systems in the building and the semantic layer with its ontology. In particular, technology connectors communicate with the control interface using well-defined mechanisms. On the other hand, the technology connector manages the access with the system behind. Basically, the interface needs to provide functions for getting and setting data values of BA resources.

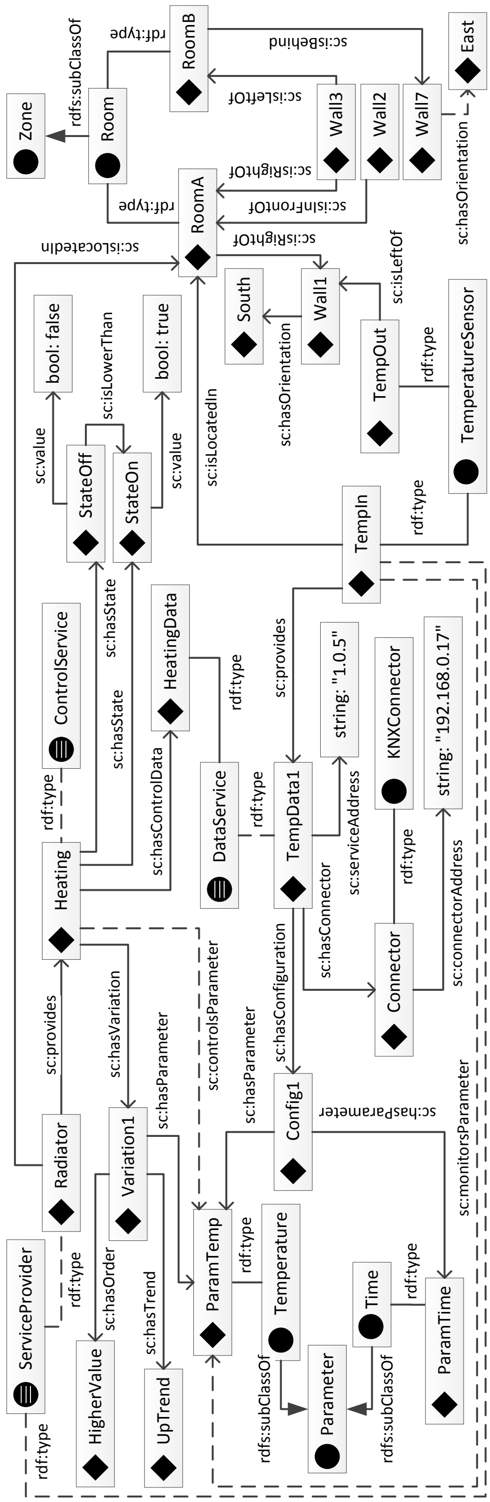

Figure 6.2: Modeling example

On the contrary, BMSs require a more sophisticated interface for querying and updating information in the ontology.

In the field of BAS integration, technologies based on Web services (WSs) can be used to provide interoperability in heterogeneous BA environments as they define object models enabling the description of BA resources in a technology-independent way. Although they are eligible for application in the BA domain, Semantic Web technologies, like the Resource Description Framework (RDF), are better suited for representation of semantic information in the context of ontologies. An interface specification based on these technologies is going to ease and unify access to the proposed ontology.

The selection of an appropriate transportation protocol is influenced by the necessity of bidirectional communication between the interface of the semantic layer and the various technology connectors. Otherwise, polling has to be used or advanced alarming and watch mechanisms are needed as workaround. Therefore, the WebSocket protocol is chosen as it enables full-duplex communication. Value changes do not need to be polled, but update messages can be actively pushed to the affected communication partner. On top of the WebSocket protocol, a closed set of methods is defined for interaction with the ontology.

The basic methods `GET` and `PUT` are used to read and write values of data services. As already mentioned, also the state values of control services are linked to data services. The `GET` method requires a uniform resource identifier (URI), which is equivalent to the URI of the particular data service in the ontology. The content of a `GET` call is empty while the response contains RDF triples stating the requested values of a data service in accordance with its parameter configuration. For example, a `GET` request for the latest value of a data service with a temperature/time parameter configuration will return two RDF triples. One triple contains the temperature value while the second one gives the related instant of time.

RDF/XML or Turtle can be used to encode the triples. On the other hand, `PUT` contains RDF triples in the request's content while the response is an empty message. Here, the given URI specifies the destination data service for the new values that are encoded in the message content. If neither the URI exists nor the content conforms to the parameter configuration, the call fails. `PUT` can be used by the control interface to push a new value to a BA resource if the BMS has set a new value in order to achieve a parameter variation. Moreover, this method is used by technology connectors to store observed values of the BAS into the ontology. The most powerful method is `POST` as entire SPARQL or SPARUL queries can be sent using the encoding of the SPARQL 1.1 Protocol. The response consists of the query result set, which is encoded using SPARQL XML. Although the connectors can be supported by simpler APIs in order to ease querying, the messages that are exchanged between the distributed connectors and the control interface contain complete SPARQL or SPARUL queries.

While the control interface of the semantic layer provides full support of `POST` requests, technology connectors are not forced to understand SPARQL and SPARUL queries.

Nevertheless, all implemented technology connectors must be able to handle `GET` and `PUT` requests. Hence, necessary functionality to exchange basic sensor, actuator, and controller data between the ontology of the semantic layer and connected BASs is guaranteed. By the use of Semantic Web technologies in combination with simple methods on top of a bidirectional communication protocol, the control interface is able to interact with both BASs and BMSs. Moreover, other systems can be linked with the semantic layer by implementing an appropriate technology connector that conforms to this proposed control interface. An example would be an Open Automated Demand Response (OpenADR) connector for smart grids.

6.5 Application scenario

Semantics facilitates a smarter, more dynamic management and control of BASs. BMSs do not need to access BA resources directly, but they make use of an intermediate semantic layer that provides shared knowledge on an abstract, interoperable level. As can be seen in Figure 6.1, the semantic layer becomes the central element in control workflows. On the one hand, the technology-independent description of building structures, devices and appliances, data services, and control services helps BMSs to decide on strategies for meeting given constraints. Although this work is focused on comfort constraints in the form of perceptible parameters, such as temperature or brightness, cost or energy constraints can also be included as all these constraints are mainly modeled using the common notion of data services. Utilization of advanced data analysis in the BMS is able to identify patterns and constraints that are available in the value histories of data services. On the other hand, BASs push sensed values and receive control commands via the well-defined interface.

Based on the modeling example illustrated in Figure 6.2, an exemplary control workflow shows the application of the proposed approach. The setting is based on the building structure and the integrated BA resources shown in Figure 6.3. An EnOcean outdoor temperature sensor (*TempOut*) and a KNX heating actuator (*Radiator*), which is able to influence the temperature in *Room A*, are integrated into an OBIX gateway. A KNX shutter actuator (*Shutter*) to control the blinds on the south side of *Room A* and a KNX component to control inlet air ventilation (*Ventilation*) are also connected to the OBIX gateway. The shutter control service can influence brightness and temperature, and the ventilation is able to influence air quality (i.e. CO_2 level) and temperature. The KNX indoor temperature sensor (*TempIn*) is directly accessed via a KNX connector based on Calimero. The BMS has an integrated connector implementing the control interface. The circles mark the installation places of the devices. OBIX and KNX technology connectors as well as the semantic layer and the BMS are realized on separate Raspberry PI boards forming a local IP network. The open source framework Jena is used as storage for the developed OWL ontology. An additional MySQL database stores the data values linked to data services (e.g. actual set of temperature values) similar to [4].

First, the BAS and BMS technology connectors initialize a WebSocket connection to the semantic layer component. Sampled temperature values from outdoor and indoor

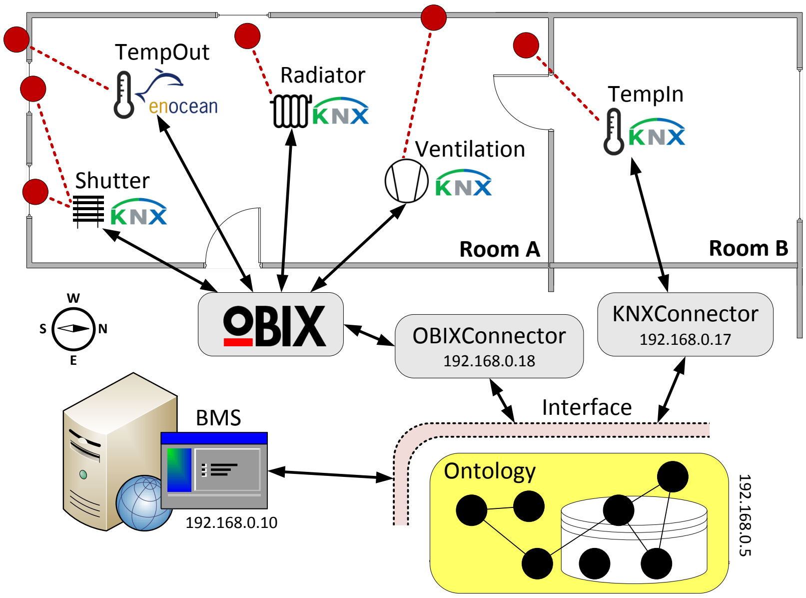

Figure 6.3: Application scenario

temperature sensors are pushed to the ontology by sending `PUT` messages that include the data service URI (e.g. `http://auto.tuwien.ac.at/sc#TempData1`) as well as RDF triples for the sensed temperature value and the corresponding instant of time (e.g. `sc:ParamTemp sc:value "19.3"^^xsd:float`). The interface of the semantic layer receives such messages, fetches the associated data service, and stores the values into the database, which is linked to the ontology. Based on the history of indoor temperature values, the BMS application is able to forecast temperature constraints. In this scenario, the BMS detects that the temperature has to be raised in order to satisfy the upcoming needs. Thus, the BMS checks the available options to increase the temperature in *Room A*. For this purpose, a SPARQL query is sent via a `POST` message. It is assumed that the outdoor temperature is below the indoor temperature, there is high solar irradiation on the south side of *Room A*, and the shutter is already in up position. Based on these facts, neither ventilation nor shutter control services can be set in a way that the indoor temperature is raised. This can be concluded by the BMS based on the available context information in the ontology. Only the radiator's heating service offers a feasible solution. Hence, its state is set to *StateOn*, which is done by adding the control value (*true*) and the current time to the heating data service (*HeatingData*). The control interface is able to detect updates in data services, and thus sends the new control value to the OBIX connector by using a `PUT` message. Also, the new temperature setpoint is forwarded. Then, the KNX heating actuator is activated and adjusts the temperature accordingly. While basic control is still part of existing controllers in the field and automation level, this smart control approach enables autonomous decisions on a higher level of abstraction using the present semantics. Otherwise, such procedures need to be preconfigured in a BMS.

By the use of reasoners, new knowledge can be inferred. Some examples are already shown in Figure 6.2. Moreover, reasoning at the interface of building structure and control as well as data services offers high potential to reveal already available, implicit knowledge. An example is the sphere of influence of control services. The building structure defines zones and zone delimiters as well as actuating places of control services within this setting. Depending on the absolute orientation of the building, material and dimension of zone delimiters, or efficiency and type of service execution, the influence of control services on adjoined or nearby zones can be estimated by reasoners using a set of rules. A BMS will benefit from this additional knowledge, which highlights the use of semantics in this field of application.

6.6 Conclusion

Semantic technologies create new opportunities in efficient management and control of buildings and their automation systems. Thus, this work presents an integration approach based on an abstract, semantic layer. An ontology as part of this layer enables semantic modeling of building structures, BA resources, data services, and control services. Based on this knowledge, BMSs can dynamically develop advanced and smart control strategies. An interface concept is specified in order to uniform the access to the semantic layer, which becomes the central, intermediate element between BASs, BMSs, and other components. Furthermore, application of this approach is demonstrated in a simplified scenario.

Further steps are the implementation of additional technology connectors and the evolution of ontological concepts. The set of rules for inferring new knowledge by the use of reasoners needs to be expanded, and the expressiveness regarding description of control services should be extended. Currently, only KNX and OBIX are supported for BAS integration, but other BA technologies should be incorporated by developing suitable connectors, as well. Additionally, a connector and some ontology adaptions to enable smart grid communication with energy retailers or grid operators are required. Furthermore, tests regarding functionality and performance need to be done with respect to common BMS control tasks. This includes the evaluation of response times for ontology reading and writing, reasoning, and querying. Finally, the developed ontology would benefit from links with further ontologies in this field of application.

Acknowledgment

This work was funded by FFG (Austrian Research Promotion Agency) under the *BProfEM* project (845627).

References

[1] D. Dietrich, D. Bruckner, G. Zucker, and P. Palensky, “Communication and Computation in Buildings: A Short Introduction and Overview”, *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3577–3584, Nov. 2010, ISSN: 0278-0046. DOI: 10.1109/TIE.2010.2046570.

[2] L. Pérez-Lombard, J. Ortiz, and C. Pout, “A review on buildings energy consumption information”, *Energy and Buildings*, vol. 40, no. 3, pp. 394–398, 2008, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2007.03.007.

[3] C. Martinez-Cruz, I. J. Blanco, and M. A. Vila, “Ontologies versus relational databases: Are they so different? A comparison”, *Artificial Intelligence Review*, vol. 38, no. 4, pp. 271–290, Dec. 2012, ISSN: 0269-2821. DOI: 10.1007/s10462-011-9251-9.

[4] D. Schachinger, W. Kastner, and S. Gaida, “Ontology-based abstraction layer for smart grid interaction in building energy management systems”, in *IEEE International Energy Conference*, Apr. 2016, pp. 1–6. DOI: 10.1109/ENERGYCON.2016.7513991.

[5] T. G. Stavropoulos, D. Vrakas, D. Vlachava, and N. Bassiliades, “BOnSAI: A Smart Building Ontology for Ambient Intelligence”, in *ACM International Conference on Web Intelligence, Mining and Semantics*, Jun. 2012, pp. 1–12. DOI: 10.1145/2254129.2254166.

[6] D. Bonino and F. Corno, “DogOnt - Ontology Modeling for Intelligent Domotic Environments”, in *The Semantic Web - ISWC 2008*, ser. Lecture Notes in Computer Science, A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, Eds., vol. 5318, Springer Berlin Heidelberg, 2008, pp. 790–803, ISBN: 9783540885634. DOI: 10.1007/978-3-540-88564-1_51.

[7] H. Chen, F. Perich, T. Finin, and A. Joshi, “SOUPA: Standard Ontology for Ubiquitous and Pervasive Applications”, in *Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, Aug. 2004, pp. 258–267. DOI: 10.1109/MOBIQ.2004.1331732.

[8] E. Corry, P. Pauwels, S. Hu, M. Keane, and J. O’Donnell, “A performance assessment ontology for the environmental and energy management of buildings”, *Automation in Construction*, vol. 57, pp. 249–259, Sep. 2015, ISSN: 0926-5805. DOI: 10.1016/j.autcon.2015.05.002.

[9] N. M. Tomašević, M. Č. Batić, L. M. Blanes, M. M. Keane, and S. Vraneš, “Ontology-based facility data model for energy management”, *Advanced Engineering Informatics*, vol. 29, no. 4, pp. 971–984, Oct. 2015, ISSN: 1474-0346. DOI: 10.1016/j.aei.2015.09.003.

[10] M. J. Kofler, C. Reinisch, and W. Kastner, “A semantic representation of energy-related information in future smart homes”, *Energy and Buildings*, vol. 47, pp. 169–179, Apr. 2012, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2011.11.044.

[11] M. Grassi, M. Nucci, and F. Piazza, "Ontologies for Smart Homes and Energy Management: An Implementation-driven Survey", in *IEEE Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*, May 2013, pp. 1–3. DOI: 10.1109/MSCPES.2013.6623319.

[12] H. Dibowski and K. Kabitzsch, "Ontology-Based Device Descriptions and Device Repository for Building Automation Devices", *EURASIP Journal on Embedded Systems*, vol. 2011, pp. 1–17, Dec. 2011, ISSN: 1687-3955. DOI: 10.1155/2011/623461.

[13] J. Ploennigs, B. Hensel, H. Dibowski, and K. Kabitzsch, "BASont - A modular, adaptive Building Automation System Ontology", in *Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2012, pp. 4827–4833. DOI: 10.1109/IECON.2012.6389583.

[14] N. F. Noy and D. L. McGuinness, "Ontology Development 101: A Guide to Creating Your First Ontology", Stanford Knowledge Systems Laboratory, Tech. Rep. KSL-01-05, Mar. 2001, pp. 1–25.

[15] M. Uschold and M. Gruninger, "Ontologies: Principles, Methods and Applications", *Knowledge Engineering Review*, vol. 11, no. 2, pp. 93–136, Jun. 1996, ISSN: 0269-8889. DOI: 10.1017/S0269888900007797.

[16] M. C. Suárez-Figueroa, A. Gómez-Pérez, and M. Fernández-López, "The NeOn Methodology for Ontology Engineering", in *Ontology Engineering in a Networked World*, M. C. Suárez-Figueroa, A. Gómez-Pérez, E. Motta, and A. Gangemi, Eds., Springer Berlin Heidelberg, 2012, ch. 2, pp. 9–34, ISBN: 9783642247934. DOI: 10.1007/978-3-642-24794-1_2.

CHAPTER 7

Semantic interface for machine-to-machine communication in building automation

Abstract: Current trends and advancements in the Internet of Things and the Semantic Web have already found their way into the domain of building automation. As machine-to-machine communication and integration of heterogeneous building automation technologies are of increasing importance, interoperability is a necessary precondition. In order to support building automation communication, a customized set of services needs to be available. Additionally, semantics of exchanged information has to be described in a machine-readable way to enable automatic interpretation of message contents. In this work, an interface based on Web technologies and Semantic Web standards is presented, which supports platform-independent machine-to-machine communication for building automation. A requirements analysis for such an interface leads to the definition of a service-oriented architecture. The semantics of exchanged message contents is described in an ontology that provides the basis for a common understanding. Moreover, feasibility and hardware requirements of the proposed approach are evaluated.

Keywords: Building automation, machine-to-machine, semantics, service-oriented architecture, system integration

The content of this chapter has been published: D. Schachinger and W. Kastner, "Semantic interface for machine-to-machine communication in building automation", in *IEEE International Workshop on Factory Communication Systems*, May 2017, pp. 1–9.

7.1 Introduction

In the last decades, the building automation (BA) domain was subject to major changes. New services evolved and communication structures were modified while the overall goal of making processes more efficient retained [1]. Still, the basic functions are cost optimization, reduction of energy consumption, increased security, compliance with safety requirements, and maintenance of comfort [2]. In recent years, especially energy efficiency became one of the most important tasks as buildings account for approximately 35% of global energy demand in 2010 with a continuous gain [3]. On the other hand, the effort to meet comfort requirements increased as more and more time is spent within buildings. BA needs to find a compromise between these competing goals. Vertical and horizontal system integration is a key enabler to solve these tasks [1]. With the vision of the Internet of Things (IoT), integration efforts are boosted once more leading to complex and pervasive networks of formerly independent and enclosed systems. Applications in the IoT are intended to influence domains like power infrastructure, factories, logistics, communities, homes and buildings, or health care [4]. Thus, devices in our buildings get progressively interconnected with their environment offering a great potential for the realization of classical BA tasks and emerging use cases in the IoT context.

For this purpose, interoperable communication between formerly separate building automation systems (BASs) is inevitable. Machine-to-machine (M2M) communication needs to be established in order to enable data and information exchange at the device level. The communication partners interact in an autonomous way without additional human intervention [5]. Interconnection, networking, and remote control of machines, i.e. devices, should be enabled by means of scalable, reliable, and low-cost technologies [6]. This corresponds with the things-oriented view of the IoT paradigm introduced by Atzori et al. [7].

In order to select such a scalable and reliable technology as basis for M2M communication, the Internet protocol suite with its platform-independent standards and technologies can be utilized. This is driven by the ongoing trend to reuse existing technologies instead of developing new communication technologies from scratch [1]. Service-oriented architectures (SOAs) on top of standardized technologies using the Internet Protocol (IP) provide a suitable solution for M2M communication [8]. Autonomy, adaptability, flexibility, and interoperability are coupled with a certain level of abstraction and reduced development costs [9]. For example, Web services (WSs) as realizations of the SOA concept are commonly used in literature for integration of BASs into the Internet or the IoT [10, 11, 12]. In general, there are two types of WS architectures. While WS-* is preferred for integration of enterprise applications, Representational State Transfer (REST) has a focus on resources that can be accessed via a uniform resource identifier (URI) using a small set of verbs [13]. This utilization of SOAs and WSs for M2M communication is perfectly in line with the Internet-oriented view of the IoT paradigm [7].

Regarding the increasing number of integrated BA technologies and the progressive complexity of management applications exploiting the interconnected devices of BASs, explicit and machine-processable definitions of underlying semantics are an essential part in order to successfully implement M2M communication. A popular way of modeling semantics is the use of ontologies known from the Semantic Web [14]. By means of standards like the Resource Description Framework (RDF) or the Web Ontology Language (OWL), a structured and machine-readable representation of semantics can be defined leading to a common understanding of a certain domain. BA ontologies are subject of current research in this area. For example, semantics in the smart home context is specified in the ThinkHome ontology [15]. Use cases over the life cycle of BASs are addressed in the BASont ontology [16]. In [17], the performance of building energy management is modeled taking into account building information modeling (BIM). With respect to M2M, the M3 ontology provides a uniform way to describe data of different domains [18]. Similarly, an ontology-based mapping approach to share information in M2M environments is presented in [19]. The focus on representing the meaning of data conforms to the semantic-oriented view of Atzori et al. [7].

Taking into account already available solutions in this field of application, this work presents a semantic interface for platform-independent M2M communication in the BA domain. The aim is to introduce a novel approach on the basis of existing technologies and standards of the Web and the Semantic Web. Although Web communication protocols have several advantages (e.g., scalability, reliability, security), many of them rely on request-response architectures, which complicates bidirectional message exchange. Additionally, semantics is often directly encoded in high-level application programs, or the representation does not meet demands of standardized, interoperable, and automatically interpretable descriptions. Thus, an ontology is set as basis for the interface in order to overcome issues regarding semantics. Combined with an elaborated service set for message exchange, the proposed M2M setting is defined.

In Section 7.2, the basic requirements of semantic M2M communication in the field of BA are addressed. Based on these results, the interface is defined in Section 7.3 including the analysis of communication protocols, the specification of the service set, and the introduction of the ontology. Section 7.4 discusses the feasibility as well as the hardware requirements of the approach. Finally, Section 7.5 concludes the paper and gives an outlook on future work.

7.2 Requirements of M2M communication

In general, M2M communication in BA is faced with several requirements relating to the architectural design (Section 7.2.1) and the set of supported services (Section 7.2.2). Moreover, semantics in communication is important in order to enable automatic information processing (Section 7.2.3).

7.2.1 Architectural needs

An appropriate M2M communication architecture is required due to the potentially high number of connected devices and the consequent increase in network traffic. Thus, the architecture has to deal with latency, bandwidth, reliability, power consumption, mobility, scalability, or security [5]. More requirements on M2M communication are specified in [20]. Although these features depend on the actual application, technologies of the Internet protocol suite already comply with most of these issues. Reuse of standards also increases interoperability and lowers development costs. Regarding latency, requirements of BA are moderate as there are no hard deadlines compared to safety-critical applications. For exchange of simple sensing and actuating information, bandwidth can be low [5]. In case of messages with higher payload size, bandwidth of utilized Web infrastructure is sufficient. Similarly, reliability, scalability, and security can be ensured relying on suitable Web protocols. Mobility is provided by using wireless technologies compatible with the Internet protocol suite. In this M2M communication design, power consumption is neglected under the assumption that all communicating devices have unrestricted access to power sources. In summary, utilization of well-established Web technologies is sufficient for M2M communication in the BA domain.

7.2.2 Supported service categories

Based on the architectural design, application layer services can be identified that are relevant for interoperable M2M communication. In particular, six essential service categories are determined as illustrated in Figure 7.1. By the use of these services, BASs are able to interact with high-level management applications like building management systems (BMSs) in terms of vertical integration. On the other hand, the services enable the communication of BA devices or BASs in the context of horizontal integration. The focus of the proposed semantic interface is on bidirectional peer-to-peer communication without support of multicasting or broadcasting. It has to be noted that the visualized BAS topology is simplified and does not claim general validity. The purpose of each service category is introduced in the following enumeration.

1. *Identification services* are necessary to check the communication partners' identity. Coupled with authentication and authorization, access control can be realized in order to restrict the use of particular services or prevent communication with invalid systems. For example, a communication partner may not be allowed to send query messages to its counterpart.

2. *Publication services* and all following services can only be executed after a successful identification. The publication service is used to inform communication partners about supported BA functions. For example, semantics about the available sensing and actuating functions are communicated via this service. Thus, the publication services provide the basis for subsequent exchange of process data like sensor values or set points.

3. *Observation services* are used to subscribe for a particular BA function. If a communication partner publishes, for example, a sensing function for indoor temperature, value changes can be observed. In order to register for updates, these observation

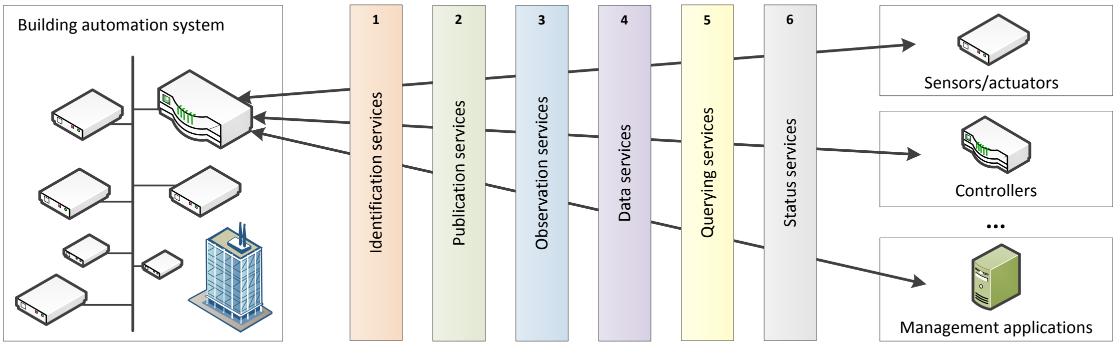

Figure 7.1: Abstract service categories for semantics-based M2M communication in the BA domain

services are used. Therefore, no polling is necessary, and update messages can be pushed to the receiver.

4. *Data services* are the most regularly used services. They are intended for the exchange of process data like humidity values sensed by a BA device. Transmission of sensor data via data services is triggered by a subscription using an observation service or a preceding read request. Similarly, new actuator set points are pushed based on the registered observations or by a read request, respectively.

5. *Querying services* bridge the gap between services of the Semantic Web and traditional M2M communication in the BA domain, which is characterized by the exchange of process or monitoring data. Thus, context information can be requested or updated, which is not possible by means of the other service categories.

6. *Status services* support error handling and acknowledgments, which is necessary for a successful M2M communication. Error codes and other transmitted information need to be machine-interpretable as well.

In summary, this service set covers the common operational interaction patterns in BA while configuration and engineering services are not considered. Although this can also be done with existing M2M communication technologies, this work goes one step further by introducing additional machine-readable and distributed semantics. Thus, it can be assumed that communication partners gain knowledge about the published functions and the particular context of device operation within a building.

7.2.3 Considerations regarding semantics

The previously identified services are the precondition for the proposed semantic interface in order to realize novel M2M communication. However, a framework for semantic modeling of context information, such as the ambient building structure or the description of actuating states, is required. The most relevant concepts, which help machines to

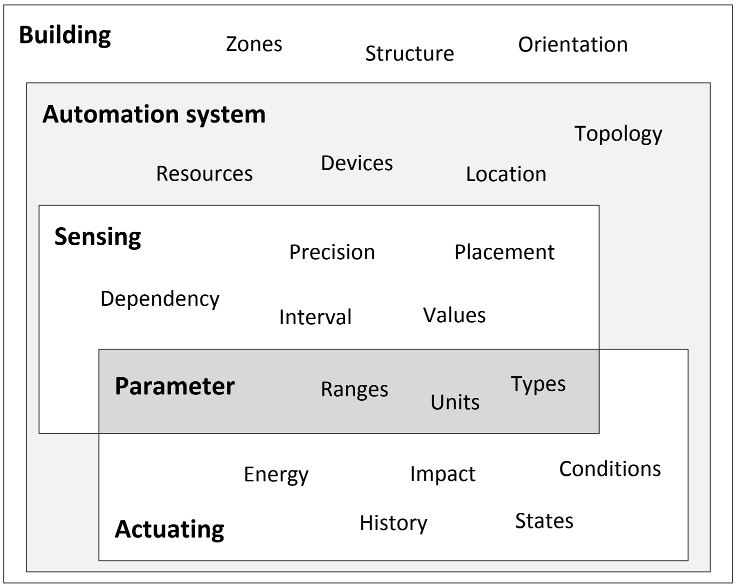

Figure 7.2: Basic domains for semantic modeling

interpret data and information in the BA context, have to be elaborated. The resulting modeling domains can be structured hierarchically as illustrated in Figure 7.2.

As BA is primarily intended to automate processes in building operation with a focus on cost or comfort [2], the *building* context is the outermost domain. Here, concepts to model zones and zone arrangements are required. Zones are used to model the building's physical elements (e.g., floors or rooms) as well as virtual areas (e.g., a part of an open-plan office). Moreover, the orientation of a building is important for interpreting location of BA devices or sensed values.

An *automation system* like a BAS is placed within a building. There are passive resources that are influenced by devices, which form the active elements of a BAS including sensors, actuators, and controllers. These devices are linked in a network topology in order to realize certain BA functions. Resources and devices are located somewhere in the building.

In general, there are two universal types of BA functions that are able to interact with the environment, i.e. they capture and control the situation within the building. On the one hand, *sensing* reads the states of environmental parameters, such as temperature, humidity, or brightness. In this context, the location of the sensor, its precision, the sensing interval, or dependencies to other sensors and actuators are important to finally interpret the monitored values. On the other hand, *actuating* influences the states of environmental parameters. For example, a dimming actuator is able to change the brightness in a particular room. Modeling the impact of an actuation helps characterizing the influence on the environment. In addition, the discrete and continuous states of set points are required in order to define the scope of action. Conditions are used to specify limitations on actuation functionality. Finally, the energy consumption profile and the history of set point changes give additional information about the function.

Sensing and actuating functions actually interact with the environment. In this context, *parameters* are monitored and controlled, respectively. Without additional semantics, the sensed values and written set points cannot be interpreted. Thus, the parameter type, its unit, or the allowed value ranges have to be characterized.

In addition to these domains shown in Figure 7.2, semantics about error and response codes is needed. Information about the identity of the communicating device or system is modeled, as well. All in all, semantics is subject to changes, which requires an extensible and standardized way of modeling besides the implementation of the architectural needs and the support of necessary services as basis for semantic M2M communication in BA.

7.3 Interface definition

Recent work in M2M communication and the IoT resulted in various technologies and standards with advantages and disadvantages. In this work, existing protocols are analyzed with respect to the addressed requirements (Section 7.3.1). An interface based on a tailored service set (Section 7.3.2) and a shared knowledge in the form of an ontology (Section 7.3.3) is introduced to handle operational M2M communication between devices, systems, and applications in the BA domain.

7.3.1 Communication protocol selection

IP has already found its way into common BA technologies. KNX, LonWorks, and BACnet provide an IP layer either for tunneling of control messages or as support of a native medium. As these solutions are limited to a specific technology, more general approaches that are depicted in Figure 7.3 are discussed in the following in order to find a suitable basis for the proposed semantic interface. The links outline default dependencies between the shown technologies.

IP-based technologies are commonly used for interoperable system integration due to the wide distribution and acceptance of the Internet Protocol. Also M2M communication efforts in BA for different types of systems from the field level up to the management level are usually built on IP. On the transport layer, the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) are the most common options as basis for application layer protocols.

The Hypertext Transfer Protocol (HTTP) following the REST paradigm is widely used in the Web. A small set of verbs is used for synchronous request-response interaction [21]. As a message exchange is initiated by clients, polling is required in order to get updates [22]. But the repetition of header information results in an overhead that is unwanted in many applications. The Simple Object Access Protocol (SOAP) is a WS-* approach using XML for message encoding. Thus, SOAP messages might need high bandwidth due to the

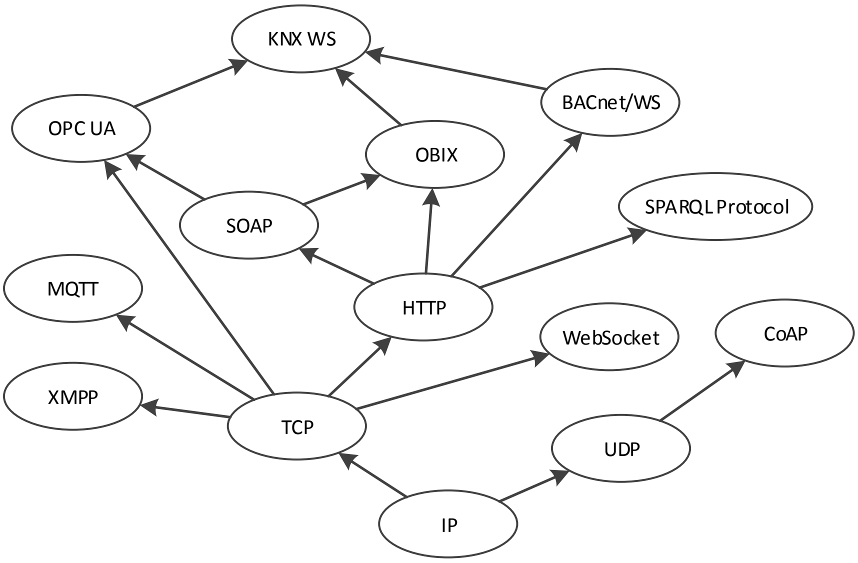

Figure 7.3: Relevant communication protocols and standards

large chunk of XML data [23]. As a result, SOAP is not suitable for M2M communication with constrained devices. Open Building Information Exchange (OBIX) [24], BACnet Web services (BACnet/WS) [25], and OPC Unified Architecture (OPC UA) [26] are standardized technologies on top of HTTP, SOAP, and TCP developed for BA-oriented but technology-independent M2M communication. The recently published KNX Web services (KNX WS) specification provides a Web interface to communicate with KNX networks based on these technologies [27]. Nevertheless, all four solutions define their own information models without using a standardized way to describe semantics (e.g., OWL), which limits integration with other domains and applications. Another relevant protocol based on HTTP is the SPARQL protocol [28]. This W3C recommendation uses the semantic language SPARQL to send queries and updates to a SPARQL service that is able to process these requests and return result sets. Moreover, there are the Message Queue Telemetry Transport (MQTT), the Extensible Messaging and Presence Protocol (XMPP), and the Constrained Application Protocol (CoAP) for M2M communication. CoAP targets resource-constrained devices and supports both request-response and publish-subscribe architectures [29]. As it runs on the unreliable UDP, it provides its own mechanism to ensure reliability. On the other hand, MQTT and XMPP run on TCP [21]. Compared to HTTP, MQTT has a lower overhead, which is beneficial for the use in constrained devices [29]. Clients can subscribe for a particular topic at a message broker to receive recent updates. Similar to CoAP, XMPP provides publish-subscribe and request-response architectures [21]. Compared to the relatively new MQTT, XMPP that was initially intended for chatting applications has better support in the Internet. Finally, the WebSocket protocol allows for bidirectional, full-duplex communication that runs on a single socket over the Web [22]. Similar to TCP, WebSocket requires another application protocol on top.

Regarding the previously introduced requirements, HTTP and SOAP drop out as they do not support publish-subscribe mechanisms and have potentially high message overhead. On the other hand, MQTT does not support request-response mechanisms, which are also relevant for a complete set of services. Although XMPP and CoAP combine both architectures, they do not provide services for semantic querying. However, this is important in order to read additional semantics that is not covered by the other services. KNX WS, OPC UA, OBIX, and BACnet/WS are not applicable as they do not use an ontology-based information modeling, which is necessary for platform-independent knowledge exchange. Therefore, the proposed solution is based on the WebSocket protocol, which meets all requirements. The bidirectional and asynchronous communication is a suitable basis for interactions in both directions. Different topologies ranging from meshed peer-to-peer connections to central message brokers are possible. Moreover, all required service categories as well as the semantics-based communication can be realized on top.

7.3.2 Service set description

On top of the WebSocket protocol, a SOA is used to define the necessary services. The message exchange between BA entities is based on a shared domain knowledge modeled by means of an ontology. Thus, semantic M2M communication is realized. Comparable to the message structure of HTTP, every message consists of an identifying message type, optional header fields, and a message content. Available header fields are the message ID (mandatory), the content type (mandatory), the sent date (optional), the expires date (optional), and the reference ID (optional/mandatory). The message content can be interpreted automatically by using the distributed semantics. An overview is given in Figure 7.4. The communication partners (e.g., BA devices) establish a (secure)

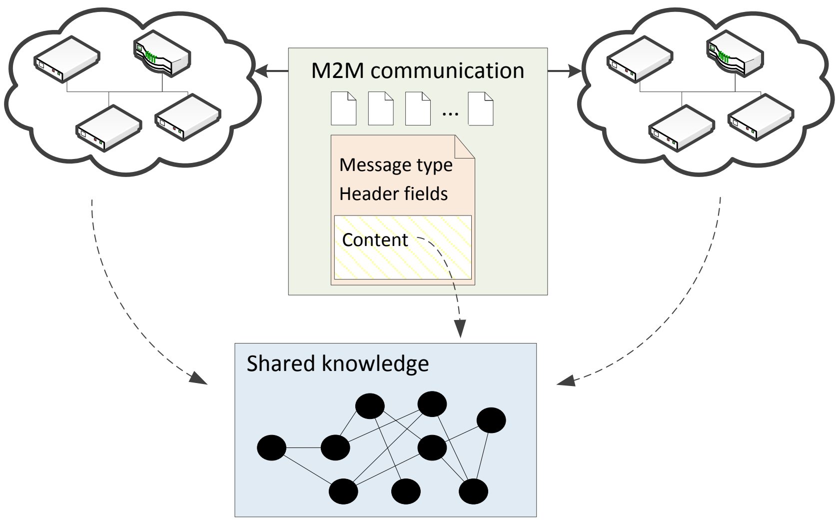

Figure 7.4: Overview of semantic M2M communication

WebSocket connection. Each partner has knowledge about the own context, which is supplemented by received information leading to a shared, distributed knowledge.

In total, the semantic interface specifies twelve services that correspond to the previously introduced six categories. Table 7.1 summarizes these services by listing their name, the identifying message type, a short description of the content, the allowed content types, and the types of response messages. The identification services (set 1) contain a register (*REG*) and a deregister (*DRE*) service. Based on those, a system can introduce itself using an RDF-based description. The communication partner tries to authenticate and authorize the registering system. If this initial registration is successful, other services can be executed. Deregistration uncouples the communicating systems. BA functions can be published and removed using *ADD* and *REM*, respectively (set 2). In case a system has already published a function, another system can subscribe for value updates with *OBS* while created observations can be removed using *DET* (set 3). Basic exchange of process data is realized with *PUT* which can be triggered by an observation. For example, a sensor device reads an updated value that is forwarded to an interested communication partner. A *PUT* is also sent in response to a *GET* request (set 4). Semantic querying services are *QUE* (read) and *UPD* (modify). Query results are returned with the *QRE* message type (set 5). Finally, status services (set 6) are realized by a single *STA* service that sends HTTP-like error codes and human-readable messages for debugging. For example, error code 200 states that processing was successful. In general, semantic descriptions are encoded in RDF/XML or Turtle while URIs use plain-text encoding. Queries are transmitted using SPARQL query and update content types. The query results are encoded in XML or JSON. Reference IDs need to be set if a message is sent in response to a previously received message (e.g., *QRE* message refers to a *QUE* message). This service set is an advancement of previous work that supports only the exchange of process data (*GET*, *PUT*) and semantic querying (*POST*) on the basis of HTTP [30].

7.3.3 Ontological concepts

A common understanding of the BA domain in the form of an ontology is a major element of the proposed semantic M2M communication as already depicted in Figure 7.4. Each communication partner has access to the ontological concepts (TBox) in order to instantiate BA devices or functions as well as other information within its own scope (ABox). The local knowledge bases are enhanced by sending and receiving semantically enriched information. Semantic Web standards like OWL and RDF ease the description of ontologies. In contrast to the representation of semantics in other M2M communication technologies, ontologies provide a platform-independent semantic modeling, which enables interlinking with other ontologies or sharing beyond system boundaries.

This work aims at using a high-level ontology that defines the most basic concepts but can be complemented by concepts of more specialized ontologies. As basis, the ontology for smart control in BA is reused [30]. Regarding the building information, a taxonomy of various zone types (e.g., room, office, site) is available. Zones can form nested structures and are arranged by means of zone delimiters. The orientation of buildings enables the

Table 7.1: Defined service set for semantic M2M communication

Set	Name	Message type	Content	Content type	Response message
1	Register system	REG	RDF-based description of registering system	application/rdf+xml, application/x-turtle	STA
	Deregister system	DRE	URI of system that will be deregistered	text/plain	STA
2	Add function	ADD	RDF description of published BA function	application/rdf+xml, application/x-turtle	STA
	Remove function	REM	URI of BA function that will be removed	text/plain	STA
3	Observe function	OBS	URI of published BA function that will be observed (optional parameter *freq* controls the periodicity of updates)	text/plain	STA
	Detach observation	DET	URI of already observed BA function	text/plain	STA
4	Put process data	PUT	RDF graph containing data values for an already added service	application/rdf+xml, application/x-turtle	STA (optional)
	Get process data	GET	URI of BA function, whose current value should be retrieved	text/plain	STA (optional), PUT
5	Query	QUE	Semantic query to read context information from ontology	application/sparql-query	STA (optional), QRE
	Query result	QRE	Results of the corresponding query	application/sparql-result+xml, application/sparql-result+json	STA (optional)
	Update	UPD	Update query to modify context information	application/sparql-update	STA
6	Status	STA	Status code and optional, human-readable description separated by a blank	text/plain	-

consideration of weather conditions. The location of BA devices and appliances is set in relation to this building structure. The modeled physical resources of the automation system are the link between the building and the BA functions. For sensing functionality, a parameter configuration specifies the semantics of the linked values. Moreover, these functions can be in dependency relations with each other. Actuating makes use of the control service concept. Here, control variations define the particular influence on environmental parameters. For conditional execution, modeling of constraints is supported. States describe the available range of set points while the history of past actions is archived in a linked sensing function. Energy needs for actuating are modeled explicitly with a formula or implicitly by measured consumption information. Relevant types of environmental parameters are also defined in the ontology. Units and value ranges characterize these parameters. In general, this ontology complies with the introduced requirements.

For further details, concepts of other ontologies can be integrated. BASont [16] or SAREF ontology [31] allow for a more precise device description. With respect to BIM, the ifcOWL ontology offers a fine-grained modeling of buildings [32]. DogOnt [33] and ThinkHome [15] are suitable for detailed representation of BASs. OWL gives the necessary language constructs to combine concepts of different ontologies and domains by introducing equivalence and subclass relations.

7.4 Feasibility evaluation

The evaluation of the proposed semantic interface is based on a proof-of-concept implementation[1]. In this work, a KNX installation consisting of several devices, such as an HVAC controller, a presence detection sensor, a multifunctional room sensor, or a switching actuator, acts as BAS. These devices, their functions, and the underlying building context are modeled as local BAS knowledge base by means of the ontology. A demo application represents a simple BMS to monitor the devices. Both the BAS and the demo BMS are embedded into the framework of the proof-of-concept implementation, where a central message broker, the semantic core, is responsible for managing connections to different systems. Thus, BMS and BAS are not directly connected in this setting but communicate via the semantic core. In such a star topology, the number of required connections will increase only linearly with the number of systems. Other BA technologies, smart grid communication, or weather services can be integrated into this framework, as well. As a result, the semantic core is aware of all exchanged information, i.e. it hosts the sum of all local knowledge bases. For this purpose, the open source triple store Apache Jena is utilized to manage the ontology. Moreover, the semantic core handles observations and is responsible for checking the identity of registering systems.

In this implementation, authentication and authorization are out of scope. Instead, the semantic core trusts the connecting systems by default in order to simplify the evaluation process. Queues for incoming and outgoing messages are used to parallelize the processing

[1] https://github.com/dschachinger/colibri

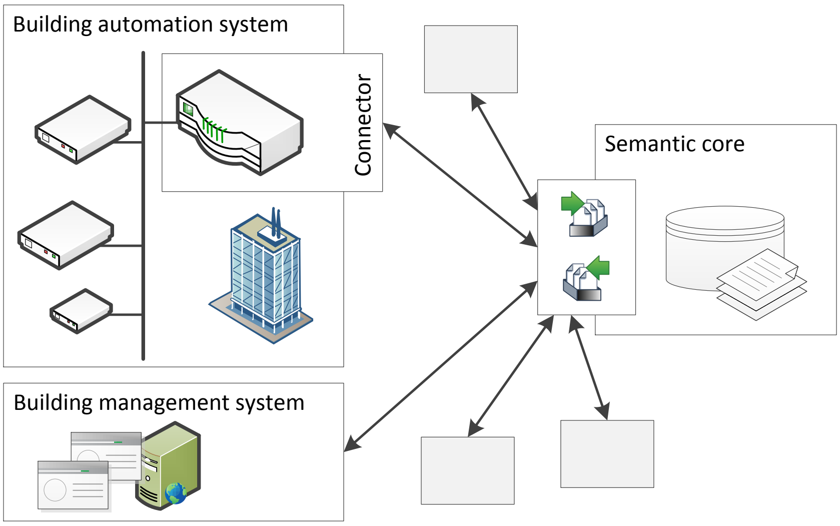

Figure 7.5: Implementation outline

tasks. While the demo BMS directly implements the proposed semantic interface, a distinct connector as gateway between the BAS communication and the semantic M2M communication is used. Figure 7.5 sketches this evaluation setting.

In order to show the feasibility in terms of functional capability, test cases are specified. The tests have predefined sequences of operations that need to be executed successfully. Optional prerequisites are used to define dependencies between tests. There is at least one test case for each of the specified interface services. Thus, error-free execution of all tests shows that the approach is generally feasible. It has to be noted that Turtle encoding for RDF-based message contents and JSON encoding for query results are not yet implemented. However, this only affects performance of message transmission and not the functional capability of the semantic M2M communication. It is assumed that the BAS connector, the demo BMS, and the semantic core are already running before the test procedures are executed, and WebSocket connections between these systems are successfully established. The connector for the KNX installation is implemented on a Raspberry Pi 2 Model B, which has 1GB RAM and an ARM Cortex-A7 CPU with 900MHz. The demo BMS and the semantic core are run on the same device, an Intel Core i7-2600 CPU at 3.4GHz with 8GB RAM.

Within the scope of this work, 16 basic test cases are defined so far. For representative purposes, one exemplary scenario as aggregation of multiple basic test cases is discussed in the following. A visualization of the sent and received messages between the BAS, the semantic core, and the BMS is presented in Figure 7.6. In order to keep the sequence diagram readable, status messages for acknowledgments are omitted. If a status message returns an error code indicating a problem, the test will be unsuccessful, anyway. Furthermore, only the relative order of exchanged messages is depicted. The test scenarios do not prescribe specific time slots for sending of messages. The aggregated test case

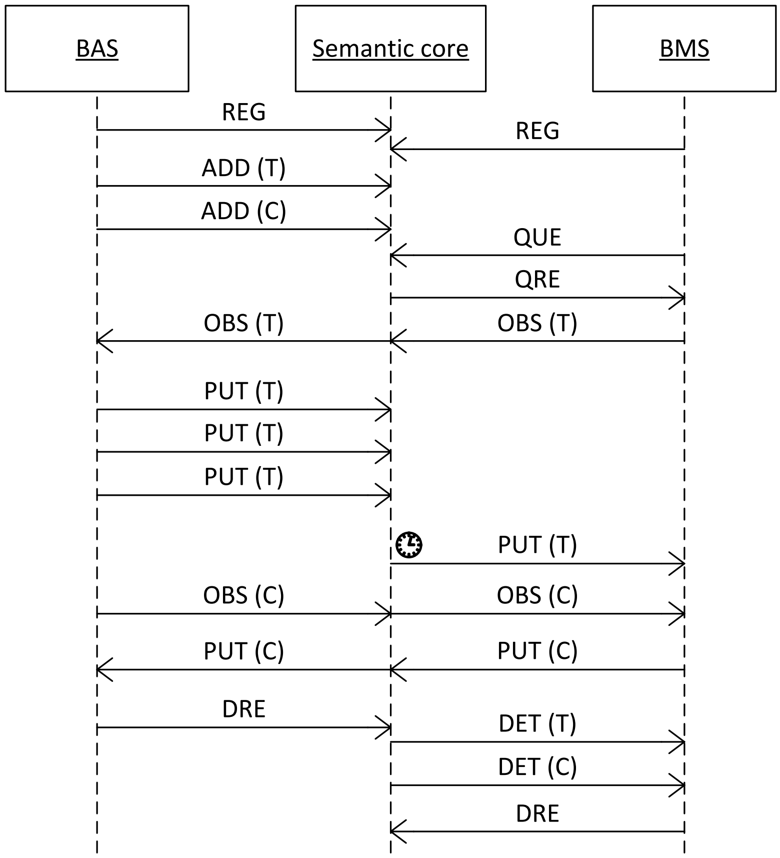

Figure 7.6: Exemplary test case

starts with registration (REG) of BAS and BMS at the semantic core. Both systems are informed about their successful identification. Then, the BAS publishes two function descriptions (ADD). First, a temperature sensing function is added to the core (T). All relevant characteristics for interpreting this function are enclosed in the description. In this example, the data configuration of the function specifies the progress of temperature values over time including unit and range. Second, an actuating function for cooling of a room is published (C). Afterwards, the demo BMS queries for all available sensing and actuating services that are known by the semantic core (QUE). The query that is partially shown in Listing 7.1 is built using the ontological concepts. The result set contains the previously added functions of the BAS (QRE).

As the BMS is interested in periodic value updates for all sensing functions, an observation for the temperature function is sent to the semantic core (OBS). Observations are forwarded to the original host of the function. Update messages with new values (PUT) are sent out not before the reception of an observation for a particular function. When the timer for the periodic observation of the BMS elapsed, the semantic core sends an update message combining all not yet forwarded PUT messages for the observed

Listing 7.1: QUE message to read sensing and actuating functions

```
QUE
Content-Type: application/sparql-query
Message-Id: 2017030123

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX colibri: <https://[...]/colibri.owl#>

SELECT (?f as ?function) ?c ?t ?u
WHERE {{ ?f rdf:type colibri:DataService.
  BIND ('sensing' AS ?c).
  OPTIONAL { ?f colibri:monitorsParameter ?p.
     ?p rdf:type ?t.
     ?t rdfs:subClassOf colibri:EnvironmentalParameter.
     OPTIONAL {?p colibri:hasUnit ?u. }}}
UNION { ?f rdf:type colibri:ControlService.
  BIND ('actuating' AS ?c)
  OPTIONAL { ?f colibri:controlsParameter ?p. [...] }}}}
ORDER BY ?function
```

function. Then, the BAS is interested in new set points for the cooling function. Thus, a non-periodic observation is sent to the core, which is forwarded to the BMS. The demo BMS is not intended to implement a low-level controller, but for testing purposes, a modified cooling set point is pushed to the core. This value is subsequently received by the BAS and written to the KNX network. When the BAS closes the connection (*DRE*), all still open observations are closed by the semantic core (*DET*). Finally, the BMS closes its connection, as well.

In addition to the functional capability, hardware requirements are considered. The semantic core consumes about 25MB RAM after garbage collection in the presented setting. The triple store does not hold the entire ontology in the main memory. Thus, also larger ontologies with about 21,000 triples need roughly the same memory. Depending on the executed operations, there are also peaks in memory usage. However, these peaks are only temporary, and constrained devices, such as the used Raspberry Pi 2 with 1GB RAM, are also able to host an implementation of the proposed M2M communication interface with a local triple store. As state-of-the-art Internet infrastructure together with manageable message sizes is used, measured message transmission times are negligible for non-critical BA applications. With respect to message processing, reasoners can be used to infer new knowledge based on received message contents. In this case, message sizes can be reduced by omitting some explicit information. However, measurements during the feasibility study indicate that the execution of reasoners is a performance bottleneck. It needs to be analyzed in future work if this is due to the proof-of-concept implementation or the utilized reasoner. On the other hand, direct processing of message contents without the preceding execution of a reasoner is within the range of acceptable response times in the BA domain.

In summary, all test cases were executed successfully. The aggregated test should give an insight into the evaluation work that was done. All sporadic problems that occurred were caused by the proof-of-concept implementation and could be fixed. Memory footprint is low enabling deployment on resource-constrained devices. Moreover, the advantage of automatic message interpretation by means of an ontology in this communication interface is highlighted. An additional, more detailed evaluation regarding performance, such as throughput or response times, needs to be done in order to determine if the approach can also be used in more time-critical applications. Nevertheless, the basic feasibility of the proposed semantic interface for M2M communication in BA could be shown.

7.5 Conclusion

Efforts in system integration are continuously proceeding with respect to the IoT. As a result, interoperability at an abstract, semantic level gets more and more important. This work presents a semantic interface for M2M communication in the BA domain that enables automatic interpretation of messages based on an ontology. The interface is specified on top of common Web technologies and uses WebSocket connections for bidirectional message exchange. The set of defined services covers identification of communication partners, publication of available BA functions, subscription to value updates of BA functions, exchange of semantically enriched process data, status notifications, and semantic querying. The underlying ontology is based on previous work and considers modeling of building structures and automation systems as well as basic sensing and actuating functionality. Approaches of existing M2M technologies are combined with the advantages of semantic modeling based on Semantic Web standards. Feasibility and hardware requirements are evaluated based on several basic and aggregated test cases. The results indicate applicability of the proposed approach.

Future work includes a more detailed evaluation of the semantic interface as well as field tests with different devices and systems. For this purpose, the proof-of-concept implementation needs to be further improved. The focus is on performance of message transmission over WebSocket with differing content encodings compared to other M2M communication technologies. Moreover, loads on the communication channel and end-to-end latencies are measured in order to determine limitations for applicability of the semantic interface in the BA domain. Measurements on hardware requirements are performed in the field tests to support selection of appropriate platforms in practice.

Acknowledgment

This work was funded by FFG (Austrian Research Promotion Agency) under the *BProfEM* project (845627) as well as by a netidee scholarship and the Lions Club Wien St. Stephan.

References

[1] T. Sauter, S. Soucek, W. Kastner, and D. Dietrich, “The Evolution of Factory and Building Automation”, *IEEE Industrial Electronics Magazine*, vol. 5, no. 3, pp. 35–48, Sep. 2011, ISSN: 1932-4529. DOI: 10.1109/MIE.2011.942175.

[2] D. Dietrich, D. Bruckner, G. Zucker, and P. Palensky, “Communication and Computation in Buildings: A Short Introduction and Overview”, *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3577–3584, Nov. 2010, ISSN: 0278-0046. DOI: 10.1109/TIE.2010.2046570.

[3] International Energy Agency, *Transition to Sustainable Buildings: Strategies and Opportunities to 2050.* OECD/IEA, 2013, ISBN: 9789264202412.

[4] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions”, *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, Sep. 2013, ISSN: 0167-739X. DOI: 10.1016/j.future.2013.01.010.

[5] T. Bourgeau, H. Chaouchi, and P. Kirci, “Machine-to-Machine Communications”, in *Next-Generation Wireless Technologies: 4G and Beyond*, N. Chilamkurti, S. Zeadally, and H. Chaouchi, Eds., Springer London, 2013, ch. 11, pp. 221–241, ISBN: 9781447151630. DOI: 10.1007/978-1-4471-5164-7_11.

[6] D. Niyato, L. Xiao, and P. Wang, “Machine-to-Machine Communications for Home Energy Management System in Smart Grid”, *IEEE Communications Magazine*, vol. 49, no. 4, pp. 53–59, Apr. 2011, ISSN: 0163-6804. DOI: 10.1109/MCOM.2011.5741146.

[7] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey”, *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010, ISSN: 1389-1286. DOI: 10.1016/j.comnet.2010.05.010.

[8] M. Jung, J. Weidinger, W. Kastner, and A. Olivieri, “Building automation and smart cities: An integration approach based on a service-oriented architecture”, in *IEEE International Conference on Advanced Information Networking and Applications*, Mar. 2013, pp. 1361–1367. DOI: 10.1109/WAINA.2013.200.

[9] F. Jammes and H. Smit, “Service-Oriented Paradigms in Industrial Automation”, *IEEE Transactions on Industrial Informatics*, vol. 1, no. 1, pp. 62–70, Feb. 2005, ISSN: 1551-3203. DOI: 10.1109/TII.2005.844419.

[10] W. Kastner, M. Kofler, M. Jung, G. Gridling, and J. Weidinger, “Building Automation Systems Integration into the Internet of Things: The IoT6 approach, its realization and validation”, in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2014, pp. 1–9. DOI: 10.1109/ETFA.2014.7005197.

[11] J. Bai, Y. Hao, and G. Miao, “Integrating Building Automation Systems based on Web Services”, *Journal of Software*, vol. 6, no. 11, pp. 2209–2216, Nov. 2011, ISSN: 1796-217X. DOI: 10.4304/jsw.6.11.2209-2216.

[12] B. Ramis, L. Gonzalez, S. Iarovyi, A. Lobov, J. L. M. Lastra, V. Vyatkin, and W. Dai, "Knowledge-based web service integration for industrial automation", in *IEEE International Conference on Industrial Informatics*, Jul. 2014, pp. 733–739. DOI: 10.1109/INDIN.2014.6945604.

[13] D. Guinard, I. Ion, and S. Mayer, "In Search of an Internet of Things Service Architecture: REST or WS-*? A Developers' Perspective", in *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, A. Puiatti and T. Gu, Eds., vol. 104, Springer Berlin Heidelberg, 2012, pp. 326–337, ISBN: 9783642309724. DOI: 10.1007/978-3-642-30973-1_32.

[14] C. Martinez-Cruz, I. J. Blanco, and M. A. Vila, "Ontologies versus relational databases: Are they so different? A comparison", *Artificial Intelligence Review*, vol. 38, no. 4, pp. 271–290, Dec. 2012, ISSN: 0269-2821. DOI: 10.1007/s10462-011-9251-9.

[15] M. J. Kofler, C. Reinisch, and W. Kastner, "A semantic representation of energy-related information in future smart homes", *Energy and Buildings*, vol. 47, pp. 169–179, Apr. 2012, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2011.11.044.

[16] J. Ploennigs, B. Hensel, H. Dibowski, and K. Kabitzsch, "BASont - A modular, adaptive Building Automation System Ontology", in *Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2012, pp. 4827–4833. DOI: 10.1109/IECON.2012.6389583.

[17] E. Corry, P. Pauwels, S. Hu, M. Keane, and J. O'Donnell, "A performance assessment ontology for the environmental and energy management of buildings", *Automation in Construction*, vol. 57, pp. 249–259, Sep. 2015, ISSN: 0926-5805. DOI: 10.1016/j.autcon.2015.05.002.

[18] A. Gyrard, C. Bonnet, and K. Boudaoud, "Enrich Machine-to-Machine Data with Semantic Web Technologies for Cross-Domain Applications", in *IEEE World Forum on Internet of Things*, Mar. 2014, pp. 559–564. DOI: 10.1109/WF-IoT.2014.6803229.

[19] M. Chen and B. Shen, "A Semantic Unification Approach for M2M Applications based on Ontology", in *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, Oct. 2011, pp. 265–271. DOI: 10.1109/WiMOB.2011.6085389.

[20] ETSI, *Machine-to-Machine communications (M2M): M2M service requirements (ETSI TS 102 689 V2.1.1)*, Jul. 2013.

[21] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A Survey on Application Layer Protocols for the Internet of Things", *Transaction on IoT and Cloud Computing*, vol. 3, no. 1, pp. 11–17, Jan. 2015.

[22] V. Pimentel and B. G. Nickerson, "Communicating and Displaying Real-Time Data with WebSocket", *IEEE Internet Computing*, vol. 16, no. 4, pp. 45–53, Jul.-Aug. 2012, ISSN: 1089-7801. DOI: 10.1109/MIC.2012.64.

[23] B. Upadhyaya, Y. Zou, H. Xiao, J. Ng, and A. Lau, “Migration of SOAP-based services to RESTful services”, in *IEEE International Symposium on Web Systems Evolution*, Sep. 2011, pp. 105–114. DOI: 10.1109/WSE.2011.6081828.

[24] OASIS, *OBIX Version 1.1, Committee Specification Draft 02 / Public Review Draft 02*, Dec. 2013.

[25] ASHRAE, *Data Communication Protocol for Building Automation and Control Networks (ANSI/ASHRAE Addendum am to ANSI/ASHRAE Standard 135-2012)*, Apr. 2016.

[26] OPC Foundation, *OPC Unified Architecture Specification, Part 1: Overview and Concepts, Release 1.03*, Oct. 2015.

[27] D. Schachinger and W. Kastner, “Integration von KNX Netzwerken in das Internet der Dinge: Die KNX Web Services Spezifikation”, in *Jahreskolloquium Kommunikation in der Automation*, Nov. 2016, pp. 1–8.

[28] W3C SPARQL Working Group, *SPARQL 1.1 Protocol, W3C Recommendation*, Mar. 2013.

[29] D. Thangavel, X. Ma, A. Valera, H.-X. Tan, and C. K.-Y. Tan, “Performance Evaluation of MQTT and CoAP via a Common Middleware”, in *IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Apr. 2014, pp. 1–6. DOI: 10.1109/ISSNIP.2014.6827678.

[30] D. Schachinger and W. Kastner, “Semantics for smart control of building automation”, in *IEEE International Symposium on Industrial Electronics*, Jun. 2016, pp. 1073–1078. DOI: 10.1109/ISIE.2016.7745041.

[31] L. Daniele, F. den Hartog, and J. Roes, “Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology”, in *Formal Ontologies Meet Industry*, ser. Lecture Notes in Business Information Processing, R. Cuel and R. Young, Eds., vol. 225, Springer International Publishing, 2015, pp. 100–112, ISBN: 9783319215440. DOI: 10.1007/978-3-319-21545-7_9.

[32] P. Pauwels and W. Terkaj, “EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology”, *Automation in Construction*, vol. 63, pp. 100–133, Mar. 2016, ISSN: 0926-5805. DOI: 10.1016/j.autcon.2015.12.003.

[33] D. Bonino and F. Corno, “DogOnt - Ontology Modeling for Intelligent Domotic Environments”, in *The Semantic Web - ISWC 2008*, ser. Lecture Notes in Computer Science, A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, Eds., vol. 5318, Springer Berlin Heidelberg, 2008, pp. 790–803, ISBN: 9783540885634. DOI: 10.1007/978-3-540-88564-1_51.

CHAPTER 8

Ontology-based generation of optimization problems for building energy management

Abstract: In general, a trade-off between comfort satisfaction and minimization of energy consumption or overall costs needs to be found by building energy management systems. Additionally, the design of energy management strategies often requires high effort and expert knowledge in order to model the dynamics within a building, which leads to very specific solutions with limited reuse. Thus, this work presents an approach for the automatic generation of optimization problems for building energy management based on machine-readable semantics. For this purpose, an ontology hosts all relevant information necessary for the optimization problem formulation. Information extraction and transformation into the optimization problem domain are addressed. Moreover, a case study demonstrates the functionality of the proposed procedure.

Keywords: Building automation, energy management, energy efficiency, optimization, ontology, semantics

The content of this chapter has been published: D. Schachinger and W. Kastner, "Ontology-based generation of optimization problems for building energy management", in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2017, pp. 1–8.

8.1 Introduction

The International Energy Agency (IEA) reported that commercial and residential buildings account for approximately 35% of global energy demand in 2010 with a continuous gain [1]. As a consequence, measures to boost energy efficiency regarding construction and operation of buildings are required in order to reach the global sustainability goals. After commissioning, building energy management systems (BEMSs) can be utilized during operation for reducing energy costs and consumption while trying to satisfy comfort requirements. Building automation systems (BASs) provide these BEMSs with data of the buildings and their environment. Thus, BASs form a major contribution to energy management in buildings while ensuring safety and security [2]. Their high potential in energy savings is due to the wide range of supported sensors and actuators to measure and control different domains in a building [3]. This situation is enforced by recent integration efforts of BASs into the Internet of Things (IoT) [4]. Based on service-oriented architectures (SOAs), heterogeneous technologies and standards in the building automation (BA) domain will become more interoperable, which eases their utilization in management applications [3]. In addition, intensified focus on smart grids and demand side management (DSM) including measures for incentive-based or time-based demand response (DR) programs offers new opportunities for energy management in buildings [5].

In general, BEMSs are faced with the conflicting goals of minimizing comfort dissatisfaction and minimizing energy consumption or costs. Surveys point out that manifold methods emerged in order to tackle different issues in energy management of residential or commercial buildings [6, 7]. Most of the approaches are focused on heating, ventilation, and air conditioning (HVAC) tasks as they are very energy-intensive. For example, Merabti et al. discuss the utilization of fuzzy logic and genetic algorithms (GAs) to improve conventional HVAC controllers [8]. Thermal comfort in home energy management based on mixed-integer linear programming (MILP) is discussed in [9]. Another MILP approach introduces the abstract concept of services to distinguish between supportive, intermediate, end-user, temporary, and permanent tasks [10]. Arikiez et al. propose a heuristic-based approach for cost minimization of air conditioning operation as they argue that exact algorithms have limited applicability in complex BEMSs [11]. In [12], particle swarm optimization (PSO) is used in a multi-agent system to find a trade-off between energy consumption and comfort regarding temperature, illumination, and air quality. Using dynamic programming (DP), Sun et al. combine HVAC, lighting, and shading in their approach [13]. A BEMS design tackling DSM and DR by means of a universal energy meter is presented in [14]. Moreover, a distributed framework for cost-minimizing energy management for smart grid users is shown in [15].

Although this related work offers good solutions for energy management, the adoption to a particular building requires manual modeling and engineering effort in order to adjust the general approaches to the actual needs. Expert knowledge is required to define the objective function and the constraints of the optimization problem. However, this knowledge is usually not available in machine-readable form in order to support automatic processing. In this context, ontologies that are known from the Semantic Web represent a suitable solution to model the required information and semantics. Regarding building information modeling (BIM), a transformation of the Industry Foundation Classes (IFC) standard to the ifcOWL ontology exists [16]. The DogOnt ontology can be used to model building resources and building environments as well as states and functionality of controllable devices [17]. Built on these concepts, the ThinkHome ontology also incorporates energy-related information [18]. The SAREF ontology for smart appliances is focused on devices and their sensing, actuating, and metering functions within a building taking into account energy profiles and prices [19]. A more detailed description of device functionality is provided in [20]. Based on this device description ontology, the BASont ontology addresses several use cases in the BA life cycle [21]. Utilizing ifcOWL and the Semantic Sensor Network (SSN) ontology, Corry et al. introduce an ontology for performance assessment in energy management of buildings [22]. An ontology for representing smart grid communication is proposed in [23].

Accordingly, this work presents an automatic process for the generation of optimization problems from an ontology that hosts all relevant information. This eases the design of BEMSs by reducing the manual effort to specify objective functions or constraints. The knowledge of different domain experts is combined in an abstract and platform-independent ontology, which can be reused and shared. In order to evaluate the objective function (fitness function), elements in the ontology, like building zones, comfort parameters, BA devices, energy suppliers, or capacities, are mapped to variables and constants as well as constraints that limit the solution space. The resulting optimization problem provides a basis for execution of online optimization algorithms or simulation-based methods on top. For example, heuristics can be applied to determine a schedule for BAS operation over an optimization period.

In Section 8.2, the scope of this work is discussed. Then, the process of extracting relevant information of the ontology as input to the problem formulation is addressed in Section 8.3. The formulation of the optimization problem is presented in Section 8.4. An exemplary case study shows the functionality of the approach in Section 8.5. Finally, the work is discussed in Section 8.6 before Section 8.7 concludes the paper.

8.2 Optimization in building energy management

Generally, optimization tries to find inputs that minimize or maximize a function considering optional constraints [6]. Regarding optimization in building energy management, there are usually multiple conflicting objectives. In such a situation, a set of trade-off solutions, called Pareto optimal solutions, can be identified that describe the best compromises between these objectives [24]. This work targets two traditional objectives that are visualized in Figure 8.1 and summarized in the following.

- First, the comfort needs of building users have to be satisfied. Thus, the objective value of comfort dissatisfaction is minimized. In this context, different comfort parameters are considered including temperature, air humidity, air quality (e.g., CO_2 level), brightness, and noise level.

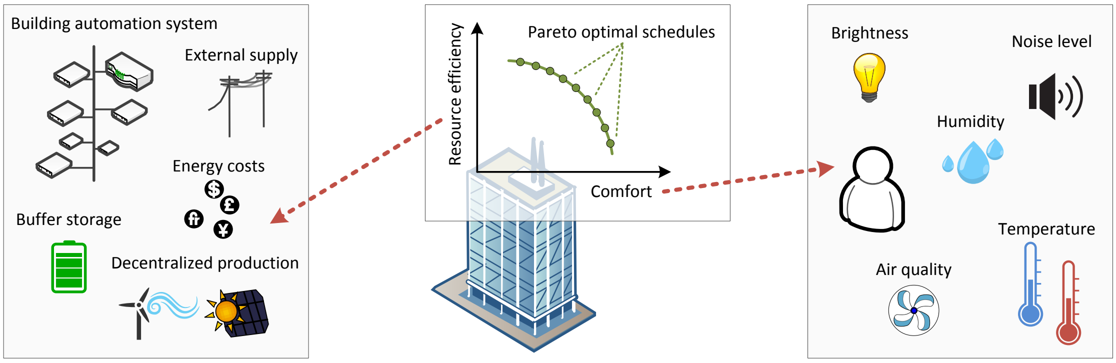

Figure 8.1: Conflicting objectives and corresponding influences in building energy management

- Second, the efficiency of resource utilization needs to be maintained. In building energy management, this main objective has different manifestations. While minimization of energy consumption tries to reduce the overall amount of demanded energy, cost minimization additionally takes into account varying costs for energy production and procurement. A balance between the energy demand of BAS devices and the energy supply of decentralized production plants, buffer storage, or the external grid is searched. In this work, the focus is on cost minimization with inherent reduction of consumption.

These objectives are combined in a single function that is used to evaluate the fitness of the identified solutions, which represent executable schedules determining state changes of energy consuming and providing devices of the building over a period of n time slots. By incremental modification of a schedule, the fitness value might be improved. Solutions that cannot be dominated any more are already on the Pareto front. However, it might be infeasible to find such an optimal solution depending on the size of the optimization problem and the search space. For such hard problems, heuristics can be utilized to find suitable solutions in reasonable time.

$$min \sum_{t=1}^{n} F_t \quad \text{where} \quad F_t = \omega \cdot c_t + (1 - \omega) \cdot e_t \tag{8.1}$$

In Equation 8.1, the minimization of the fitness function F_t is sketched. Comfort dissatisfaction and resource efficiency are calculated at discrete intervals from $t = 1..n$. The function e_t quantifies the resource consumption in time slot t while the function c_t is used for the quantification of discomfort. The sum of all intervals gives the fitness value of the entire schedule. In order to describe the differing priorities of these two objectives in a single function, the weight ω is introduced.

In summary, the scope of the optimization problem generation in this work is on (1) minimizing comfort dissatisfaction of building users regarding the mentioned parameters and (2) minimizing costs of energy consumption. Besides monitoring data of the building, day-ahead price information from the smart grid and reliable weather forecasts are required to exploit the full potential of load shifting, decentralized production, and fluctuating price levels in combination with overall reduction of consumption in the building context. Flexibility trading is currently out of scope and subject to future work. Moreover, adjustment of weights in order to find the optimal balance between the objectives is not considered in this work.

8.3 Ontology-based information extraction

According to this general goal, the extraction of information in order to form an optimization problem by means of decision variables, constants, objective function, and constraints is defined. As already mentioned, semantics about the BAS, the building, and the environment is modeled in an ontology. Descriptions of controllable devices that can influence comfort parameters of the building are needed. Their functional capabilities are the basis for a final control schedule. With respect to energy efficiency, information about the energy demand of these devices is necessary. Additionally, details about the energy suppliers (e.g., renewable energy resources, such as a local PV plant) have to be integrated into the ontology. The result is a preferably complete representation of the situation in and around the building. It has to be noted that a complete ontology covering all possible use cases cannot be ensured.

As basis for this work, the abstract ontology for smart control in BA is reused that aims at defining the most basic concepts, which can be complemented by concepts of more specialized ontologies [25]. A taxonomy of various zone types (e.g., room, office) is available to describe building information. Zones can form nested structures and arbitrary arrangements. Moreover, they are surrounded by zone delimiters. The orientation of a building enables the consideration of environmental conditions. BA devices and appliances are located in relation to this building structure and provide different services. Sensing functionality is described as data service that contains a parameter configuration specifying the semantics of related values. For this purpose, relevant types of parameters are defined in the ontology. A unit system and process ranges characterize these parameters. Data services are used for collecting both comfort and metering data. Similarly, price information or priority signals from the smart grid are modeled using this concept. On the other hand, actuating functionality makes use of the control service concept by defining control variations that influence comfort parameters. States define the available range of set points for these control services. Energy demand for actuating is modeled explicitly with a formula or implicitly by measured consumption data. In order to model local energy production plants and external supply from the grid, the concept of an energy service is added to the ontology. For conditional execution of services or capacities of energy supply, the definition of constraints is supported. Energy consumers and providers are matched by a set of energy types that are specified in the ontology. For modeling the priorities of comfort parameters over time, the ontology is

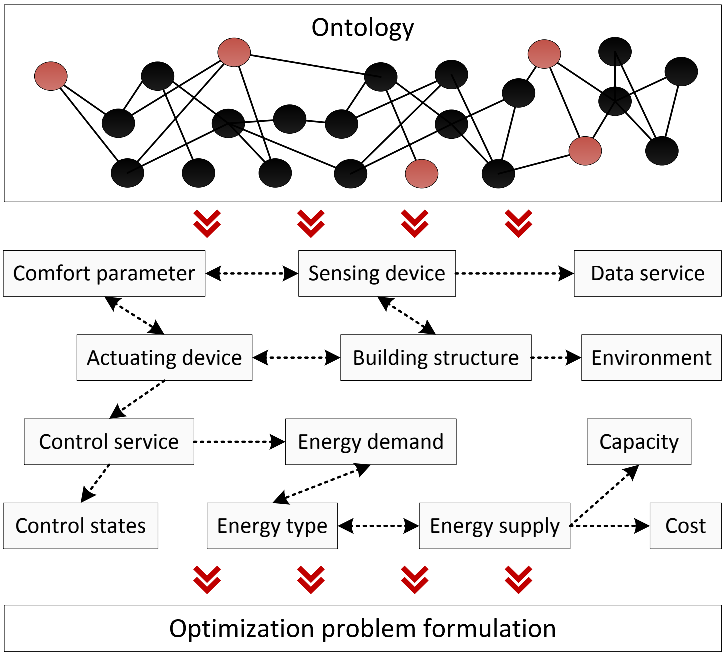

Figure 8.2: Information extraction procedure

extended by a new object property to link data services. The final OWL file is part of the Colibri project[1].

The extraction process aims at preparing the information for the subsequent optimization problem formulation. This procedure is illustrated in Figure 8.2. The arrows visualize the available navigation directions. For example, the actuating devices per zone can be retrieved based on the already requested building structure. On the other hand, the zones that are influenced by an actuating device can be extracted leading to the same result. In general, extraction means to request information from the OWL-based ontology by sending SPARQL queries.

Starting from the ontology as common knowledge base that contains the modeled expert knowledge for the BEMS design, there are multiple eligible paths through the linked information. Although different entry points for information extraction can be used, this work describes the procedure beginning with the set of defined comfort parameters as this turns out to be a reasonable approach. With the comfort parameters, their priority values are read. Next, the sensing and actuating devices per parameter are queried. Taking both device sets into account, the monitored and controlled zones are fetched, respectively. Based on the zones and their orientation, information about the environment is extracted. Moreover, the data forecasts of sensing devices that give the desired comfort values for the optimization period are read from the data services. Although this prediction is stored in the ontology, it is calculated out-of-band based on historic values. An actuating

[1] https://github.com/dschachinger/colibri

device is characterized by control services that are retrieved in the next phase. The device functionality is identified by control states that influence comfort parameters in a certain way. These discrete, continuous, relative, or absolute states represent the range of accepted values for the corresponding decision variables in the optimization problem. Control services have some energy demand of a certain type. It is assumed that there is only one local network of energy consumers and providers per energy type. Finally, the energy suppliers per energy type are extracted from the ontology. This includes the information about costs and constraints for their production or supply capabilities. As an assumption, each supplier provides only one type of energy. Although the sensing and actuating devices consume energy, the focus of this work is on the energy demand that is needed to change comfort parameters via control services.

As can be seen, the extraction schema starts with reading comfort-specific information before the energy-related domain is searched. The predefined extraction path specifies which information should be read. Based on the semantics of the underlying ontological concepts, the SPARQL queries can be built dynamically to request this information.

8.4 Automatic problem formulation

In order to realize the automatic optimization problem formulation, the extracted information needs to be mapped to control variables, decision variables, and constants. In this context, the term constant implies that such values are read during generation of the optimization problem and cannot be modified in contrast to decision variables. In addition, device-specific and building-specific constraints are generated. The prerequisites for the transformation process are the generic objective function (Section 8.4.1), rules for the mapping of variables and constants (Section 8.4.2), and a set of default constraints (Section 8.4.3). A summary of all symbols used in the following is given in Table 8.1.

8.4.1 Objective function

The basic objective function shown in Equation 8.1 is extended by detailed specifications of the time-dependent functions c_t and e_t for comfort and cost efficiency, respectively. Both functions are computed for all time intervals $t = 1..n$. Equation 8.2 gives the generic structure for evaluating comfort dissatisfaction. For each comfort parameter p and each building zone z, the square deviation from the desired value r_{tpz} is calculated. The constant m_{pz} determines if there is some measurement and control of parameter p in zone z. The priority λ_{tp} is used to weight the comfort parameters. The prediction function v_{tpz} estimates the value of parameter p in zone z and time slot t. This function takes into account immutable influences $\boldsymbol{i}_t$ (e.g., outdoor temperature) and states of controllable loads $\boldsymbol{l}_t$ (e.g., HVAC device) that have an impact on the considered parameter and zone. Depending on the implementation, this function can be based on different methods. For example, a recurrent neural network can be used to predict the comfort value with respect to a relevant subset of $\boldsymbol{i}_t$ and $\boldsymbol{l}_t$. When a zone is unoccupied given by the binary constant o_{tz}, the comfort dissatisfaction is neglected.

Table 8.1: Variables, constants, and functions

p	Comfort parameter	Control variable
z	Building zone	Control variable
g	Energy type	Control variable
y	Energy supply unit	Control variable
t	Time slot	Control variable
l	State of controllable load	Decision variable
s	State of energy supply unit	Decision variable
q	Coverage ratio of energy demand	Decision variable
b	Energy storage level	Auxiliary variable
m	Monitoring and control indicator	Binary constant
j	Energy supply indicator	Binary constant
o	Occupancy	Binary constant
r	Desired comfort value	Continuous constant
f	Energy price	Continuous constant
i	Immutable influence	Continuous constant
λ	Priority value	Continuous constant
a	Charging capacity	Continuous constant
u	Discharging capacity	Continuous constant
w	Energy loss	Continuous constant
d	Expected energy demand	Prediction function
v	Excepted comfort value	Prediction function

Similarly, Equation 8.3 determines the costs for energy allocation. The energy demand per energy type g is calculated by means of the prediction function d_{tg}. Again, this function uses device states $\boldsymbol{l}_t$ and immutable data $\boldsymbol{i}_t$. The full vectors are set as input, but the function uses only a relevant subset of the values depending on the energy type g. The resulting demand is apportioned among the energy supply units y covering local production plants, energy storage, and external supply. The constant j_{gy} indicates if supply unit y is able to provide energy of type g. The ratio per device q_{ty} is multiplied with the price per unit f_{ty} and the binary state of the supply unit s_{ty}.

$$c_t = \sum_p \sum_z m_{pz} \cdot \lambda_{tp} \cdot o_{tz} \cdot (v_{tpz}(\boldsymbol{l}_t, \boldsymbol{i}_t) - r_{tpz})^2 \tag{8.2}$$

$$e_t = \sum_g \sum_y j_{gy} \cdot q_{ty} \cdot f_{ty} \cdot s_{ty} \cdot d_{tg}(\boldsymbol{l}_t, \boldsymbol{i}_t) \tag{8.3}$$

8.4.2 Variable and constant mapping

An important task is the mapping of decision variables, control variables, and constants of the optimization problem. Lists are assigned to the control variables that contain the corresponding elements of the ontology. For example, the supported energy types g are fed into a 1-dimensional collection. The list of building zones z is ordered bottom-up in accordance with the zone hierarchy. Regarding the list of energy supply units y, the storage systems are added first before the mere suppliers are mapped. This is also done in all other data structures that host energy storage systems. The optimization period is split into n time slots indexed by t. For the auxiliary variable b, a 2-dimensional array is initialized to hold energy storage levels per storage unit y and time slot t.

The constants for monitoring and control indicators, energy supply indicators, occupancy, energy prices, and priority values are initialized as 2-dimensional data structures. A parameter is monitored or controlled in a zone if there are corresponding sensing and actuating devices. Thus, parameter p and zone z form the dimensions of the data structure for m. If an energy supply unit y provides energy of type g, the corresponding binary constant for j is set. The binary occupancy value o per zone z and time slot t is predicted based on historic data. Energy prices f are received from an energy retailer and stored in the ontology. The priority values λ are ordered by time slot t and comfort parameter p. A 3-dimensional data structure is needed for the desired comfort values r per time slot t, parameter p , and zone z. Charging and discharging capacities as well as energy loss of storage systems in terms of energy per time slot are stored in a list indexed by a, u, and w, respectively. Based on a 2-dimensional data structure, objects are created per influence i and time slot t containing relations to building zones and comfort parameters besides the actual value. It has to be noted that conversions of values might be necessary according to the associated units.

Finally, the decision variables are mapped that represent a control schedule. The coverage ratios of energy demand q accept values in the interval $[0, 1]$ per time slot t and supply unit y. Another 2-dimensional array is needed for representing the states s of energy supply units y per time slot t in the form of binary decision variables. Similarly, the data structure for the states of controllable loads l per time slot t is created. Here, the charging states of storage devices are mapped to binary decision variables while the states for BA devices that directly influence comfort parameters are mapped from the ontology. In order to define the scope of action for identifying appropriate schedules, an additional object is created for each control service, which describes the possible parameter variations per building zone via the available states.

8.4.3 Constraints

With respect to the energy efficiency objective, a set of default constraints has to be instantiated in the problem formulation process. In order to give an overview, some of them are explained in the following. First, flow conservation needs to be ensured for all storage devices. The energy level b_{ty} at the end of time slot t is the sum of the level at the end of slot $t-1$ and the changes in time slot t, i.e., energy inflow a_y, outflow by discharging u_y, and loss of energy w_y (Equation 8.4). Storage is charged if the device consumes energy (l_{ty}). On the other hand, discharging represents energy supply (s_{ty}). A storage cannot be charged and discharged at the same time (Equation 8.5), and the storage level always needs to be positive (Equation 8.6). The sum over the demand ratios q_{ty} must be 1 for all energy types g (Equation 8.7). Furthermore, a constraint for the assumption that each energy supply unit can only provide one type of energy is needed (Equation 8.8).

These default constraints are instantiated during the mapping process. All other device or building-specific constraints result from the information modeled in the ontology. For example, the maximum storage level of a battery that must not be exceeded at any time is extracted. Also the energy production capability of a PV plant has to correspond with the chosen demand ratios in the optimization period. Thresholds for CO_2 become constraints, as well. Furthermore, constraints that specify the accepted values for decision variables are generated. Some of these constraints need continuous evaluation in each time slot while others are checked only once. In summary, the optimization problem as input for an online algorithm or a simulation-based evaluation is automatically formulated using the extracted information of the ontology.

$$b_{(t-1)y} + l_{ty} \cdot a_y - s_{ty} \cdot u_y - w_y = b_{ty} \quad \forall t, y \tag{8.4}$$

$$l_{ty} + s_{ty} \leq 1 \quad \forall t, y \tag{8.5}$$

$$b_{ty} \geq 0 \quad \forall t, y \tag{8.6}$$

$$\sum_y j_{gy} \cdot q_{ty} \cdot s_{ty} = 1 \quad \forall t, g \tag{8.7}$$

$$\sum_g j_{gy} = 1 \quad \forall y \tag{8.8}$$

8.5 Case study

The functionality of the introduced approach is shown with an exemplary case study. For this purpose, two rooms of an office building are modeled in the ontology. The room structure and some BAS components are visualized in Figure 8.3.

Office 3.12 has two BA services including heating (A) and artificial lighting (B). On the other hand, Office 3.11 has artificial lighting (C, F, I), heating (D, H), and a device with combined cooling and ventilation (G). Moreover, there are shading services (E, J) on one side of the room that are able to influence the brightness and the temperature. For the sake of simplicity, it is assumed that all these devices and their services consume electric energy provided by the external grid (1), a battery (2), and a local PV plant (3). The battery has a capacity of 4kWh as well as charging and discharging power of 2.5kW. The power loss of this storage is omitted in this example. There is an additional threshold

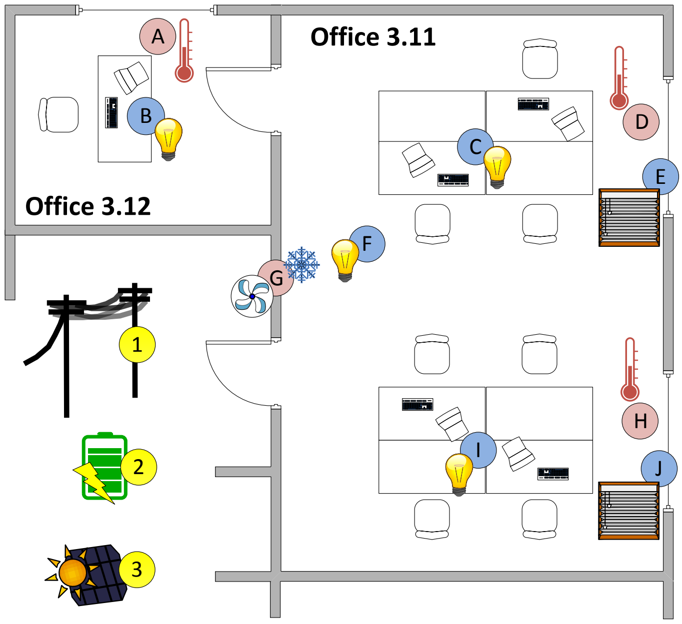

Figure 8.3: Overview of case study setting

of 1000ppm for CO_2 concentration in Office 3.11. Sensing devices for temperature, air quality, brightness, and humidity are also available in these rooms although only actuating devices and control services are visualized. Figure 8.4 illustrates some elements of this case study modeled in the ontology. In the upper part, lighting service C (*Service_C*) of Office 3.11 is shown. This control service is provided by an actuating device (*Actuator_1*) and has three ordered states (*State_1*, *State_2*, *State_3*). The functionality to influence the brightness in Office 3.11 is defined by a control variation (*Variation_B*). As specified, a higher state value leads to an uptrend regarding the brightness. On the other hand, the battery (*Battery*) as energy service provider is modeled with the energy service (*Service_2*) and the maximum capacity (*MaxCapacity*). Both services are linked to the electricity energy type (*Electricity*) connecting energy demand and energy supply. Dashed lines are used for relations that are inferred by the reasoner while solid lines represent asserted relations. The namespace *sc* is used for all properties of the utilized smart control ontology. It has to be noted that this figure visualizes only a part of the ontology in order to demonstrate the information modeling.

Next, the information is extracted from the ontology and mapped to the optimization problem. Starting with the comfort parameters, the extraction path is executed. In this example, the building structure is read in parallel. Afterwards, the sensing and actuating devices are fetched per comfort parameter and building zone. Listing 8.1 shows the basic query to get all actuating devices that are able to influence the brightness in Office 3.11 (C, E, F, I, J). The queries are dynamically created as the extraction process is aware of the semantics of properties and concepts in the ontology and knows which elements should be searched.

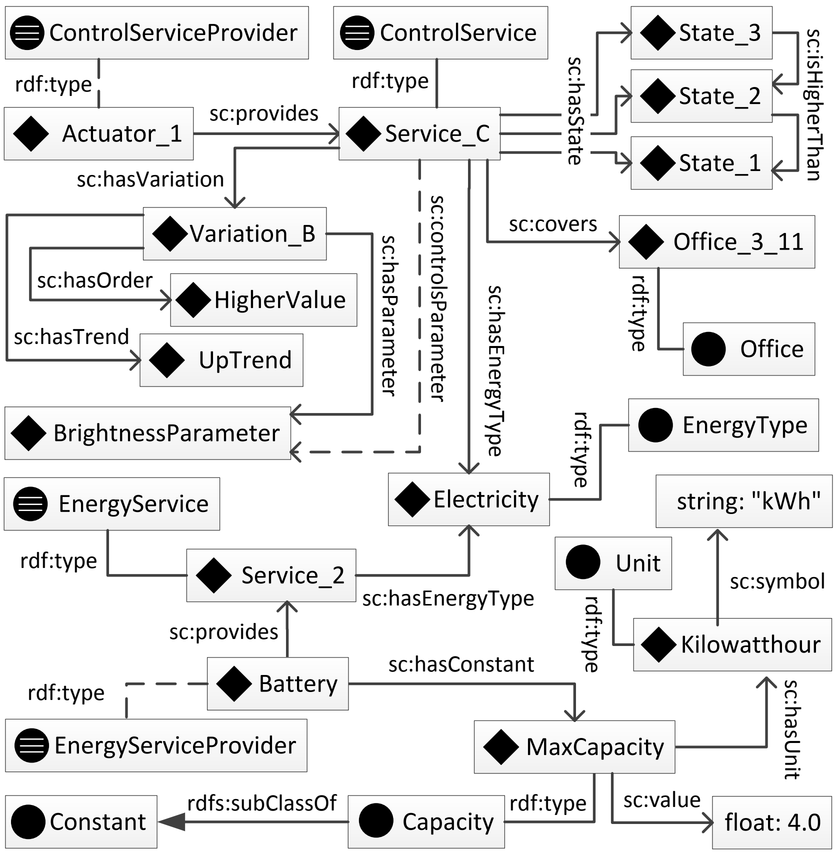

Figure 8.4: Information modeling using the ontology

Afterwards, the extracted information is mapped to the set of variables and constants that are required for the objective function and the constraints (see Table 8.1). There are four comfort parameters, namely temperature ($p = 1$), humidity ($p = 2$), air quality ($p = 3$), and brightness ($p = 4$). Control variables and corresponding lists for building zones ($z = 1..2$), energy types ($g = 1$), and energy supply units ($y = 1..3$) are initialized. The optimization period of one day is split into intervals of 15 minutes resolution ($t = 1..96$). Data structures for occupancy (o), desired comfort values (r), and immutable influences (i) are generated using the predicted or externally provided data available in the ontology. Priority values for parameters (λ) are all modeled with 1. Regarding price information (f), values for external supply from the grid are based on day-ahead signals while battery and PV plant offer their energy for free. The values for monitoring and control indicators (m) as well as energy supply indicators (j) are set to 1 except for the air quality control in Office 3.12. Here, only the manually operated window can be used for ventilation. The data structures for decision variables (s, l, q) are created but not initialized as this is the task of an optimization algorithm solving the energy management problem. The prediction functions d and v are created out-of-band depending on the implementation of the online optimization algorithm or the simulation-based methods on top of the

Listing 8.1: Get actuating devices for brightness control in Office 3.11

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX colibri: <https://[...]/colibri.owl#>

SELECT ?device ?service ?energytype
WHERE
{
  ?service colibri:covers ?zone.
  ?service colibri:controlsParameter ?param.
  ?param rdf:type ?type.
  ?device colibri:provides ?service.
  OPTIONAL {?service colibri:hasEnergyType ?energytype}.
  FILTER (?zone = <http://www.example.org/Office_3_11>) .
  FILTER (?type = colibri:BrightnessParameter)
}
```

optimization problem, but the outputs are already linked to the generated objective function. For each control service of an actuating device, an object is built that describes the functionality. For example, brightness service C is able to increase the brightness in Office 3.11 if a higher state value is used. For this purpose, the available dimming states from 1 to 3 are ordered. Finally, the constants for the charging and discharging capacity of the battery (a, u) are set to $2.5\text{kW} \cdot 0.25\text{h} = 0.625\text{kWh}$. The auxiliary variable for the energy storage level of the battery (b) is initialized with the current charge state.

While the objective function is used as described in Equations 8.1, 8.2, and 8.3, the default constraints are extended by device and building-specific conditions. The CO_2 threshold of Office 3.11 ($z = 1$) results in a constraint that is evaluated in each time slot t (Equation 8.9). A constraint limiting the maximum charge state of the battery to 4kWh is also created as this is modeled in the ontology. Furthermore, constraints are generated to specify the accepted values for all decision variables, which are either binary or based on the set of control states of a service.

$$v_{tpz} \leq 1000 \text{ with } t = 1..96, p = 3, z = 1 \tag{8.9}$$

8.6 Discussion

The presented approach for automatic generation of an optimization problem for building energy management using an ontology as underlying knowledge base has some advantages compared to the traditional design by domain experts. First, already available, semantically-enriched data can be reused to populate the ontology. In planning and construction, BIM enables a comprehensive description of buildings. Involved crafts and domains are able to publish their data based on shared vocabularies (e.g., IFC). These modeled data that describe the building structure or the facilitated BAS components can be linked to the ontology concepts of this work. Similarly, other data and information sources can be reused. Second, the optimization problem used in a BEMS does not need to be defined manually as all relevant information is available in machine-readable form. Investing time to model the expert knowledge once in the ontology enables repeated extraction in order to create BEMSs. Third, the expert knowledge can also be used by others due to the explicit definition in the ontology, which offers potential for synergy effects.

Performance of the proposed generation process directly depends on the size of the ontology. While the execution is still faster than manual instantiation of variables, constraints, and the objective function, there is some effort needed to populate the ontology. Keeping in mind that the ontology can be linked to other sources and the entire information can be reused, the approach is more efficient than initializing each BEMS separately by domain experts. Thus, companies that are active in this field of application can benefit from such an approach.

Moreover, the resulting optimization problem is the basis for different methods besides the utilization in the online scheduling of a BEMS. Defined schedules can also be evaluated via a simulation-based approach by means of this optimization problem. In this case, the prediction functions for energy demand d and comfort v match the simulation outputs while online scheduling uses data analysis methods. This might increase the accuracy of optimization runs under the condition that the utilized simulation model conforms to the real building. Using such a digital twin, schedules can be tested offline before reconfiguration of BAS needs to be done. In summary, the presented approach eases BEMS design, enables exchange of machine-readable expert knowledge, and offers a common basis for further processing.

8.7 Conclusion

BEMSs are often used to implement an energy-efficient operation of buildings by controlling BA devices. Instead of designing the underlying optimization problem manually by experts, this work presents an approach for the automatic generation of this problem based on an ontology as common knowledge base. This ontology enables the semantic modeling of all relevant information. Thus, expert knowledge for BEMS design is made available in machine-readable form to allow for automatic processing. An extraction process specifies the path of information querying in the ontology. By means of a generic objective function that combines minimization of comfort dissatisfaction and energy costs, the mapping of this information to decision variables, control variables, constants, and constraints is described. The result is an optimization problem formulation that can be used by various methods and algorithms. Applicability and functionality of the approach are discussed by means of an exemplary case study.

The next step is to implement an optimization algorithm on top of this extraction approach in order to identify issues in the optimization problem formulation. Tests and simulations of optimization runs will show if the semantics in the ontology is sufficient in order to create suitable schedules based on the optimization problem formulation. Currently, only comfort in terms of temperature, brightness, humidity, air quality, and noise level is covered. In the future, this can be extended by integration of white and brown goods into the optimization problem utilizing their potential to shift and shed loads in smart homes. Finally, the capability to trade flexibilities in the smart grid context can be added to the problem formulation increasing the applicability of the approach.

Acknowledgment

This work was funded by FFG (Austrian Research Promotion Agency) under the *BProfEM* project (845627) as well as by a netidee scholarship and the Lions Club Wien St. Stephan.

References

[1] International Energy Agency, *Transition to Sustainable Buildings: Strategies and Opportunities to 2050*. OECD/IEA, 2013, ISBN: 9789264202412.

[2] D. Dietrich, D. Bruckner, G. Zucker, and P. Palensky, "Communication and Computation in Buildings: A Short Introduction and Overview", *IEEE Transactions on Industrial Electronics*, vol. 57, no. 11, pp. 3577–3584, Nov. 2010, ISSN: 0278-0046. DOI: 10.1109/TIE.2010.2046570.

[3] P. Domingues, P. Carreira, R. Vieira, and W. Kastner, "Building automation systems: Concepts and technology review", *Computer Standards & Interfaces*, vol. 45, pp. 1–12, Mar. 2016, ISSN: 0920-5489. DOI: 10.1016/j.csi.2015.11.005.

[4] W. Kastner, M. Kofler, M. Jung, G. Gridling, and J. Weidinger, "Building Automation Systems Integration into the Internet of Things: The IoT6 approach, its realization and validation", in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2014, pp. 1–9. DOI: 10.1109/ETFA.2014.7005197.

[5] P. Palensky and D. Dietrich, "Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads", *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381–388, Aug. 2011, ISSN: 1551-3203. DOI: 10.1109/TII.2011.2158841.

[6] R. Baños, F. Manzano-Agugliaro, F. G. Montoya, C. Gil, A. Alcayde, and J. Gómez, "Optimization methods applied to renewable and sustainable energy: A review", *Renewable and Sustainable Energy Reviews*, vol. 15, no. 4, pp. 1753–1766, May 2011, ISSN: 1364-0321. DOI: 10.1016/j.rser.2010.12.008.

[7] P. H. Shaikh, N. B. M. Nor, P. Nallagownden, I. Elamvazuthi, and T. Ibrahim, "A review on optimized control systems for building energy and comfort management of smart sustainable buildings", *Renewable and Sustainable Energy Reviews*, vol. 34, pp. 409–429, Jun. 2014, ISSN: 1364-0321. DOI: 10.1016/j.rser.2014.03.027.

[8] S. Merabti, B. Draoui, and F. Bounaama, "A review of control systems for energy and comfort management in buildings", in *International Conference on Modelling, Identification and Control*, Nov. 2016, pp. 478–486. DOI: 10.1109/ICMIC.2016.7804161.

[9] F. De Angelis, M. Boaro, D. Fuselli, S. Squartini, F. Piazza, and Q. Wei, "Optimal Home Energy Management Under Dynamic Electrical and Thermal Constraints", *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1518–1527, Aug. 2013, ISSN: 1551-3203. DOI: 10.1109/TII.2012.2230637.

[10] D. L. Ha, H. Joumaa, S. Ploix, and M. Jacomino, "An optimal approach for electrical management problem in dwellings", *Energy and Buildings*, vol. 45, pp. 1–14, Feb. 2012, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2011.11.027.

[11] M. Arikiez, F. Grasso, D. Kowalski, and M. Zito, "Heuristic Algorithm for Minimizing the Electricity Cost of Air Conditioners on a Smart Grid", in *IEEE International Energy Conference*, Apr. 2016, pp. 1–6. DOI: 10.1109/ENERGYCON.2016.7513915.

[12] L. Wang, Z. Wang, and R. Yang, "Intelligent Multiagent Control System for Energy and Comfort Management in Smart and Sustainable Buildings", *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 605–617, Jun. 2012, ISSN: 1949-3053. DOI: 10.1109/TSG.2011.2178044.

[13] B. Sun, P. B. Luh, Q.-S. Jia, Z. Jiang, F. Wang, and C. Song, "Building Energy Management: Integrated Control of Active and Passive Heating, Cooling, Lighting, Shading, and Ventilation Systems", *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 588–602, Jul. 2013, ISSN: 1545-5955. DOI: 10.1109/TASE.2012.2205567.

[14] A. Ożadowicz and J. Grela, "An Event-Driven Building Energy Management System Enabling Active Demand Side Management", in *International Conference on Event-based Control, Communication, and Signal Processing*, Jun. 2016, pp. 1–8. DOI: 10.1109/EBCCSP.2016.7605082.

[15] P. Chavali, P. Yang, and A. Nehorai, "A Distributed Algorithm of Appliance Scheduling for Home Energy Management System", *IEEE Transactions on Smart Grid*, vol. 5, no. 1, pp. 282–290, Jan. 2014, ISSN: 1949-3053. DOI: 10.1109/TSG.2013.2291003.

[16] J. Beetz, J. van Leeuwen, and B. de Vries, "IfcOWL: A case of transforming EXPRESS schemas into ontologies", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 23, no. 1, pp. 89–101, Feb. 2009, ISSN: 0890-0604. DOI: 10.1017/S0890060409000122.

[17] D. Bonino and F. Corno, "DogOnt - Ontology Modeling for Intelligent Domotic Environments", in *The Semantic Web - ISWC 2008*, ser. Lecture Notes in Computer Science, A. Sheth, S. Staab, M. Dean, M. Paolucci, D. Maynard, T. Finin, and K. Thirunarayan, Eds., vol. 5318, Springer Berlin Heidelberg, 2008, pp. 790–803, ISBN: 9783540885634. DOI: 10.1007/978-3-540-88564-1_51.

[18] M. J. Kofler, C. Reinisch, and W. Kastner, "A semantic representation of energy-related information in future smart homes", *Energy and Buildings*, vol. 47, pp. 169–179, Apr. 2012, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2011.11.044.

[19] L. Daniele, F. den Hartog, and J. Roes, "Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology", in *Formal Ontologies Meet Industry*, ser. Lecture Notes in Business Information Processing, R. Cuel and R. Young, Eds., vol. 225, Springer International Publishing, 2015, pp. 100–112, ISBN: 9783319215440. DOI: 10.1007/978-3-319-21545-7_9.

[20] H. Dibowski and K. Kabitzsch, "Ontology-Based Device Descriptions and Device Repository for Building Automation Devices", *EURASIP Journal on Embedded Systems*, vol. 2011, pp. 1–17, Dec. 2011, ISSN: 1687-3955. DOI: 10.1155/2011/623461.

[21] J. Ploennigs, B. Hensel, H. Dibowski, and K. Kabitzsch, "BASont - A modular, adaptive Building Automation System Ontology", in *Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2012, pp. 4827–4833. DOI: 10.1109/IECON.2012.6389583.

[22] E. Corry, P. Pauwels, S. Hu, M. Keane, and J. O'Donnell, "A performance assessment ontology for the environmental and energy management of buildings", *Automation in Construction*, vol. 57, pp. 249–259, Sep. 2015, ISSN: 0926-5805. DOI: 10.1016/j.autcon.2015.05.002.

[23] D. Schachinger, W. Kastner, and S. Gaida, "Ontology-based abstraction layer for smart grid interaction in building energy management systems", in *IEEE International Energy Conference*, Apr. 2016, pp. 1–6. DOI: 10.1109/ENERGYCON.2016.7513991.

[24] R. Yang and L. Wang, "Multi-objective optimization for decision-making of energy and comfort management in building automation and control", *Sustainable Cities and Society*, vol. 2, no. 1, pp. 1–7, Feb. 2012, ISSN: 2210-6707. DOI: 10.1016/j.scs.2011.09.001.

[25] D. Schachinger and W. Kastner, "Semantics for smart control of building automation", in *IEEE International Symposium on Industrial Electronics*, Jun. 2016, pp. 1073–1078. DOI: 10.1109/ISIE.2016.7745041.

CHAPTER 9

Adaptive learning-based time series prediction framework for building energy management

Abstract: Sustainable building energy management is inevitable in order to reduce global energy demand. For this purpose, building energy management systems need to know the expected behavior of building automation systems, energy production units, or thermal dynamics. Designing the underlying models by domain experts might be a complex and expensive task. However, the models are already inherent in the growing amount of available monitoring data. Thus, this work proposes a framework utilizing learning-based modeling for the prediction of relevant time series in order to support comfort satisfaction and resource efficiency in building energy management. A set of neural networks is generated and trained using monitoring data and building context information modeled in an ontology. Autonomous and building-independent application is achieved by continuous performance evaluation and conditional adaption of the neural networks. The evaluation presents exemplary results and discusses the major findings.

Keywords: Energy management, buildings, forecasting, neural networks, context awareness, semantics

---

The content of this chapter has been published: D. Schachinger, J. Pannosch, and W. Kastner, "Adaptive learning-based time series prediction framework for building energy management", in *IEEE International Conference on Industrial Electronics for Sustainable Energy Systems*, Jan. 2018, pp. 453–458.

9.1 Introduction

As one of the three dominating sectors, buildings account for approximately one-third of global final energy use in 2010 with a continuous gain [1]. Measures in the form of elaborate energy management strategies are necessary consequences. Hence, building energy management systems (BEMSs) as part of building automation systems (BASs) are utilized to control, for example, heating, ventilation, and air conditioning (HVAC) or lighting systems in a building [2]. In general, a trade-off between the energy-efficient operation of building automation (BA) resources and the comfort satisfaction of building users is searched for. In this context, BEMSs usually make use of learning-based methods, model-predictive control (MPC), or agent-based control systems [3]. Literature refers to manifold approaches that focus on different aspects of energy management in residential or commercial buildings [4, 3]. Examples for used exact and heuristic methods to handle comfort management and energy consumption are fuzzy logic (FL) and genetic algorithms (GAs) [5], mixed-integer linear programming (MILP) [6], particle swarm optimization (PSO) [7], dynamic programming (DP) [8], or MPC [9].

A precondition for BEMS operation is an adequate modeling of the processes in a building. These models are used to predict future effects on the building behavior due to changes in the BAS. Based on [10] and [11], there are three main categories of prediction methods for energy consumption that can also be used to classify modeling of building behavior in general. First, the engineering method uses physical principles and structural properties of a building. This requires a lot of detailed expert knowledge and is very building-specific. Second, statistical methods create models by correlating a target output with its influencing inputs by means of historic data. Examples are autoregressive moving average (ARMA) or autoregressive integrated moving average (ARIMA). Third, artificial intelligence methods make use of historic data, as well. Nonlinear relationships between inputs and outputs are captured in order to model the behavior of a process. According to [10], modeling based on engineering methods has high accuracy but requires detailed input, which might not be available. On the other hand, artificial neural networks (ANNs) and support vector machines (SVMs) as examples of artificial intelligence methods have a similar accuracy but require only historic data. The integration of such data-driven techniques is also identified as key enabler for more intelligent BEMSs [2]. Moreover, the use of artificial intelligence methods is stimulated by the increasing amount of available data due to the rise of smart infrastructures in the context of the Internet of Things (IoT) [12]. In the BA domain, this is realized by means of server management frameworks enabling homogeneous control of formerly heterogeneous technologies [13].

In the field of building energy management, a lot of related work exists that utilizes artificial intelligence methods for the prediction of time series representing expected future behavior of building processes. For example, building energy needs are estimated based on a set of physical characteristics in [14]. Here, training is done with three different building samples. A recurrent neural network design for the prediction of electric power demand is presented in [11]. The features as well as the number of hidden neurons are determined by a GA-based optimization. Benedetti et al. propose a methodology to control accuracy and enable automatic retraining in the field of energy consumption prediction as the quality of results usually reduces over time [15]. In [16], a neural network is used to evaluate building configurations in a multi-objective optimization approach. Gareta et al. analyze the forecast of electricity prices by means of neural networks [17]. The prediction of wind power generation depending on external weather conditions using SVM is addressed in [18]. Nevertheless, neural networks are characterized by higher running speed compared to SVMs [10]. While training of neural networks is usually done using local optimization, global model optimization for cooling demand prediction is discussed in [19]. An important issue in neural network design is the feature selection. Hence, an approach for the automatic evaluation of features is presented in [20]. However, these approaches are primarily focused on energy related data. Furthermore, machine-readable semantics of the building context to ease automatic processing in BEMSs is often missing.

Thus, this work introduces a prediction framework that is not limited to forecasts of energy demand or local production but integrates also comfort-related time series that are relevant in building energy management. Neural networks are chosen as suitable data-driven modeling technique as they can learn from examples where processes are complex or hardly definable [15]. An automatic creation process is realized by the integration of machine-readable semantics regarding the building context that is defined in an ontology [21]. Based on related work, an online assessment procedure is developed to evaluate the accuracy of forecasts [15]. Moreover, a heuristic to improve the setup of neural networks is presented. As a result, the data-driven approach is building-independent and can be used without explicit expert modeling. The overall aim is to support the optimization task of BEMSs with forecasts.

In Section 9.2, the components of the prediction framework and its integration into the optimization workflow are addressed. The algorithms for identifying, designing, evaluating, and improving the prediction models are described in Section 9.3. Section 9.4 discusses the applicability and efficiency of the approach. Finally, Section 9.5 concludes the paper and gives an outlook on future work.

9.2 Time series prediction framework

The basis for a data-centric or data-driven building energy management is a central storage that concentrates data from various sources. Relevant forecast data for a particular optimization period are fetched from this storage. If they are not already provided by external sources (e.g. day-ahead energy prices, weather forecasts), these data need to be predicted internally. Therefore, a set of neural networks that model the behavior of building processes is provided by the proposed prediction framework. It has to be noted that a detailed description of neural networks is out of scope of this paper but can be found in respective literature, like [22]. The mere data storage needs to be complemented by semantics of the building structure, the BAS, the smart grid information exchange, or weather influences leading to an ontology according to [21]. While the extraction of an optimization problem based on this ontology is already addressed in [23], the automatic

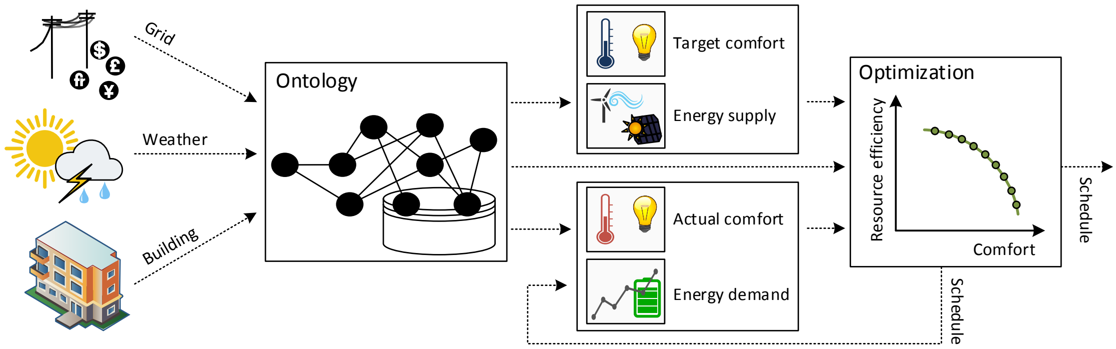

Figure 9.1: Integration of the learning-based time series prediction framework into the optimization workflow

generation and adaptive maintenance of neural networks is in the focus of this work. Finally, the gap of missing but required forecast data to support the optimization is closed. The building-independent, learning-based modeling approach relies only on the central ontology and its machine-readable semantics.

The integration of the time series prediction framework into the optimization workflow is visualized in Figure 9.1. The ontology as a sink for data and information from the building, the weather, and the smart grid is the starting point of the workflow. The periodic optimization runs require data for the generation of schedules, i.e. planned set point changes of BA resources in a particular optimization period. For this purpose, data is gathered from the ontology. While some data are directly offered by the ontology, some time series require prior execution of the neural networks. In summary, four types of prediction models in the form of neural networks are necessary in the context of building energy management.

1. The models for *target comfort* provide the optimization with desired values regarding indoor comfort parameters (e.g. brightness or temperature). If a parameter is both monitored by a sensor and controlled by an actuator of the BAS within a particular building zone (e.g. room), a corresponding neural network is created. The target values that should be reached to satisfy comfort needs are calculated once per optimization period.

2. The expected trend of *energy supply* by local production units (e.g. PV plants or wind turbines) for the optimization period has to be known in advance, as well. Hence, neural networks for each local production unit are simulated per optimization run.

3. On the other hand, the impact of a schedule on the building behavior needs to be predicted in order to evaluate the fitness of a solution. For this purpose, the estimated *actual values* of the comfort parameters in the optimization period are of interest. Again, the prediction models for this type are generated per zone

and comfort parameter in order to be independent from potential interferences in training and simulation. Compared to models of type 1 and 2, the inputs to the neural network are extended by the set points of those BA resources that have an influence on the predicted comfort parameter.

4. Finally, the impact of a schedule on the expected *energy demand* must be analyzed. For all supported energy types (e.g. electricity), neural networks are generated starting with the lowermost measurement unit and following the hierarchy bottom-up. Similar to type 3, each prediction model depends on a subset of the schedule containing all BA resources in the domain of the particular measurement unit.

9.3 Algorithmic principles

Based on the algorithms described in this section, an automated framework for time series prediction is realized that provides forecasts without considerable human intervention.

9.3.1 Prediction model identification

Some of the data required for optimization in a BEMS need to be generated by appropriate prediction models (i.e. neural networks). It is the task of the proposed framework to identify the necessity of these models by searching the context information in the ontology [21]. Time series are modeled as data services (*DataService*), which host the monitoring data. Thus, they are the root sources for learning-based modeling by means of neural networks. However, not all data services have to be forecast. Regarding energy supply (type 2), only the data services of local production units are subject to a prediction model. On the other hand, models are required for all data services that meter energy or power consumption (type 4). With respect to the desired comfort values, models are necessary for all comfort parameters that are monitored and controlled in a building zone (type 1). For this purpose, the SPARQL query in Listing 9.1 is used. The prefix *sc* addresses the namespace of the used (smart control) ontology. The result set lists

Listing 9.1: Query to get comfort data services and their dependencies

```
SELECT DISTINCT ?data ?zone ?type ?parent
WHERE
{
 ?data rdf:type sc:DataService.
 ?data sc:covers ?zone.
 ?data sc:dependsOn ?parent.
 ?control sc:covers ?zone.
 ?data sc:monitorsParameter ?paramData.
 ?control sc:controlsParameter ?paramControl.
 ?paramData rdf:type ?type.
 ?paramControl rdf:type ?type.
 ?type rdfs:subClassOf+ sc:EnvironmentalParameter.
}
```

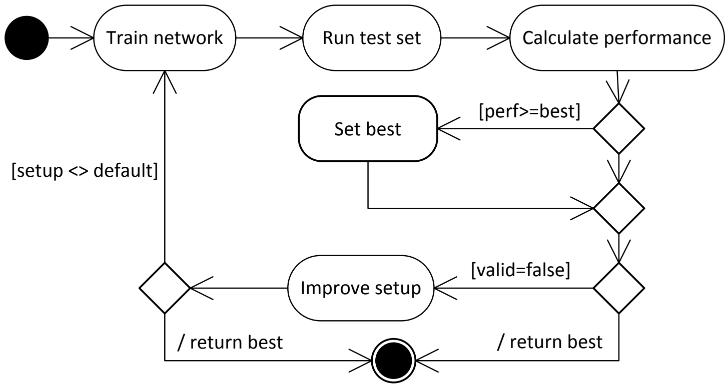

Figure 9.2: Schematic neural network design algorithm

the data services, the covered zone, the influencing data services linked by the object property *dependsOn*, and the type of the comfort parameter. Similar queries are used for the other model types, as well. Finally, the models for actual comfort values are identified (type 3). This semantics-based identification procedure is integrated into the extraction of the optimization problem as described in [23].

9.3.2 Neural network design algorithm

The identified neural networks have to be initialized and trained before they can be used to provide data for a BEMS. In Figure 9.2, a UML activity diagram sketches the underlying design algorithm. First, the network is trained using the Levenberg–Marquardt algorithm. With respect to a potential annual seasonality, the training data set is assembled accordingly. The length of this data set and some other setup options are subject to change by the improvement algorithm described in Section 9.3.4. The trained network is simulated on a test data set located between the training data set and the next optimization period. The test output is compared to the target output using the performance calculation. If the performance is better than the current best solution, the new network is set as the best network. Then, the validity of the network is checked with respect to the predefined thresholds. Valid networks are directly returned. Otherwise, the setup is modified by the improve function. If the new setup is the same as the initial default setup, the design loop is quit. This is necessary to ensure termination without performance gains. If there is a setup that was not used, the loop starts again with training the network. In any case, the result of the design algorithm is either the first valid or the best non valid network.

9.3.3 Performance calculation

Evaluating the performance (i.e. the accuracy) of a neural network forecast is an essential part of the proposed prediction framework. Similar to [15], it is checked if predefined thresholds t_i are exceeded. In this case, the network needs to be retrained or the setup has to be modified in order to ensure good forecasts for the upcoming runs. Depending on the BEMS operation cycle, performance calculations are done in periodic intervals after monitoring data are available. Based on the forecast error $e_t = y_t - f_t$ with y_t as the actual value at time t and f_t as the corresponding forecast value, several measures m_i are calculated in order to determine the performance of a network. Regarding scale-dependent errors, the mean absolute error (MAE), the maximum absolute error (MAX), and the root mean squared error (RMSE) are supported. Moreover, the mean absolute scaled error (MASE) and the symmetric mean absolute percentage error (SMAPE) are available for performance analysis. Details about these measures can be found in related literature.

For each measure m_i, the relative deviation from the corresponding threshold $t_i > 0$ multiplied by a weight $\omega_i \geq 0$ is calculated as $p_i = \frac{t_i - m_i}{t_i} \cdot \omega_i$. The output of a network is valid in general if $p_i \geq 0$ for all $i = 1..n$ with n as the number of measures. In order to compare the results of multiple performance calculations, the sum over all p_i is used as performance metric. According to the type of prediction model and the associated comfort parameter or energy production unit, the thresholds t_i and weights ω_i are configuration inputs to the framework. Thus, they can be modified by the building users. For example, temperature forecasts must be in a tolerance range of 0.5 °C, which is specified as threshold for the MAX measure of the corresponding data service. This is the only intervention point of users into the prediction framework. However, this possibility is important as thresholds might depend on the building users' needs. Initial values for thresholds and weights are evaluated empirically giving the users a basis for their modifications.

9.3.4 Improvement heuristic

When the performance indicates that a network's forecast was not accurate enough, the setup will be modified by means of an improvement heuristic. As visualized in Figure 9.3, four basic variables for this reconfiguration are specified. Regarding the input of a neural network, both the set of *features* and the length of the *time frame* of the data can influence the forecast performance. In this work, only one hidden layer is used. However, the number of *neurons* of this hidden layer is another configuration variable. The adjustable length of an optional tapped *delay line* represents the fourth variable. Long delay lines duplicate the number of connections between the input and the hidden layer. However, the longer the delay line, the more shifted historic inputs can be considered by the network. Depending on the prediction model type, a recurrent network structure might be chosen as visualized by the input $y(t)$ in Figure 9.3. Without any delay or a recurrent input, a static feed-forward neural network is created (energy production). Otherwise, a dynamic system in the form of a time delay neural network or a nonlinear autoregressive neural network with external input is used (target comfort, actual comfort, and energy demand). The used network structure is predefined and is not adjustable by this algorithm.

The improvement algorithm, first, tries to adjust the feature set. Besides the influences modeled in the ontology, cyclically encoded time features are added to the initial feature

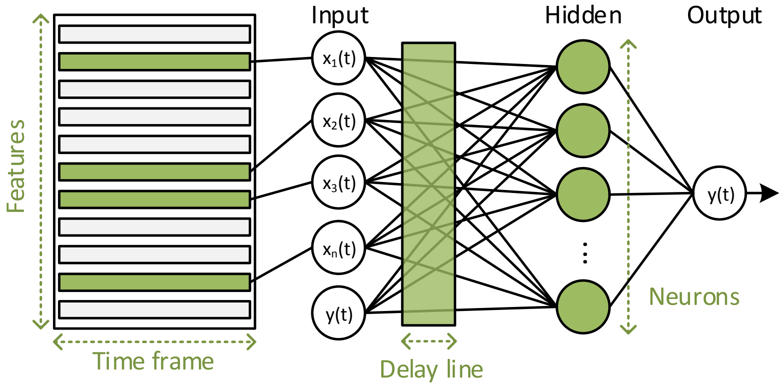

Figure 9.3: Setup options for reconfiguration of neural networks

set. While some features are mandatory, all others are removed step-by-step and added again if the performance after retraining is not increased. When the best feature set is chosen, the length of the time frame is varied in both directions between 30 and 180 days with step size of 30 days. The same is done for the number of hidden neurons within a range of 4 to 20 nodes with step size 2. If the network forecast is still not valid, the delay line is adjusted from 4 to 12 in steps of 4 delay elements. Instead of dealing with all permutations, the heuristic approach stops after a number of steps without performance gain. Then, the direction is changed or rather the next variable is considered for modification.

9.3.5 Online assessment procedure

This part of the prediction framework bridges the gap between continuous execution of prediction models by the BEMS and retraining as well as reconfiguration of these models in case of performance problems. After availability of new monitoring data, the online assessment is triggered and starts with calculating the performance as described in Section 9.3.3. If the analyzed forecast is not valid, the model is retrained using the same setup at first. This is done using the mobile training principle discussed in [15]. If this leads to a valid model, the newly trained network is returned and can be used by the BEMS. On the contrary, the setup is changed according to the heuristic described in Section 9.3.4. In summary, the online assessment is similar to the general design algorithm presented in Section 9.3.2, but it starts with a performance check before the neural network is retrained.

9.4 Evaluation and discussion

A major part of building energy consumption is allotted to thermal comfort preparation. Moreover, local energy production becomes more and more important. Thus, an appropriate EnergyPlus simulation model of a simple office building is used in this work in order to provide reproducible data for the evaluation of the proposed approach. The simulated building is located in Vienna and has three floors with five zones per floor. Furthermore, the building is equipped with an HVAC system to control the thermal comfort that is influenced by the weather (e.g. outside temperature, solar radiation), the occupancy, the lighting system, and the electric equipment. Default schedules for the HVAC set points as well as the other systems are applied. In addition, there is a PV plant for local energy supply. The building and its systems are modeled in the central ontology and complemented by monitoring data based on the simulation outputs of 2015. This is the basis for subsequent execution of the time series prediction framework including identification and design, training, and continuous improvement of the neural networks. As a detailed presentation of evaluation results would go beyond the scope of this paper, this section is focused on discussing the major outcomes of this work based on some exemplary findings.

While the data and information exchange with the ontology and the identification of necessary prediction models are realized in JAVA, the other parts of the proof-of-concept implementation are developed in MATLAB utilizing the Neural Network Toolbox. The experimental setup starts with an initial training of all identified neural networks based on a reasonable primary setup for each model. Then, prediction runs are executed on a daily basis. On the one hand, forecasts are generated using the default neural network after initial training. On the other hand, the continuously checked and improved neural networks of the prediction framework are simulated. Thus, the proposed approach can be compared to both the actual monitoring data and the predicted values of the default neural network by means of the online assessment procedure. An example is illustrated in Figure 9.4. The target indoor temperature values of a south-oriented zone on the third floor of the office building are forecast for 3 weeks in December. The blue line (—) shows the monitored data while the green line (– · –) represents the predicted data without improving the neural network. Third, the red line (– –) visualizes the forecast based on the proposed prediction framework. The vertical boxes in the figure mark days with performance problems that lead to retraining and setup modifications. It can be seen that the continuous performance assessment and improvement leads to more accurate results. In this example, the maximum error of the improved forecast is approximately 1 °C less than in the default case. Moreover, 99.5% of all errors are below 0.77 °C in the improved case and 1.92 °C otherwise. In the shown interval, performance problems arose for four days. As a first consequence, the neural network is retrained. For the first two problems, the numbers of hidden neurons and delay line elements are additionally modified. Likewise, evaluation is done for local energy production of the PV plant. As this network has much less features and the data are rather simple, both forecasts hit the monitored data quite well although the improved network provides slightly better overall results. Heating energy consumption of individual zones considering the HVAC set points is analyzed, as well. Here, the improved neural network generates more stable forecasts in most situations although the overall performance is very similar.

In general, the performance of the neural networks significantly depends on the selected training data set. Marginal shifts of the data set (e.g. by one or two days) often leads

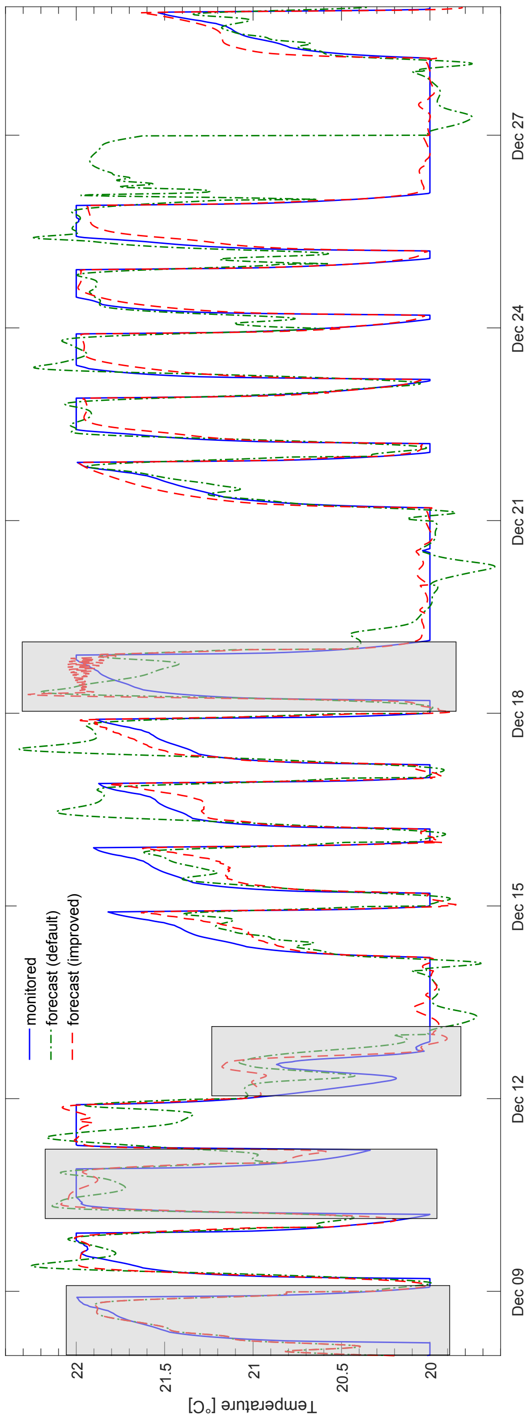

Figure 9.4: Monitored data, default forecast, and improved forecast of target indoor temperature from Dec 8, 2015 to Dec 28, 2015

to rather different outcomes. Instead of choosing a series of training data immediately before the forecast period, data with similar outdoor conditions that are further in the past can be used, as well. This can help to generate good forecasts even in transition periods (e.g. spring, fall). A precondition for this measure is that enough monitoring data are available. Moreover, the thresholds and weights for the performance assessment have to be chosen carefully in order to avoid unnecessary improvement runs or inaccurate forecasts. While this work aims at establishing the prediction framework and describing the algorithmic principles, additional mechanisms need to be introduced to address these issues. For example, the thresholds can be gradually adjusted based on the initial training and the results of the improvement heuristic. Another important issue is that the performance can only be calculated with certainty after the corresponding monitoring data are available. For periodic and regular data, performance trends can be calculated in advance indicating the probability that the next forecast will be within the defined thresholds. In this case, the prediction model can be modified prior to the problem. However, such statements are problematic for irregular, fluctuating, and unsteady time series. Thus, future work must deal with the search for a reliable method to estimate the validity of a forecast in advance. Overfitting and underfitting should not be disregarded although this is usually handled by the training algorithm. A final forecast smoothing can be done by applying filter methods on the raw data. This can reduce unwanted volatility in the output time series.

In summary, the evaluation results show that the proposed prediction framework is a suitable alternative compared to the time-consuming and complex task of modeling building processes by experts in order to support BEMSs. Although several issues have to be considered, the framework of automatic initialization and training, online assessment, and conditional setup improvement forms a transparent and universal system to facilitate BEMS operation.

9.5 Conclusion

Modeling of building processes in order to predict their behavior is an important task in building energy management. Instead of complex, expensive, and particularly building-specific modeling by domain experts, learning-based methods such as neural networks can be utilized to detect process behavior that is inherent in the growing amount of available monitoring data. This work presents an approach that unifies the prediction of all relevant time series regarding energy demand and comfort needs used in building energy management by means of an autonomous and adaptive framework. The identification and initialization of prediction models in the form of neural networks is based on an ontology that hosts the monitoring data and the corresponding semantics of the building context. Furthermore, the framework is able to detect problems in the network performance, improve the setup using a heuristic procedure, and retrain the network in order to provide reasonable forecasts for the BEMS optimization runs. In the evaluation, important outcomes as well as exemplary results are discussed.

The next step is to search for a general approximation function based on past performance trends. Then, it would be possible to detect bad forecasts with a certain probability before this forecast is actually used in the BEMS. Moreover, research to identify the cause of a performance problem can help to enhance the heuristic of finding a good neural network setup by eliminating misleading settings. In addition, the framework should be supplemented by mechanisms for automatic threshold determination and more sophisticated training set variation. A detailed evaluation based on other monitoring data sets and multiple comfort parameters is necessary in order to further analyze the reliability of the approach. Finally, the prediction framework should be tested with an optimization algorithm based on the problem formulation of [23].

Acknowledgment

This work was funded by FFG (Austrian Research Promotion Agency) under the *BProfEM* project (845627) as well as by a netidee scholarship and the Lions Club Wien St. Stephan.

References

[1] International Energy Agency, *Transition to Sustainable Buildings: Strategies and Opportunities to 2050*. OECD/IEA, 2013, ISBN: 9789264202412.

[2] M. Manic, D. Wijayasekara, K. Amarasinghe, and J. J. Rodriguez-Andina, “Building Energy Management Systems: The Age of Intelligent and Adaptive Buildings”, *IEEE Industrial Electronics Magazine*, vol. 10, no. 1, pp. 25–39, Mar. 2016, ISSN: 1932-4529. DOI: 10.1109/MIE.2015.2513749.

[3] P. H. Shaikh, N. B. M. Nor, P. Nallagownden, I. Elamvazuthi, and T. Ibrahim, “A review on optimized control systems for building energy and comfort management of smart sustainable buildings”, *Renewable and Sustainable Energy Reviews*, vol. 34, pp. 409–429, Jun. 2014, ISSN: 1364-0321. DOI: 10.1016/j.rser.2014.03.027.

[4] R. Baños, F. Manzano-Agugliaro, F. G. Montoya, C. Gil, A. Alcayde, and J. Gómez, “Optimization methods applied to renewable and sustainable energy: A review”, *Renewable and Sustainable Energy Reviews*, vol. 15, no. 4, pp. 1753–1766, May 2011, ISSN: 1364-0321. DOI: 10.1016/j.rser.2010.12.008.

[5] S. Merabti, B. Draoui, and F. Bounaama, “A review of control systems for energy and comfort management in buildings”, in *International Conference on Modelling, Identification and Control*, Nov. 2016, pp. 478–486. DOI: 10.1109/ICMIC.2016.7804161.

[6] F. De Angelis, M. Boaro, D. Fuselli, S. Squartini, F. Piazza, and Q. Wei, “Optimal Home Energy Management Under Dynamic Electrical and Thermal Constraints”, *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1518–1527, Aug. 2013, ISSN: 1551-3203. DOI: 10.1109/TII.2012.2230637.

[7] L. Wang, Z. Wang, and R. Yang, “Intelligent Multiagent Control System for Energy and Comfort Management in Smart and Sustainable Buildings”, *IEEE Transactions on Smart Grid*, vol. 3, no. 2, pp. 605–617, Jun. 2012, ISSN: 1949-3053. DOI: 10.1109/TSG.2011.2178044.

[8] B. Sun, P. B. Luh, Q.-S. Jia, Z. Jiang, F. Wang, and C. Song, “Building Energy Management: Integrated Control of Active and Passive Heating, Cooling, Lighting, Shading, and Ventilation Systems”, *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 588–602, Jul. 2013, ISSN: 1545-5955. DOI: 10.1109/TASE.2012.2205567.

[9] C. Chen, J. Wang, Y. Heo, and S. Kishore, “MPC-Based Appliance Scheduling for Residential Building Energy Management Controller”, *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1401–1410, Sep. 2013, ISSN: 1949-3053. DOI: 10.1109/TSG.2013.2265239.

[10] H.-x. Zhao and F. Magoulès, “A review on the prediction of building energy consumption”, *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 3586–3592, Aug. 2012, ISSN: 1364-0321. DOI: 10.1016/j.rser.2012.02.049.

[11] H. R. Khosravani, M. D. M. Castilla, M. Berenguel, A. E. Ruano, and P. M. Ferreira, “A Comparison of Energy Consumption Prediction Models Based on Neural Networks of a Bioclimatic Building”, *Energies*, vol. 9, no. 1, pp. 1–24, Jan. 2016, ISSN: 1996-1073. DOI: 10.3390/en9010057.

[12] A. Whitmore, A. Agarwal, and L. Da Xu, “The Internet of Things—A survey of topics and trends”, *Information Systems Frontiers*, vol. 17, no. 2, pp. 261–274, Apr. 2015, ISSN: 1387-3326. DOI: 10.1007/s10796-014-9489-2.

[13] P. Domingues, P. Carreira, R. Vieira, and W. Kastner, “Building automation systems: Concepts and technology review”, *Computer Standards & Interfaces*, vol. 45, pp. 1–12, Mar. 2016, ISSN: 0920-5489. DOI: 10.1016/j.csi.2015.11.005.

[14] B. B. Ekici and U. T. Aksoy, “Prediction of building energy consumption by using artificial neural networks”, *Advances in Engineering Software*, vol. 40, no. 5, pp. 356–362, May 2009, ISSN: 0965-9978. DOI: 10.1016/j.advengsoft.2008.05.003.

[15] M. Benedetti, V. Cesarotti, V. Introna, and J. Serranti, “Energy consumption control automation using Artificial Neural Networks and adaptive algorithms: Proposal of a new methodology and case study”, *Applied Energy*, vol. 165, pp. 60–71, Mar. 2016, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2015.12.066.

[16] E. Asadi, M. G. da Silva, C. H. Antunes, L. Dias, and L. Glicksman, “Multi-objective optimization for building retrofit: A model using genetic algorithm and artificial neural network and an application”, *Energy and Buildings*, vol. 81, pp. 444–456, Oct. 2014, ISSN: 0378-7788. DOI: 10.1016/j.enbuild.2014.06.009.

[17] R. Gareta, L. M. Romeo, and A. Gil, “Forecasting of electricity prices with neural networks”, *Energy Conversion and Management*, vol. 47, no. 13–14, pp. 1770–1778, Aug. 2006, ISSN: 0196-8904. DOI: 10.1016/j.enconman.2005.10.010.

[18] Z. Zhu, D. Zhou, and Z. Fan, “Short term forecast of wind power generation based on SVM with pattern matching”, in *IEEE International Energy Conference*, Apr. 2016, pp. 1–6. DOI: 10.1109/ENERGYCON.2016.7514064.

[19] R. Yokoyama, T. Wakui, and R. Satake, “Prediction of energy demands using neural network with model identification by global optimization”, *Energy Conversion and Management*, vol. 50, no. 2, pp. 319–327, Feb. 2009, ISSN: 0196-8904. DOI: 10.1016/j.enconman.2008.09.017.

[20] S. F. Crone and N. Kourentzes, “Feature selection for time series prediction – A combined filter and wrapper approach for neural networks”, *Neurocomputing*, vol. 73, no. 10–12, pp. 1923–1936, Jun. 2010, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2010.01.017.

[21] D. Schachinger and W. Kastner, “Semantics for smart control of building automation”, in *IEEE International Symposium on Industrial Electronics*, Jun. 2016, pp. 1073–1078. DOI: 10.1109/ISIE.2016.7745041.

[22] B. M. Wilamowski, “Understanding Neural Networks”, in *The Industrial Electronics Handbook: Intelligent Systems*, B. M. Wilamowski and J. D. Irwin, Eds., Second Edition, CRC Press, 2011, ch. 5, pp. 1–11, ISBN: 9781439802830. DOI: 10.1201/b10604.

[23] D. Schachinger and W. Kastner, “Ontology-based generation of optimization problems for building energy management”, in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2017, pp. 1–8. DOI: 10.1109/ETFA.2017.8247565.

CHAPTER 10

Context-aware optimization strategies for universal application in smart building energy management

Abstract: In building operation, the continuous forward planning of energy-efficient schedules to maintain user comfort is a challenging task. Although the design of building energy management systems is an active field of research, existing solutions are often faced with limited reusability due to specialization on certain buildings, comfort parameters, or building automation technologies. Thus, this work introduces a set of context-aware strategies that are generally applicable for the optimization in building energy management systems. For this purpose, machine-readable semantics of the building and the building automation system is exploited in order to design a heuristic approach. The aim is to reduce the optimization effort while targeting both energy efficiency and cross-domain comfort satisfaction on a building-independent level. An embedding of the proposed approach into common metaheuristics is described to provide a basis for further reuse. Finally, case studies are used for evaluation of a proof-of-concept implementation.

Keywords: Building automation, context awareness, energy management, optimization, semantics, smart buildings

The content of this chapter has been published: D. Schachinger and W. Kastner, "Context-aware optimization strategies for universal application in smart building energy management", in *IEEE International Conference on Industrial Informatics*, Jul. 2018, pp. 478-483.

10.1 Introduction

In general, building automation systems (BASs) are able to maintain an energy-efficient building operation [1]. Nevertheless, studies show that global energy demand of buildings will rise by 50% between 2010 and 2050 under unchanged conditions [2]. In order to reverse this trend, advanced control and management of BASs are necessary to release their potentials in energy savings. Hence, building energy management systems (BEMSs) as part of or supplement to BASs are utilized to ensure energy-efficient building operation in conformance with building users' comfort requirements [3]. Major requirements for advanced BEMSs include the applicability independent of the building type, the support of various kinds of equipment, the integration of decentralized energy resources, an easy design and implementation, and little need for training and expert knowledge [4]. Although the focus of these requirements is on heating, ventilation, and air conditioning (HVAC) systems, they are relevant for BEMSs in general.

A suitable basis to overcome these issues is provided by establishing interoperability and introducing additional machine-readable semantics regarding the building context, the utilized equipment, or the relevant external and internal influences. Modeling and processing of semantics using knowledge engineering methods act as key enabler for smart buildings as part of emerging smart grids or smart cities. In particular, BEMSs will benefit from an abstract and machine-readable representation of expert knowledge with regard to universal applicability and reusability. Recent research identified ontologies as suitable method for structured semantic modeling. For example, the CTRLont ontology covers the explicit modeling of control logic in the building automation (BA) domain [5]. Advanced description of device functionality is presented in [6]. Based on this, the BASont ontology addresses relevant use cases in the BA life cycle [7]. In the context of building information modeling (BIM), the Industry Foundation Classes (IFC) are mapped to the ifcOWL ontology [8]. Brick aims at representing building infrastructure as basis for smart building applications [9]. The SAREF ontology for smart appliances enables modeling of sensing, actuating, or metering devices as well as energy profiles or price information [10].

On the other hand, efficient optimization is required as there might be a lot of variables and constraints even for short optimization periods [11]. In literature, various methods and algorithms are used to tackle the multi-objective goal of reducing comfort dissatisfaction and minimizing energy consumption or costs. Shaikh et al. summarize related work with a focus on comfort and energy management in smart buildings [12]. Utilized methods are, for example, mixed-integer linear programming (MILP), genetic algorithms (GAs), or artificial neural networks (ANNs). In [13], classical control approaches are compared to predictive and adaptive controllers with respect to the nonlinear behavior in the HVAC domain. A BEMS architecture based on evolutionary algorithms is presented in [11]. Thermal comfort in home energy management based on MILP is discussed in [14]. In [4], advanced model-based control methods for BEMSs are compared, such as dynamic programming (DP) or model-predictive control (MPC). A heuristic-based approach for cost minimization of air conditioning operation is proposed in [15]. In [16], particle swarm optimization (PSO) is used in a multi-zone BEMS optimization. HVAC, lighting, and shading are targeted by means of DP in [17]. Although this related work provides notable solutions for BEMSs, the approaches are often tailored to particular buildings, comfort parameters, or BA domains. Design and configuration are often done manually, which limits reconfiguration and reuse in other settings.

Thus, this work introduces universally applicable optimization strategies by analyzing general characteristics of optimization problems in BEMSs. The approach takes advantage of a semantically enriched system representation that is used to gain knowledge about relations between building zones, comfort parameters, set point changes, sensor measurements, external weather influences, or supply of locally produced energy. The aim is to design an intelligent search for feasible solutions of the optimization problem on an abstract, building-independent level. Furthermore, the integration into common metaheuristics is presented in order to provide a reusable and applicable basis for individual BEMS implementations. Case studies are used to evaluate a proof-of-concept implementation, which is based on the variable neighborhood descent (VND) metaheuristic.

10.2 Background and semantic requirements

Basically, optimization in BEMSs tries to find a suitable solution for the conflicting goals of minimizing comfort dissatisfaction and minimizing energy consumption or costs in a given optimization period. Multiple domains have an impact on the result of this process. Figure 10.1 summarizes the most important influencing factors that are in strong relation with each other. Building zones provide the spatial basis for optimization. Actions in these building zones are set by actuators and controllers of the BAS leading to a more or less significant change in comfort and energy consumption. Monitoring data of various comfort parameters, on the other hand, are provided by sensors of the BAS. Comfort targets and constraints are defined by building users. Weather conditions can influence local energy supply and building comfort behavior. Regarding the integration into smart grids, variable energy tariffs and grid emergency signals are supported while flexibility trading and feed-in of surplus energy are out of scope of this work.

Information about these domains needs to be modeled in an ontology in order to provide machine-readable semantics for further processing in the optimization. For this purpose, the smart control ontology is reused and extended [18]. Runtime or configuration data (*data services*) and basic device functionality to influence comfort parameters (*control services*) as well as energy supply and energy demand of resources (*energy services*) can be described on top of the physical situation (*building zone*, *resource*). A set of object properties enables the definition of functional and temporal dependencies between services (*affects*, *follows*, *precedes*). Moreover, their embedding into the building context regarding the sphere of influence or the zone affiliation needs to be defined (*covers*, *monitors*, *controls*, *consumes*, *provides*). Knowledge for intelligent decision making in the optimization process is queried from this semantic representation. For example, information about the influence of state changes of a heating control service on the

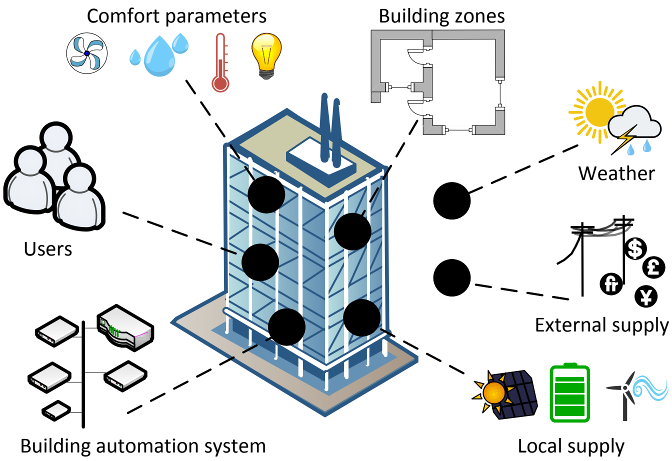

Figure 10.1: Relevant influencing factors for optimization in BEMSs

temperature in a certain building zone can be used in order to increase the thermal comfort. On the other hand, new findings on the spatial or temporal impact of taken actions during the optimization are fed back into the ontology.

According to the optimization problem formulation in [19], the output of the optimization is a predetermined execution schedule covering states $s_{x,t}$ per BA resource x, states $s_{y,t}$ for each energy supplier y including storage resources, and quotas $q_{y,t}$ of energy provided per supplier y in each time slot t of the optimization period. Depending on the number of devices, the building size, and the granularity and length of the optimization period, the space of feasible and infeasible solutions can explode. Finding optimal solutions might be time-consuming or even impossible. Thus, this work uses the abstract context information, which can be easily modeled in the ontology without specific expert knowledge, in order to reduce the solution space instead of searching all possible combinations. The developed strategies can be used subsequently in smart buildings as basis for individual BEMS implementations.

10.3 Optimization strategies

The smart search of the solution space using problem-specific knowledge is realized in three steps. First, the most expensive subproblems in terms of comfort and energy costs in a time-discrete optimization period are identified (Section 10.3.1). Second, the schedule is partially modified with respect to the isolated subproblem (Section 10.3.2). Third, the impacts of the performed changes are analyzed to infer new knowledge for further optimization (Section 10.3.3).

10.3.1 Subproblem identification

Following the divide and conquer paradigm, the overall problem of schedule optimization is first divided into subproblems using problem-specific knowledge in order to lower complexity. First, the objective value of solutions (i.e. schedules) needs to be calculated.

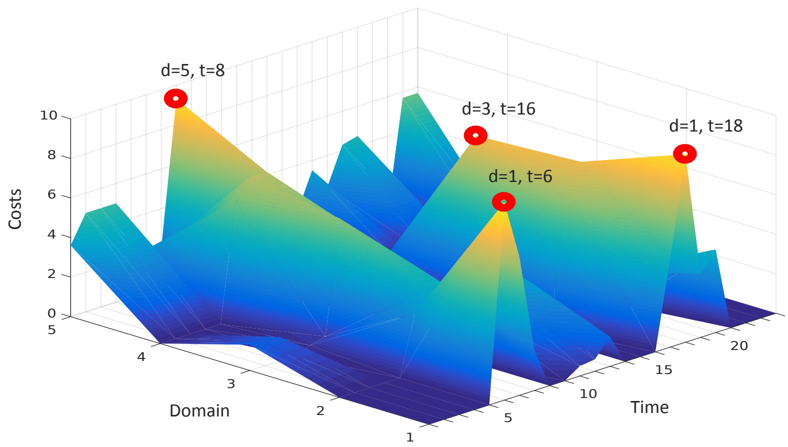

Figure 10.2: Exemplary cost distribution with most expensive subproblems

As described in [19], deviations between target and actual values of comfort parameters per building zones are transformed into comfort costs and summed up with the costs of energy supply that are necessary to balance the required demand. Priority factors are used to harmonize costs of different comfort parameters while a weight factor supports the conversion between comfort costs and energy costs. Thus, an $n \times m$ matrix of comparable cost components can be derived with n as the number of time slots, and m as the number of comfort domains (zone/parameter) and energy domains (grid/supplier). As higher costs indicate potential problems in the schedule under consideration, a set of the c most expensive cost components is further investigated. According to the actual costs, the components are prioritized such that higher costs lead to higher priority. Another ranking criterion is the time slot of cost occurrence. Here, earlier slots are higher ranked. The priorities are proportional to the selection probability in a subsequent, randomized selection using the roulette-wheel principle. Finally, a subproblem determined by domain d and time slot t is returned. The surface plot in Figure 10.2 visualizes a cost matrix with five domains ($m = 5$) and 24 time slots ($n = 24$). The marked elements (red dots) represent the candidate cost components ($c = 4$).

A list of visited subproblems is stored to decide on final termination of the optimization. However, subproblems can be removed from this list because of indirect changes due to interferences in building behavior. Hence, a tabu list of length u is used to avoid cyclic selection in the identification process.

10.3.2 Partial modification

Next, the selected subproblem is conquered by estimating the cause of the high costs and identifying solutions in the form of state changes. The output of the partial modification is an altered schedule. Figure 10.3 visualizes the modification schema in a simplified and abstract way. The shown exemplary problem as input for the modification is the deviation of brightness comfort from the intended user target in a building zone

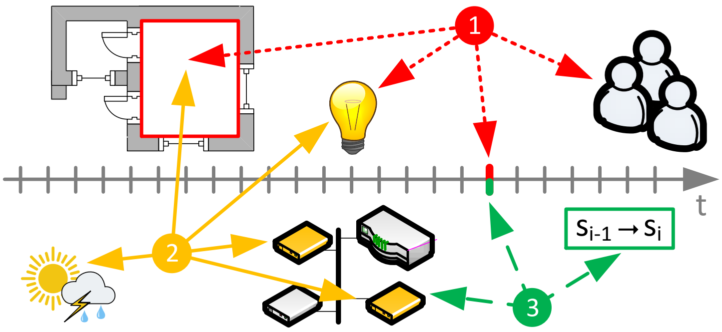

Figure 10.3: Simplified modification schema

at a certain time (1). In contrast, problems regarding energy costs are related to a supplier and a local grid at a time. The modification looks for appropriate solution candidates (i.e. sets of state changes) by querying the ontology (2). A neighborhood N that describes the possible sets of state changes is generated in proximity to the subproblem. In this context, each state change of a resource at a time is called *move*. In order to form more complex neighborhoods, moves can be concatenated to *paths* resulting in a set of nested neighborhoods $N_1 \subset N_2 \subset \cdots \subset N_k$ covering the solution space. For example, neighborhood N_2 contains paths of 2 consecutive moves. In general, the characteristics of BA resources, like their set of ordered states or their impact on building zones, comfort parameters, or energy demand, are considered. Moreover, external influences (e.g. weather, energy prices) as well as user-defined and physical constraints (e.g. thresholds, operating cycles) are taken into account. The temporal impact of state changes on dynamic building processes influences the neighborhood size, as well. For problems representing high comfort costs, the algorithm builds the neighbors with respect to reduction or increase of the actual comfort value to a lower or higher target value, respectively. Impact dimension, relative trends, corresponding energy demands, or time delays between state change and reaction are considered. On the other hand, high energy costs can be lowered by substituting energy-intensive resources by more energy-efficient resources or states. Ideally, a path is found leading to nearly the same comfort impact at reduced energy consumption. Moreover, energy demand is reduced by detecting comfort over-fulfillment. With respect to the costs of energy supply to balance the required demand, locally produced energy has the highest priority. By default, demand that overshoots local production is saturated by external supply. Thus, storage resources are utilized to shift demand to periods of lower costs.

Essential in the modification procedure is the efficient search of the neighborhood and the selection of a probably good candidate (3). In order to avoid visiting all possible neighbors, an intelligent navigation strategy is necessary. While the solution space is actually reduced to feasible paths by neighborhood design, prioritization biases the path selection. Basically, more suitable paths are higher ranked. In general, the interferences between energy demand and comfort are investigated. First, the direct influences of paths on the problem's domain in value and time are analyzed. Then, the indirect impacts on other domains are estimated. Available knowledge about specific state change impacts, generic control variations, and basic relations in the building context is exploited to determine the overall priority of a path. Finally, the priority-proportional selection of a path is based on a randomization, which is a widely-used measure in heuristics to avoid getting stuck in a local minimum. Hence, good candidates are chosen with higher probability while others have also a (minor) chance of being selected. All feasible moves or paths are reachable, but some are preferred over others. The modification is terminated if certain thresholds in solution quality and number of iterations are reached or all paths of a neighborhood are visited.

10.3.3 Impact assessment

An important characteristic of a smart system is its ability to learn from made experiences and infer new knowledge for upcoming challenges. In the context of BEMSs, the lessons that can be learned are related to the impact of state changes on comfort and energy domains. Future optimization runs can benefit from this additional knowledge by writing it back into the ontology. Thus, a faster convergence towards a global optimum can be achieved.

Comparing the results of schedule S_{i-1} and schedule S_i, significant differences $\Delta_{d,t}$ per domain d and time slot t regarding comfort satisfaction or energy demand can be determined. Moreover, the temporal impact λ_d of a state change on a domain d is analyzed in order to get information about the delay or inertia of a building process. Based on this analysis, *specific rules* are derived that describe the exact impact of a move in delay and value with respect to a domain. Depending on the actual situation (e.g. weather, occupancy), the value of the impact as well as the delay may vary, which necessitates an averaging. *Generic rules* are derived that follow the principle of control variations in the underlying ontology. The total ordering of resource states $s_0 < s_1 < \cdots < s_l$ and the sign of the deviation $\mathrm{sgn}(\Delta_{d,t})$ are the basis for the inference of general statements about the relation of a resource's state trend ($s\uparrow$, $s\downarrow$) to a domain's value trend ($d\uparrow$, $d\downarrow$). For example, a state change $\langle s_{i-1} \rightarrow s_i \rangle$ with $s_i < s_{i-1}$ leads to $s\downarrow$ while a value trend $d\downarrow$ is the result of $\mathrm{sgn}(\Delta_{d,t}) = -1$. Finally, *basic relations* between BA resources, comfort parameters, building zones, and local grids are established by further abstracting from specific and generic rules.

The level of detail in describing the impacts of moves can be varied. A reasonable tradeoff between the processing efforts of very complex impact descriptions compared to the effective benefit needs to be found. In summary, this impact assessment and the self-consistent rule inference are a supplement to classical ontology reasoning. The byproducts of consecutive schedule evaluation are recycled to gradually improve the knowledge about the managed system.

10.4 Embedding into common metaheuristics

For optimization problem solving in BEMSs, exact and especially linear algorithms might not be feasible due to the potentially high complexity of the combinatorial problem and the nonlinear behavior of some building processes. On the other hand, metaheuristics are independent of particular optimization problems and describe abstract problem solving to find good solutions in reasonable time even for hard problems. The wide range of metaheuristics can be divided into single-solution and population-based metaheuristics [20]. Thus, metaheuristics fit to this work's overall aim of universal applicability and complement the context-aware optimization strategies. The embedding of the functions to divide (subproblem identification) and conquer (partial modification) the optimization problem as well as to learn from changes in the schedule (impact assessment) into common metaheuristics that are described in [20] is discussed in the following. This integration provides a suitable basis for BEMS implementations.

Starting with single-solution metaheuristics, *local search (LS)* looks for improvements in a specific neighborhood of a given solution. Neighbor selection is realized with partial modification. After fitness evaluation, the impact assessment can be executed. LS is called for each identified subproblem. Similarly, the embedding into the *variable neighborhood descent (VND)* can be implemented. In contrast to LS, VND provides a set of multiple neighborhoods. Hence, an optimum with respect to all neighborhood structures is searched. The *general variable neighborhood search (GVNS)* utilizes VND to improve randomly generated neighbors. Here, two different neighborhood structures are used. Again, impact assessment is used after the fitness evaluation. Partial modification remains a part of VND. However, an adapted version of the modification function is combined with the problem identification to form the outer neighborhood structure for random shaking. The *simulated annealing (SA)* principle to accept worse schedules with a certain probability can be used, for example, after the selection of a new neighbor.

A famous representative of population-based metaheuristics is the *genetic algorithm (GA)*. Following the principles of biological evolution, a set of parents is selected for recombination. The resulting children are mutated before they replace worse solutions of the population. Impact assessment is basically used in the evaluation phase of each GA iteration. The mutation is based on partial modification of a random cost component. Optionally, subproblem identification can be used to provide a starting point for partial modification. Moreover, problem identification is part of the recombination to support the selection of suitable recombination points. The *ant colony optimization (ACO)* is inspired by swarm intelligence. Global information in the form of pheromones and local information are considered. Every ant creates a complete solution (i.e. schedule) that can be based on one of the other metaheuristics using the proposed strategies. However, the neighborhood navigation in the partial modification is additionally based on the emergent swarm knowledge to improve solution convergence.

10.5 Evaluation

Based on a proof of concept (Section 10.5.1), the strategies are evaluated by means of case studies (Section 10.5.2). Moreover, results and findings are discussed (Section 10.5.3).

Algorithm 1 Extended variable neighborhood descent *(S)*

1: **while** stopping criteria not satisfied **do**
2:     $p \leftarrow$ identify subproblem *(S)*
3:     $k_{max} \leftarrow$ get neighborhoods *(S, p)*
4:     $k \leftarrow 1$
5:     **while** $k < k_{max}$ **do**
6:         **while** no improvement & neighbors unvisited **do**
7:             $S' \leftarrow$ run partial modification *(S, k, p)*
8:             assess impacts *(S, S′)*
9:         **end while**
10:         **if** $f(S') < f(S)$ **then**
11:             $S \leftarrow S'$
12:             $k \leftarrow 1$
13:         **else**
14:             $k \leftarrow k+1$
15:         **end if**
16:     **end while**
17: **end while**
18: **return** S

10.5.1 Proof-of-concept implementation

The Java-based proof of concept uses the triple store Apache Jena to manage the ontology. The building context modeled in an OWL file is loaded into the application. Runtime data such as time series for comfort targets as well as start schedules are imported from spreadsheet files, which ease editing outside of the application. For traceability, simplified models are used to predict influences on comfort and energy domains instead of utilizing data-driven forecasting methods for fitness evaluation.

The implemented algorithm extends the VND metaheuristic. Algorithm 1 shows the corresponding pseudo code. Variable initialization and error handling are mostly omitted for better readability. The VND (Lines 4-16) is wrapped in a loop to iterate through the identified subproblems. Based on the input solution S, a subproblem p and its number of neighborhoods k_{max} are determined. In the VND loop, the schedule S is modified by searching the neighborhoods for new neighbors according to problem p (Lines 6-9). Impact assessment is executed after the return of a new neighbor S'. In case of an improvement, schedule S' is set as new best schedule. Termination is primarily based on the number of overall iterations and the number of consecutive unsuccessful iterations.

10.5.2 Case studies

Diverse case studies are defined to evaluate the proof-of-concept implementation. This section gives an overview on the mode of operation of the proposed strategies and their interaction with the ontology in common BEMS situations. Thus, a subset of these case studies is chosen that cover building processes with different dynamics (visual and thermal comfort) as well as issues emerging from the use of electric vehicles (Table 10.1). These case studies are rather small, but the focus is on analyzing the basic principle of solution space reduction and impact assessment. Privacy (#1) and mobility (#3) are artificial comfort parameters. The brightness targets (#1) are designed in accordance with EN 12464-1. Weights are chosen to compare comfort and energy costs with respect to the case study. A vehicle-to-grid scenario is out of scope (#3). An EnergyPlus simulation is used to evaluate the impacts of state changes in the heating schedule (#2). Other processes use simplified mathematical models. Charging and discharging power are assumed to be constant. For all case studies, the optimization period is set to $n = 24$ time slots. The impacts of schedule changes are continuously assessed leading to a set of inferred rules that speed up convergence of solution quality. Examples for inferred generic rules are listed in Table 10.1.

10.5.3 Discussion

The results of the evaluation indicate advantages of the approach as well as challenges that need to be addressed in further research. The ontology that is used in the optimization strategies is continuously amended by newly inferred knowledge. This is a key feature to improve the finding of proper state changes after dividing the overall optimization problem in smaller subproblems. Scalability with respect to more complex instances (i.e. buildings) depends on the extent of knowledge about the relations between BA resources, comfort domains, or energy consumption in the building context. Both artificial (e.g. privacy) and real comfort parameters (e.g. temperature) can be combined due to the abstract semantic modeling. Moreover, equipment that cannot be controlled automatically (e.g. windows) can be integrated by means of suitable human-machine interfaces to instruct building users.

A critical issue is the definition of priorities for the randomized selection of feasible moves and paths in the partial modification. If priorities are too close together, selection degenerates to a pure random selection that ignores the significant differences of state changes. Moreover, scaling factors between conflicting comfort parameters and the conversion weight between comfort and energy costs need to be specified with care. Thresholds for termination or the length of the tabu list must be adapted to the time available for an optimization run or the length of the optimization period. Finally, population-based methods should use a set of preferably diverse, randomly generated starting solutions while the execution schedule of the previous optimization period might be a good choice for single-solution heuristics.

In summary, the proposed context-aware optimization approach represents a counterpart of specific solutions that are tailored to individual buildings or comfort domains. In the latter, expert knowledge is manually integrated during the design process. On the other hand, the ontology-based semantic modeling used in this work offers a base for abstract and reusable optimization in BEMSs. Although performance of specialized implementations can be better, the presented approach offers intelligent strategies for universal applicability independent of building type, size, or equipment. Furthermore,

Table 10.1: Summary of selected case studies

	Case study #1	**Case study #2**	**Case study #3**
Task	Tradeoff between privacy, brightness, and energy consumption	Centralized room heating with different room targets	Electric vehicle charging with fluctuating energy prices
Resources	Blind ($0\% \prec 100\%$, $0\% =$ up), lamp ($0\% \prec 100\%$, $0\% =$ off)	Central heating ($15°C \prec 30°C$)	PV, battery (discharge $\prec$ off $\prec$ charge), electric vehicle (off $\prec$ charge)
Zones	Home office	South-facing room (A), north-facing room (B)	Office building
Influences	Occupancy (0/1), sunlight (lx), energy tariff (€)	Outdoor temperature (°C), solar radiation (W/m^2), energy tariff (€)	PV production (kW), energy tariff (€), vehicle availability (true/false)
Domains	Electric energy (kWh), brightness (lx), privacy (sat/unsat)	Temperature A/B (°C), humidity A/B (%), electric energy (kWh)	Mobility (%), electric energy (kWh)
Initial situation	High priority on privacy, lamp always on (100%), blind always up (0%), privacy required in the evening	Start schedule of 18°C in A and B, temperature target of 21-23°C in A, temperature target of 19-21°C in B	High energy prices (12pm-3pm), target mobility (charge state) of 70% (4pm), initial charge of battery at 50%
Observed behavior	First, high costs of privacy are tackled. Blind is shut and light is turned on although sun is shining. Energy costs in these slots are lower than comfort deviation costs. During day, lamp is turned off if sunlight is enough.	Initially, there is high comfort deviation in all slots and domains. As central heating influences both rooms, schedule converges to tradeoff. Solar radiation can reduce set point at noon with reduced energy costs but higher comfort costs in B (less radiation).	Battery is used when vehicle charging needs more energy than provided from PV. Vehicle is available from 11am to 4pm. Battery is charged in some slots before 12pm to have enough (cheaper) energy for vehicle charging.
Example rules	$s_{lamp} \uparrow \Rightarrow d_{brightness} \uparrow$ $s_{blind} \uparrow \Rightarrow d_{privacy} \uparrow$	$s_{heating} \uparrow \Rightarrow d_{temperature} \uparrow$ $s_{heating} \downarrow \Rightarrow d_{temperature} \downarrow$	$s_{vehicle} \uparrow \Rightarrow d_{mobility} \uparrow$ $s_{vehicle} \uparrow \Rightarrow d_{electricity} \uparrow$

the machine-readable information can be used in several other use cases in the smart building context.

10.6 Conclusion

Optimization algorithms in BEMSs need to be efficient in order to plan ahead energy-saving but comfort-compliant BAS schedules. Existing approaches for specific buildings, comfort domains, or equipment are often limited in their reusability as expert knowledge is directly integrated in the implementation. Semantic modeling enables the mapping of this expert knowledge to an abstract, explicit, and machine-readable representation. As a result, building-independent and universal optimization algorithms can be created, which can be easily applied to manifold settings. Therefore, this work introduces an approach to design the optimization in BEMSs by means of context-aware strategies based on additional semantics. An ontology is used to support subproblem identification, partial modification, and impact assessment in BAS schedule optimization. Embedding of these modular strategies into popular metaheuristics offers a suitable base for actual implementations. The evaluation of a proof-of-concept implementation using case studies shows the functional capability and the applicability of this smart building approach.

The next step is to compare the presented VND implementation with realizations of this approach based on other metaheuristics. Moreover, the computational performance and the neighborhood generation of the proof of concept will be improved in order to evaluate the suitability and applicability in more complex scenarios. Regarding the impact assessment, active consideration of external and internal influences should be integrated to better estimate side effects and interferences before new knowledge is written to the ontology. With respect to flexibility trading in the smart grid, both the fitness function of the optimization problem and the partial modification procedure need to be adapted. Finally, a comprehensive performance analysis is required to make statements on the supported building size, the maximum optimization period, or the achievable solution quality compared to execution time.

Acknowledgment

This work was funded by a netidee scholarship and the Lions Club Wien St. Stephan.

References

[1] J. Haase, G. Zucker, and M. Alahmad, “Energy efficient building automation: A survey paper on approaches and technologies for optimized building operation”, in *Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2014, pp. 5350–5356. DOI: 10.1109/IECON.2014.7049317.

[2] International Energy Agency, *Transition to Sustainable Buildings: Strategies and Opportunities to 2050.* OECD/IEA, 2013, ISBN: 9789264202412.

[3] M. Manic, D. Wijayasekara, K. Amarasinghe, and J. J. Rodriguez-Andina, “Building Energy Management Systems: The Age of Intelligent and Adaptive Buildings”, *IEEE Industrial Electronics Magazine*, vol. 10, no. 1, pp. 25–39, Mar. 2016, ISSN: 1932-4529. DOI: 10.1109/MIE.2015.2513749.

[4] E. Atam, “New Paths Toward Energy-Efficient Buildings: A Multiaspect Discussion of Advanced Model-Based Control”, *IEEE Industrial Electronics Magazine*, vol. 10, no. 4, pp. 50–66, Dec. 2016, ISSN: 1932-4529. DOI: 10.1109/MIE.2016.2615127.

[5] G. F. Schneider, P. Pauwels, and S. Steiger, “Ontology-Based Modeling of Control Logic in Building Automation Systems”, *IEEE Transactions on Industrial Informatics*, vol. 13, no. 6, pp. 3350–3360, Dec. 2017, ISSN: 1551-3203. DOI: 10.1109/TII.2017.2743221.

[6] H. Dibowski and K. Kabitzsch, “Ontology-Based Device Descriptions and Device Repository for Building Automation Devices”, *EURASIP Journal on Embedded Systems*, vol. 2011, pp. 1–17, Dec. 2011, ISSN: 1687-3955. DOI: 10.1155/2011/623461.

[7] J. Ploennigs, B. Hensel, H. Dibowski, and K. Kabitzsch, “BASont - A modular, adaptive Building Automation System Ontology”, in *Annual Conference of the IEEE Industrial Electronics Society*, Oct. 2012, pp. 4827–4833. DOI: 10.1109/IECON.2012.6389583.

[8] P. Pauwels and W. Terkaj, “EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology”, *Automation in Construction*, vol. 63, pp. 100–133, Mar. 2016, ISSN: 0926-5805. DOI: 10.1016/j.autcon.2015.12.003.

[9] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal, M. Bergés, D. Culler, R. K. Gupta, M. B. Kjærgaard, M. Srivastava, and K. Whitehouse, “Brick: Metadata schema for portable smart building applications”, *Applied Energy*, vol. 226, pp. 1273–1292, Sep. 2018, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2018.02.091.

[10] L. Daniele, F. den Hartog, and J. Roes, “Created in Close Interaction with the Industry: The Smart Appliances REFerence (SAREF) Ontology”, in *Formal Ontologies Meet Industry*, ser. Lecture Notes in Business Information Processing, R. Cuel and R. Young, Eds., vol. 225, Springer International Publishing, 2015, pp. 100–112, ISBN: 9783319215440. DOI: 10.1007/978-3-319-21545-7_9.

[11] I. Mauser, J. Müller, F. Allerding, and H. Schmeck, “Adaptive building energy management with multiple commodities and flexible evolutionary optimization”, *Renewable Energy*, vol. 87, pp. 911–921, Mar. 2016, ISSN: 0960-1481. DOI: 10.1016/j.renene.2015.09.003.

[12] P. H. Shaikh, N. B. M. Nor, P. Nallagownden, I. Elamvazuthi, and T. Ibrahim, “A review on optimized control systems for building energy and comfort management of smart sustainable buildings”, *Renewable and Sustainable Energy Reviews*, vol. 34, pp. 409–429, Jun. 2014, ISSN: 1364-0321. DOI: 10.1016/j.rser.2014.03.027.

[13] S. Merabti, B. Draoui, and F. Bounaama, “A review of control systems for energy and comfort management in buildings”, in *International Conference on Modelling, Identification and Control*, Nov. 2016, pp. 478–486. DOI: 10.1109/ICMIC.2016.7804161.

[14] F. De Angelis, M. Boaro, D. Fuselli, S. Squartini, F. Piazza, and Q. Wei, “Optimal Home Energy Management Under Dynamic Electrical and Thermal Constraints”, *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1518–1527, Aug. 2013, ISSN: 1551-3203. DOI: 10.1109/TII.2012.2230637.

[15] M. Arikiez, F. Grasso, D. Kowalski, and M. Zito, “Heuristic Algorithm for Minimizing the Electricity Cost of Air Conditioners on a Smart Grid”, in *IEEE International Energy Conference*, Apr. 2016, pp. 1–6. DOI: 10.1109/ENERGYCON.2016.7513915.

[16] R. Yang and L. Wang, “Multi-zone building energy management using intelligent control and optimization”, *Sustainable Cities and Society*, vol. 6, pp. 16–21, Feb. 2013, ISSN: 2210-6707. DOI: 10.1016/j.scs.2012.07.001.

[17] B. Sun, P. B. Luh, Q.-S. Jia, Z. Jiang, F. Wang, and C. Song, “Building Energy Management: Integrated Control of Active and Passive Heating, Cooling, Lighting, Shading, and Ventilation Systems”, *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 3, pp. 588–602, Jul. 2013, ISSN: 1545-5955. DOI: 10.1109/TASE.2012.2205567.

[18] D. Schachinger and W. Kastner, “Semantics for smart control of building automation”, in *IEEE International Symposium on Industrial Electronics*, Jun. 2016, pp. 1073–1078. DOI: 10.1109/ISIE.2016.7745041.

[19] ——, “Ontology-based generation of optimization problems for building energy management”, in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2017, pp. 1–8. DOI: 10.1109/ETFA.2017.8247565.

[20] M. Gendreau and J.-Y. Potvin, Eds., *Handbook of Metaheuristics*, Second Edition, ser. International Series in Operations Research & Management Science. Springer US, 2010, ISBN: 9781441916655. DOI: 10.1007/978-1-4419-1665-5.

List of Figures

List of Tables

List of Listings

List of Algorithms

List of Acronyms

6LoWPAN IPv6 over Low-Power Wireless Personal Area Networks.

ACL Agent communication language.

ACO Ant colony optimization.

ANN Artificial neural network.

ANSI American National Standards Institute.

API Application programming interface.

ARIMA Autoregressive integrated moving average.

ARMA Autoregressive moving average.

ASHRAE American Society of Heating, Refrigerating, and Air-Conditioning Engineers.

BA Building automation.

BACnet Building Automation and Control Networks.

BACnet/WS BACnet Web services.

BAS Building automation system.

BEMS Building energy management system.

BIM Building information modeling.

BMS Building management system.

CIM Common Information Model.

CoAP Constrained Application Protocol.

DER Decentralized energy resource.

DP Dynamic programming.

DPWS Devices Profile for Web Services.

DR Demand response.

DSL Digital subscriber line.

DSM Demand side management.

EI Energy Interoperation.

EMF Eclipse Modeling Framework.

EMS Energy management system.

ETS Engineering Tool Software.

EXI Efficient XML Interchange.

FL Fuzzy logic.

GA Genetic algorithm.

GSM Global System for Mobile Communications.

GVNS General variable neighborhood search.

HTTP Hypertext Transfer Protocol.

HVAC Heating, ventilation, and air conditioning.

ICT Information and communication technology.

IEA International Energy Agency.

IEC International Electrotechnical Commission.

IEEE Institute of Electrical and Electronics Engineers.

IFC Industry Foundation Classes.

IoT Internet of Things.

IP Internet Protocol.

IPv6 Internet Protocol version 6.

ISO International Organization for Standardization.

IT Information technology.

JID Jabber Identifier.

JSON JavaScript Object Notation.

JSONP JSON with padding.

KNX WS KNX Web services.

LAN Local area network.

LS Local search.

LSB Least significant bit.

M2M Machine-to-machine.

MAE Mean absolute error.

MAS Multi-agent system.

MASE Mean absolute scaled error.

MAX Maximum absolute error.

MDA Model-Driven Architecture.

MDE Model-Driven Engineering.

MILP Mixed-integer linear programming.

MPC Model-predictive control.

MQTT Message Queue Telemetry Transport.

MSB Most significant bit.

OASIS Organization for the Advancement of Structured Information Standards.

OBIX Open Building Information Exchange.

OMG Object Management Group.

OPC UA OPC Unified Architecture.

OpenADR Open Automated Demand Response.

OSI Open Systems Interconnection.

OWL Web Ontology Language.

PKI Public key infrastructure.

PLC Powerline communication.

PSO Particle swarm optimization.

QoS Quality of Service.

QVT Query/View/Transformation.

RA Room automation.

RDF Resource Description Framework.

RDFS RDF Schema.

REST Representational State Transfer.

RMSE Root mean squared error.

RTP Real-time pricing.

SA Simulated annealing.

SEP Smart Energy Profile.

SIP Session Initiation Protocol.

SMAPE Symmetric mean absolute percentage error.

SOA Service-oriented architecture.

SOAP Simple Object Access Protocol.

SPARQL SPARQL Protocol and RDF Query Language.

SPARUL SPARQL/Update.

SSN Semantic Sensor Network.

SVM Support vector machine.

TCP Transmission Control Protocol.

TLS Transport Layer Security.

TOU Time-of-use.

UDP User Datagram Protocol.

UMTS Universal Mobile Telecommunications System.

URI Uniform resource identifier.

VND Variable neighborhood descent.

WAN Wide area network.

WS Web service.

XML Extensible Markup Language.

XMPP Extensible Messaging and Presence Protocol.

Curriculum Vitae

Personal information

First name, Surname	**Daniel Schachinger**
Address	Bauernstraße 23, 4645 Grünau (Austria)
E-mail	danielschachinger@gmx.at

Work experience

	TU Wien, Institute of Computer Engineering
since 07/2014	**Research assistant** Research projects in the field of automation systems, assistance in the courses "Home and Building Automation" and "Wireless in Automation"
08/2013 – 06/2014	**Project staff** Research projects in the field of automation systems
10/2010 – 02/2014	**Tutor** Assistance in the courses "Dezentrale Automation", "Formale Modellierung", and "Datenmodellierung"

Education

since 10/2014	**Doctoral program in Engineering Sciences** TU Wien Area of concentration: Computer Sciences Dissertation: "Building Energy Management Optimization based on a Semantic Abstraction Layer"
10/2011 – 06/2014	**Master program in Computer Engineering**, Dipl.-Ing., passed with distinction TU Wien Diploma thesis: "Model-Driven Engineering for Building Automation Systems"
10/2010 – 11/2014	**Bachelor program in Business, Economics and Social Sciences**, BSc (WU) WU Wien Major: Business Administration Specializations: Production Management, Information Management Bachelor thesis: "A Motion Measurement Stack"
10/2008 – 09/2011	**Bachelor program in Computer Engineering**, BSc, passed with distinction TU Wien Bachelor thesis: "Real-Time Monitoring und WCET Analyse im Kontext der Runtime Verification"
09/2002 – 06/2007	**Secondary College of Engineering**, Matura, passed with distinction HTBLA Leonding, Department of Electronic Data Processing and Economics Project work: "CAREER.MANAGER" Diploma thesis: "eLogon"

Awards

02.06.2017	Award **Lions Förderpreis 2017** (Lions Club Wien St. Stephan)
17.11.2016	Scholarship **netidee Stipendium 2016** (Internet Foundation Austria)
16.10.2015	Award **Fred Margulies Preis 2015** (ÖGART, Österreichischer IFAC-Beirat)
20.08.2014	Admission to the **Rector's List** in summer term 2014 (WU Wien)
10/2008 – 06/2013	Granting of **Scholarship for Excellence** (TU Wien)

Publications

Articles

D. Schachinger, A. Fernbach, and W. Kastner, “Modeling framework for IoT integration of building automation systems”, *at – Automatisierungstechnik*, vol. 65, no. 9, pp. 630–640, Sep. 2017.

Conference papers

D. Schachinger and W. Kastner, “Context-aware optimization strategies for universal application in smart building energy management”, in *IEEE International Conference on Industrial Informatics*, Jul. 2018, pp. 478–483.

D. Schachinger, J. Pannosch, and W. Kastner, “Adaptive learning-based time series prediction framework for building energy management”, in *IEEE International Conference on Industrial Electronics for Sustainable Energy Systems*, Jan. 2018, pp. 453–458.

D. Schachinger and W. Kastner, “Ontology-based generation of optimization problems for building energy management”, in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2017, pp. 1–8.

F. Petrushevski, S. Gaida, B. Beigelböck, M. Sipetic, G. Zucker, C. Schiefer, D. Schachinger, and W. Kastner, “Semantic Building Systems Modeling for Advanced Data Analytics for Energy Efficiency”, in *Building Simulation*, Aug. 2017, pp. 622–627.

D. Schachinger and W. Kastner, “Semantic interface for machine-to-machine communication in building automation”, in *IEEE International Workshop on Factory Communication Systems*, May 2017, pp. 1–9.

D. Schachinger and W. Kastner, “Integration von KNX Netzwerken in das Internet der Dinge: Die KNX Web Services Spezifikation”, in *Jahreskolloquium Kommunikation in der Automation*, Nov. 2016, pp. 1–8.

D. Schachinger, S. Gaida, W. Kastner, F. Petrushevski, C. Reinthaler, M. Sipetic, and G. Zucker, “An Advanced Data Analytics Framework for Energy Efficiency in Buildings”, in *IEEE International Conference on Emerging Technologies and Factory Automation*, Sep. 2016, pp. 1–4.

D. Schachinger and W. Kastner, “Semantics for smart control of building automation”, in *IEEE International Symposium on Industrial Electronics*, Jun. 2016, pp. 1073–1078.

D. Schachinger, W. Kastner, and S. Gaida, “Ontology-based abstraction layer for smart grid interaction in building energy management systems”, in *IEEE International Energy Conference*, Apr. 2016, pp. 1–6.

D. Schachinger, C. Stampfel, and W. Kastner, “Interoperable integration of building automation systems using RESTful BACnet Web services”, in *Annual Conference of the IEEE Industrial Electronics Society*, Nov. 2015, pp. 3899–3904.

D. Schachinger, S. Gaida, and W. Kastner, “Smart grid communication at the interface of customer buildings with focus on demand response”, in *International Symposium on Smart Electric Distribution Systems and Technologies*, Sep. 2015, pp. 368–373.

D. Schachinger and W. Kastner, “Model-Driven Integration of Building Automation Systems into Web Service Gateways”, in *IEEE World Conference on Factory Communication Systems*, May 2015, pp. 1–8.

T. Reinbacher, J. Brauer, D. Schachinger, A. Steininger, and S. Kowalewski, “Automated Test-Trace Inspection for Microcontroller Binary Code”, in *International Conference on Runtime Verification*, Sep. 2011, pp. 239–244.

Industry conference papers	D. Schachinger and W. Kastner, "IoT integration based on the KNX Web services specification", in *KNX Scientific Conference*, Oct. 2016, pp. 1–12.
	D. Schachinger, M. Jung, and W. Kastner, "KNX and Model-Driven Engineering", in *KNX Scientific Conference*, Oct. 2014, pp. 1–12.
Talks	D. Schachinger and L. Krammer, "Building Profiling for Energy Management", *Blickpunkt Forschung: Smart Communities and Technologies @ TU Wien*, Vienna, Austria, Oct. 2016.
	D. Schachinger and W. Kastner, "Future Energy – Diskussion & Zukunftsbild", *IMAGINE15*, Vienna, Austria, Jun. 2015.