

Article

Engineering Support for Handling Controller Conflicts in Energy Storage Systems Applications

Claudia Zanabria ^{*},  Ali Tayyebi, Filip Pröbstl Andrén, Johannes Kathan and Thomas Strasser 

Center for Energy–Electric Energy Systems, AIT Austrian Institute of Technology, 1210 Vienna, Austria; Ali.Tayyebi-Khameneh@ait.ac.at (A.T.); Filip.Proestl-Andren@ait.ac.at (F.P.A.); Johannes.Kathan@ait.ac.at (J.K.); Thomas.Strasser@ait.ac.at (T.S.)

* Correspondence: claudia.zanabria@ait.ac.at

Received: 29 August 2017 ; Accepted: 28 September 2017; Published: 13 October 2017

Abstract: Energy storage systems will play a major role in the decarbonization of future sustainable electric power systems, allowing a high penetration of distributed renewable energy sources and contributing to the distribution network stability and reliability. To accomplish this, a storage system is required to provide multiple services such as self-consumption, grid support, peak-shaving, etc. The simultaneous activation of controllers operation may lead to conflicts, as a consequence the execution of committed services is not guaranteed. This paper presents and discusses a solution to the exposed issue by developing an engineering support approach to semi-automatically detect and handle conflicts for multi-usage storage systems applications. To accomplish that an ontology is developed and exploited by model-driven engineering mechanisms. The proposed approach is evaluated by implementing a use case example, where detection of conflicts is automatically done at an early design stage. Besides this, exploitable source code for conflicts resolution is generated and used during the design and prototype stages of controllers development. Thus, the proposed engineering support enhances the design and development of storage system controllers, especially for multi-usage applications.

Keywords: energy storage systems; energy management system; engineering support; smart grid architecture model; model-driven engineering; rapid prototyping; ontology; semantic web technologies; description logic; conflicts resolution

1. Introduction

A sustainable electric power supply requires the integration of renewable, Distributed Energy Resources (DER) [1]. In addition, Energy Storage Systems (ESS) presents interesting solutions for a higher share of DERs and also to support several stakeholders in electrical systems besides their contribution to maintaining power quality, reducing energy costs, and enhancing grid stability/reliability [2]. In this context, the main goal of small residential storage systems is to consume the self-generated electricity to reduce energy costs. On the other hand, they may also contribute to the enhancement of power quality and system stability (primary control reserve in a pooling scheme [3] and voltage/frequency droop control [4]). Storage systems with higher capacities provide additional economic profits and support with much more services such as participation on balancing market services, power trading, ancillary services, etc. Hence, cost-effective solutions for households/industries and a simultaneous provision of services to stakeholders require the development of a multi-use storage system approach [5]. To this aim, a wide range of controller approaches such as open/close loop, multi-agent systems, optimization methods, etc. are implemented locally/externally to the ESS. A challenge derived from this occurs when the interaction of those controllers is omitted and not handled, resulting in an unexpected controller behavior. For example,

the ELECTRA IRP project [6] points out this issue and develops an analysis of controller conflicts in scenarios with a high penetration of DERs. Concluding that overlapping of controllers could lead to the non-provision of committed services, destabilizing the whole system and in some cases harming the power quality of electrical grid. For instance an over-frequency event requires charging the ESS, at the same time the market operator requires to discharge the battery to optimize energy costs. In this case, a lack of a support to coordinate the mentioned services would prevent the battery of charging/discharging enough power to cover the requirements defined by frequency support and market services. Additionally, a battery discharging behavior, resulted from the overlapping of use cases, could incentive the exacerbation of the over-frequency state of the grid [3]. Hence, both services cannot be supported all at once, then one potential solution is the setting up of priorities per service. The “EERA Joint Programme on Smart Grids” [7], presents examples of multi-functional ESS, one example carries out secondary control power and peak/base arbitrage, those services are simultaneously supported. The alignment of them is based on the allocation of battery capacities. Another example is presented in [8], a voltage control combined with an increased self-consumption strategy is provided by a photovoltaic-battery system, an automatic voltage limitation strategy is run to coordinate the provision of both services. The aforementioned approaches justify the needed for a mechanism to analyze and handle the conflicts within a multi-service ESS context.

The development process of multi-use ESS applications should consider the identification of controllers conflicts in an early stage, before any practical implementation such as the design and elaboration of control schemes, deployment of exploitable source code, execution of offline simulations, etc. An anticipate detection of conflicts would allow to select the right control scheme configuration and to save time on doing unusable and profitable work, then reaching the promised services. Base on this, the current work proposes a methodology for a semi-automatic identification and handling of conflicts within a multi-use ESS, this process is carried out during the specification stage. Additionally, this methodology is used as a support for the rapid prototyping of ESS control applications. Two methods are currently employed to model power system control applications. They are the Smart Grid Architecture Model (SGAM) [9] and the IEC 62599 approach [10], those methodologies attempt to gather and model knowledge of a specific smart grid use case. However, models defined by them are not enough for a full identification and handling of conflicts. A preliminary idea that foster the mentioned statement is briefly outlined in [11]. It proposes a first version of a SGAM-based data model for a partial identification of conflicts within a multi-use ESS application. This work presents a first classification of conflicts and a selected conflict type is analyzed by the SGAM-based model, arguing that SGAM aligned with the IEC 62599 approach need to be extended for a further conflict analysis. In contrast with the mentioned paper, this work extends the aforementioned one by designing and implementing not only a data model but an ontology for a comprehensive analysis and conflicts resolution. Moreover, Model-Driven Engineering (MDE) exploits the proposed ontology to support domain experts during the controllers development process.

The paper is structure as follows: Section 2 addresses the related work, the process development of storage systems applications and the benefits of applying ontology and MDE-based concepts to resolve potential ESS controller conflicts and to support the development of ESS applications. In Section 3 a comprehensive classification of conflicts is provided, enabling the design of an ontology for conflicts identification. As a sequel, in Section 4 conflict resolution approaches are encouraged, motivating the broadening of the ontology to fully cover identification and handling. Finally, in Section 5 the resulted ontology is evaluated by a selected example and exploited by the MDE approach followed by the conclusions and an outlook about the future research work in Section 6.

2. Related Work and Background

2.1. Conflicts Within Storage Systems Use Cases

A brief overview of potential multi-functional Use Cases (UC) related to ESS participation mainly in low voltage power distribution grids is provided in Table 1 including a classification of services per objective and corresponding stakeholder(s). These use cases assume a small/medium scale ESS equipped with an Energy Management System (EMS) to implement and manage the provision of different services [11,12]. Hence, a multi-use ESS is reached. Those services are deployed either remotely (i.e., EMS set-points based on external decision making) or locally (i.e., set-point evaluation based on local available EMS data) as also used in the examples in Section 4. Both deployments require to set up a communication infrastructure between the EMS and external systems such as network operators (e.g., Transmission System Operator (TSO), Distribution System Operator (DSO)) and meters. To illustrate this a communication architecture that carries out the set-up of an EMS connected to an ESS is shown in Figure 1. Besides of this, the EMS sets active and reactive power values to the ESS in order to control the energy injected/consumed from the grid. In this context, a main challenge to deal with the EMS development is the coordination of single-services, thus to identify the occurrence of conflicts.

Table 1. Typical multi-functional storage system use cases.

Use Case/Name	Objective	Stakeholder	Description & Examples
electric energy time shift/ UC_1	market integration	market operator	Economical benefits are maximized. ESS is charged when the spot market prices are low and during off-peak times and usually discharged when prices are high. It is based on a daily optimization strategy [7].
voltage control/ UC_2	power quality	DSO	The rise of voltage levels can be regulated by injecting reactive power or by consuming active power. ESS participates in voltage regulation by implementing different control strategies [8,13].
primary control reserve/ UC_3	power system stability	TSO	A high penetration of DER may result in a change of the grid frequency. The ESS participates in the frequency regulation by injecting/consuming active power [3,14]. Such a service is usually used by the TSO.
peak-shaving/ UC_4	reduction of supply cost	end user	ESS is used to prevent high peaks of consumption. The ESS is discharged when a load higher than a specific set-point is switch on [15].
minimization of prices/ UC_5	reduction of supply cost	end user	Electricity costs are minimized by an objective function. It takes into account the cash received from selling energy and the cash paid for energy consumed. Moreover, forecast on load consumption, DER device generation and electricity prices (€/kWh) are calculated [16].
self-consumption/ UC_6	reduction of supply cost	end user	Local self-consumption is the main target. The difference of the local generation and demand is charged into the battery and discharged from it when the demand exceeds the local generation [7].

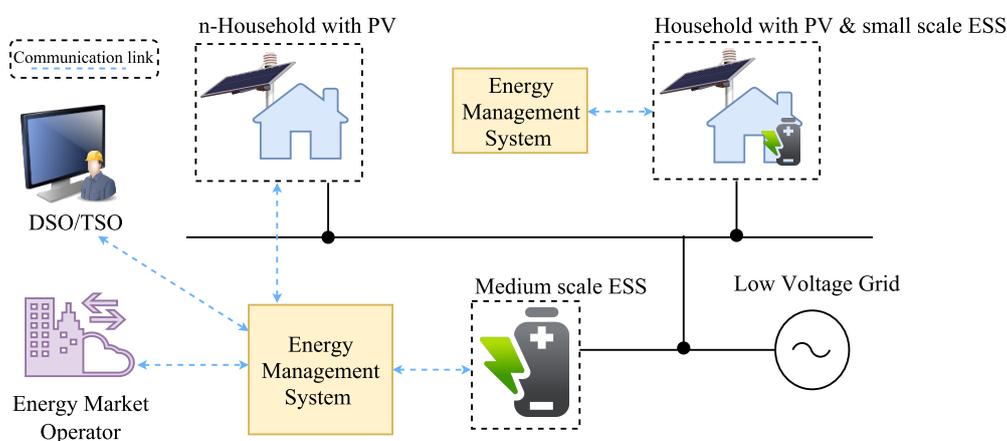


Figure 1. Communication architecture of a multi-use ESS approach.

A conflict may appear from the simultaneous interaction of stakeholders (e.g., DSO and Energy Market Operator (EMO)) orientated services. This may lead to an inefficient behavior of the EMS controller, as well as to the no provision of committed services. A conflict identification approach that targets large flexible power system architectures (e.g., virtual power plants, large-scale deployment of DER) is addressed by the ELECTRA IRP project [6]. In this study, scenarios of controller conflicts and suggestions for conflict's resolution are presented. Furthermore, a methodology for conflict identification based on a set of advises such as detailed use case specification, impact of controllers interactions on system stability, classification of conflicts by acceptable and unacceptable, etc. is presented. Against previous concept, this work proposes an engineering support to semi-automatically identify and resolve controllers conflicts in a specific power system domain: a multi-use storage system. To afford this, a domain expert or user needs to provide information about a multi-functional ESS, this process is manually done and is the starting point to proceed with a fully-automatic identification of conflicts. Additionally, handling approaches are proposed in the form of code and specific models, a validation and correction of them needs to be done manually by the user. Moreover, the proposed approach is developed in a generic way to be used in any ESS control scheme configuration. To this end, the implementation and simulation of services proposed in Table 1 is a meaningful first step to analyze the occurrence of potential conflicts and to derive a classification of causes of conflicts occurrence, this classification is carried out in Section 3.1.

2.2. Storage Systems Application Development

The main steps to be carried out during the development of ESS applications are the design, prototype and realization stages [17,18]. In the first one a conceptual design of the EMS controller is elaborated and verified. It involves a model of the electrical network, the ESS converter controller and corresponding systems or physical devices (e.g., photovoltaic inverter, load profile, Supervisory Control and Data Acquisition (SCADA)). As a sequel, the design of control algorithms to fulfill a set of services requirements is carried out. Those requirements vary depending on time-scale and accuracy. For instance, a voltage control use case requires a response time of at least 60 s and a maximum voltage deviation of $\pm 6.5\%$. Following this, a proof of concept is carried out to validate the initial specifications. This is mainly done in an offline simulator for power system simulations (e.g., Matlab/Simulink, DlgSILENT/PowerFactory) [19]. After the conceptual design is verified, a prototype is realized. Often the transformation from concept design into a prototype entails communication delays or non-linearities issues. Then, an iterative refinement of the algorithms is carried out. The realization stage requires testing the validated prototype in a real environment.

This work is focused on the analysis of requirements at the design phase with the aim of identifying conflicts between control algorithms. Additionally, possible handling solutions for those conflicts are advised and deployed during software and following hardware-based tests (e.g., pure offline and Controller-Hardware-in-the-Loop (CHIL) simulations). An early conflict detection would enable a free-of-conflict storage system application at the design phase, before the elaboration of any hands-on work such as control algorithms models implementation. Additionally, this would also support the selection of the right control scheme for a multi-functional ESS approach.

2.3. Rapid Prototyping of ESS Applications

Ontologies abstract information of a specific domain with the aim to represent objects, type of objects, and their semantic relations in a formal machine-readable way. Additionally, a set of inference rules and restrictions on relations are defined to support semantic interpretation [20]. This kind of modeling approach has also already been successfully applied to the domain of power systems: for example, reference [21] has designed an ontology to model the coordination between building energy management systems and smart grids stakeholders in a demand response context. The main entities to be modeled are communication technologies, service interfaces and grid structure. Thanks

to reasoning techniques, a matching of services and their communication technology is achieved. Besides of this, reference [22] develops an ontology by applying fuzzy theory to diagnostic faults in power transformers, demonstrating that fuzzy ontology identifies faults that are unidentifiable by a basic ontology model. The mentioned studies highlight the use of key features of ontologies such as automatic reasoning and share of knowledge in the smart grid domain. In contrast to the mentioned studies, this paper is focused on a specific power system domain: multi-use storage systems connected to low voltage power distribution grids. This system is analyzed, resulting in the development of an ontology that targets controllers conflicts. This ontology is also used to handle the localized conflicts. To achieve this, data consistency checking and inference of knowledge—assets of ontologies [23]—are performed.

A further interesting modeling approach—Model-Driven Design (MDD) or Model-Driven Engineering (MDE)—focuses on an abstract representation of knowledge by the usage of models specification [24]. An advantage of this is to support the development of software applications by improving the interoperability between systems. In this context, two main types of transformations are considered: (i) Model-to-Text transformation (M2T) that generates executable source code, documentation or other text files from source models; and (ii) Model-to-Model transformation (M2M) that transforms source models into specific target models by executing pre-defined transformation rules [25]. MDE and SGAM are lately employed in smart grid projects to gain a common understanding of smart grid architecture elements [26] and to foster the rapid prototyping of smart grid applications [18,27]. In this context, the proposed ontology is aligned to the SGAM model and exploited by the means of M2T techniques.

In summary, this paper joins concepts from MDE and ontologies to improve the rapid prototyping of multi-use ESS applications and to handle conflicts derived from the overlapping of corresponding applications. Hence, an engineering support consisting in a methodology is derived.

3. Ontologies for Multi-Use ESS Conflicts Identification

The scope of this section is to introduce a methodology supporting the identification of multi-use EES conflicts. This process is carried out during the specification phase of controllers, resulting in an appropriate control scheme to be implemented. To this end, a classification of conflicts based on their causes is proposed. Additionally, this classification is formally modeled by ontologies resulting in an EMS-ontology, a data model that targets conflicts identification. As a sequel, reasoning mechanisms are carried out to entail the deduction of conflicts.

3.1. Categorization of Controller Conflicts

The main objective of a multi-functional ESS control is to provide more than one service by injecting/consuming active or reactive power (either with locally or remotely connected EMS controllers). It is illustrated in Figure 2, where an EMS sets the power to be injected/consumed by the ESS, additionally the EMS communicates with external systems (e.g., DSO, TSO, EMO) depending on the configuration of the EMS controllers. The interaction between those controllers could lead to an undesired behavior of the ESS system and a non-provision of the corresponding service. Because of this, the reason(s) causing the conflicts need to be identified and investigated. Table 2 provides a corresponding overview of typical controller conflicts resulting from the multi-use of ESS (extended from [11]).

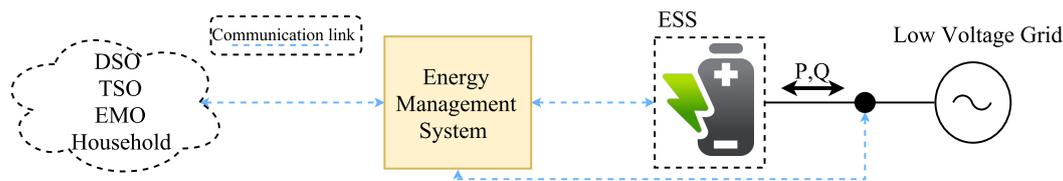


Figure 2. Injection/consumption of power to address a multi-use ESS.

Table 2. Categorization of multi-functional ESS conflicts.

Conflict Name/Type	Description & Examples
Multi-objective optimization/ C_I	Two UCs are optimizing the functions $f(x)$ and $g(x)$ respectively. This entails a conflict when $f(x)$ and $g(x)$ need to be manipulated simultaneously. For instance, an EMS supports to the stabilization of the grid frequency (UC_3) and to the minimization of energy prices (UC_5). An over frequency event would require a reduction of the active power (P). In the meantime, UC_5 requires to increase P , then a coordination of control schemes is needed.
Maximal limit dependency of control variables/ C_{II}	Two UCs are controlling u_1 and u_2 respectively. Additionally, the limits of u_1 depend on the value of u_2 (i.e., $f(u_2) \leq u_1$). For instance, an EMS contributes to voltage regulation (UC_2) by setting the reactive power (Q). Besides of this the EMS receives set-points to deliver active power (P) in a context of UC_6 provision. The limits of P depend on Q according to $P < \sqrt{S_{max}^2 - Q^2}$. Hence, an alignment of services is suitable to avoid the saturation of P and Q .
Set-point out of limits/ C_{III}	This conflict occurs when at least two UCs control the same variable and the total set-point value exceeds the limits. For instance an EMS provides Primary Control Reserve (PCR) and market service (UC_1) by receiving set-points from EMO and TSO to control P . Even when each set-point respects the limits of P an overall violation of the set-points is possible.
Set-point sign conflicts/ C_{IV}	This conflict is based on the tracking of active power provision by monitoring the State of Charge (SOC) of the ESS. On this basis, when more than one service affects the value of P by two set-points with opposite signs, the tracking of single-services is lost. For instance, an EMS provides UC_6 and UC_4 , then the value of P is set. When the total set-point is zero, the SOC remains the same leading to a wrongly non-provision of the services conclusion.
Set-point set by at least two UCs/ C_V	At least two different use cases have the intention of controlling the same variable of a system.
Interrelated manipulated variables/ C_{VI}	A UC is controlling a variable u_1 to manipulate y_1 whereas a second UC is controlling u_2 to manipulate y_2 . Additionally one of the following statements is true: $y_1 = G_{11} \times u_1 + G_{12} \times u_2$ or $y_2 = G_{21} \times u_1 + G_{22} \times u_2$. It means that the manipulated variable y_1 or y_2 is affected by the first UC and the second one. For instance, an EMS provides voltage control (UC_2) by injecting/consuming reactive power (Q). In the meantime, the TSO requires to balance the active power (P) in a UC_3 context. As a consequence, voltage of the grid is regulated by UC_2 however it is also affected by UC_3 . An analysis of the whole multi-functional system (voltage control and PCR) is required.

3.2. Ontologies for Modeling ESS Applications

The previous conflicts categorization enables the identification of data involved in a conflict occurrence. This information is used as a basis to establish a data model for an automatic investigation of conflicts, hence an ontology aligned with the SGAM model and the use case templates suggested by the smart grid coordination group—sustainable processes in [28], is developed (i.e., EMS-ontology). Ontologies introduce the terms of classes and relations [20]. A class intends to classify information by categories and a relation is defined as a relation between classes. Following this, the services involved in a multi-use ESS are classified under the class High Level Use Case (*HLUC*), this concept describes a general requirement and is independent of technical specifications and technologies. A detailed description of the functionalities covered by a *HLUC* is defined under the class Primary Use Case (*PUC*). The EMS application is encapsulated under the class *Application*, the variables controlled by an application are defined by a *ControlVar* class. On the other hand, a relation can be intuitively interpreted by looking at the related classes. For instance, the role *hasHLUC* means that an *Application* has a *HLUC*, the role *optimize* is understood as a *PUC* regulates/optimizes a variable, etc.

An overview of the proposed classes is depicted in Figure 3. This figure also shows the whole EMS-ontology by a graph representation, where a node of the graph is defined as a class and an oriented arc as a relation. Classes are structured under two main concepts: active and passive device. An active device (e.g., EMS) embeds controllers and control a passive device (e.g., ESS). More complex relation between classes (e.g., inclusions, transitivity axioms) cannot be illustrated by a graph. Hence, a formal knowledge representation is employed.

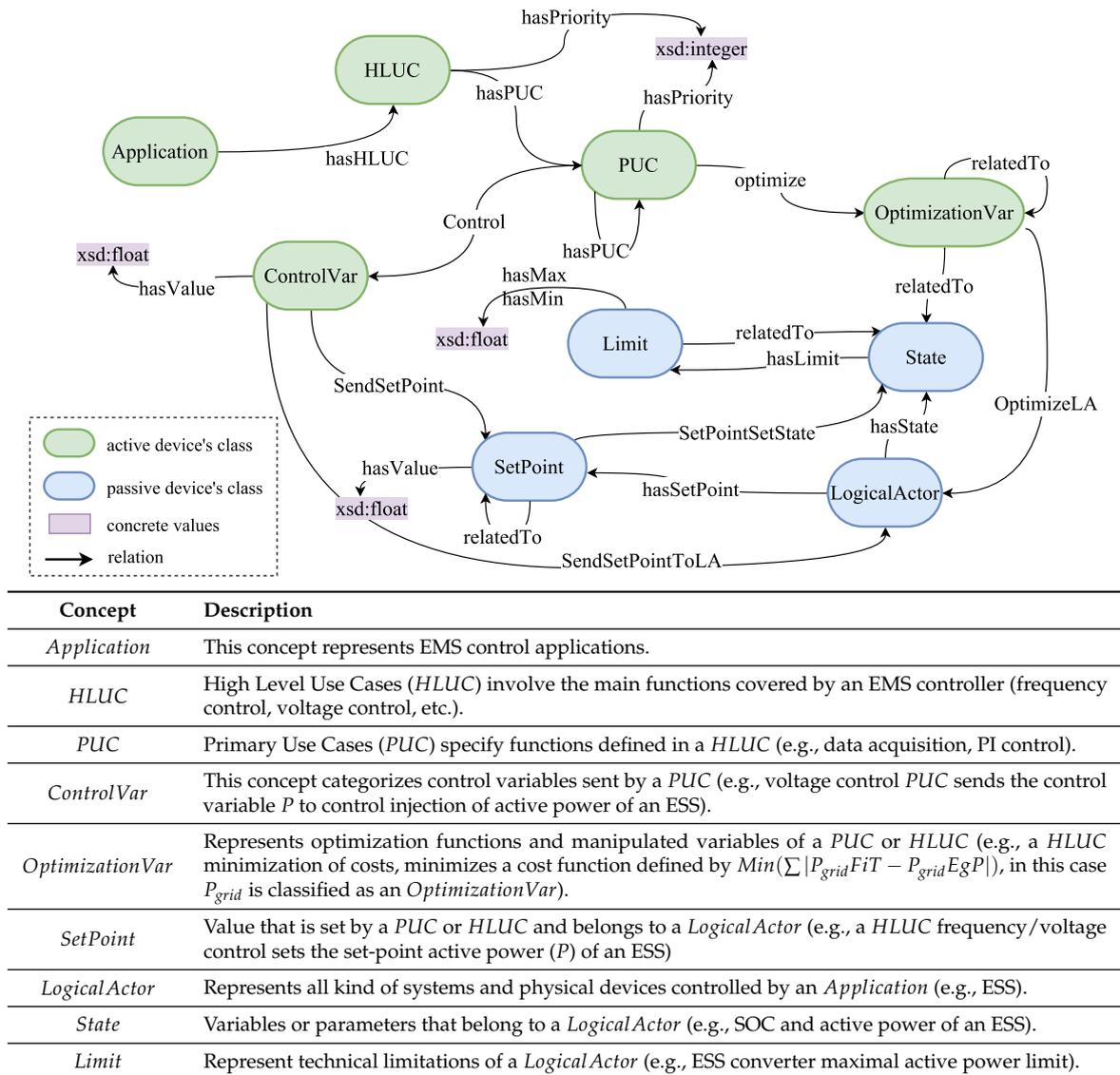


Figure 3. Overview of the proposed EMS-ontology for modeling multi-functional ESS applications.

3.3. Formal Representation of the EMS-Ontology

There exists a variety of languages for a formal knowledge representation of ontologies like Knowledge Interchangeable Format (KIF), Frame Logic (F-Logic), Description Logic (DL), etc. [29]. In this work, DL is more suitable for its expressiveness and rigor. Despite its reduced set of language constructors, DL provides a logic reasoning system, check of consistency and classification of data [23]. Thus, the previously introduced EMS-ontology is formally represented in DL notation.

DL introduces the terms *TBox* for terminological box and *ABox* for assertional box. In general the *TBox* defines concepts (i.e., classes) and roles (i.e., relations) as well as how roles and concepts are related to each other. The *ABox* establishes assertions matching individuals to concepts and roles. For instance the statement “An EMS provides two services frequency-watt and voltage control” belongs to the *ABox*, while the statement “An Application contains at least one HLUC” belongs to the *TBox*. Both assertions correspond to the proposed EMS-ontology, they are shown in Figure 4. Hence a syntax characterized by logical constructors (e.g., \sqsubseteq , \exists , $trans()$) is set [23]. DL notation assists the representation of more complex relations, for instance the constructor $trans(relatedTo)$ characterizes a transitivity of roles. It means, if *var1* is *relatedTo* *var2* and *var2* is *relatedTo* *var3* then *var1* is *relatedTo* *var3*. The complete representation of the EMS-ontology is shown in Appendix A.

$$TBox = \{Application \sqsubseteq Application \sqcap \exists hasHLUC.HLUC, trans(relatedTo), \dots$$

$$ABox = \{HLUC(frequencycontrol), HLUC(voltagecontrol), Application(EMS), \dots$$

Figure 4. Formal representation of the EMS-ontology (extract of the TBox and ABox.)

3.4. Methodology for Identifying Controller Conflicts

One main goal of the above introduced EMS-ontology is to detect the classified conflicts from Section 3.1. To this end, the methodology depicted in Figure 5 is used. This process consists on the extraction of information from an ESS domain expert to create a knowledge base (i.e., the ABox). Subsequently, this data together with the TBox of the EMS-ontology are analyzed by a reasoner engine. A reasoner performs consistency checking and inference mechanisms to generate additional facts from the existing knowledge base. Additionally, the inferred data is queried with the aim of detecting conflicts. To this effect, a set of queries targeting conflicts is proposed.

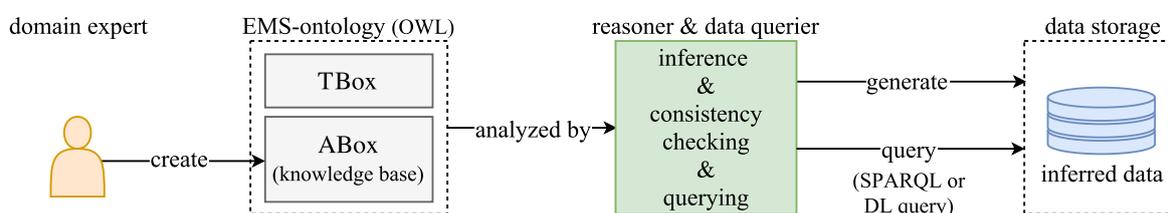


Figure 5. From knowledge base to inferred data.

The evaluation of the EMS-ontology is performed against a set of questions corresponding to the conflicts from Section 3.1. The formalization of those questions is done by stating DL queries using mathematical constructors [23]. This results in a set of appropriate DL queries to detect each conflict type as outlined in Table 3. They are executed in sequence, from this execution a report stating the identification of potential conflicts is provided. One of this DL queries is following described: The axiom $OptimizationVar \sqcap \geq 2Optimize^{-1}.PUC$ addresses variables that are optimized/regulated by at least two services (PUC). The other queries are intuitively apprehended by looking at the concepts and constructors meaning. In the case of Queries 3–4, a DL notation is not suitable due to the carrying out of arithmetic logic, then SPARQL queries are employed instead (see also Section 3.5).

Table 3. Definition of DL/SPARQL queries per conflict type.

Question/Conflict Type	DL/SPARQL Notation
Question 1: What are the variables that are optimized or regulated by at least two different services?/C _I	$OptimizationVar \sqcap \geq 2Optimize^{-1}.PUC$
Question 2: What are the states that are controlled by a use case application and its limits are in turn controlled by a second control application?/C _{II}	$Limit \sqcap \exists relatedTo. (State \sqcap \exists SetPoint SetState^{-1}. (SetPoint \sqcap \exists SendSetPoint^{-1}. (ControlVar \sqcap \exists Control^{-1}. PUC)))$
Question 3: Does a set of variables exist which intends to control a set-point of an ESS, causing a violation of technical limits imposed to this set-point?/C _{III}	supported with SPARQL (see Appendix C)
Question 4: Are there a set of control variables that control a determined set-point of an ESS and the multiplication of their values is negative?/C _{IV}	supported with SPARQL (see Appendix C)
Question 5: What are the variables that control a same set-point of an ESS?/C _V	$SetPoint \sqcap \geq 2SendSetPoint^{-1}. (ControlVar \sqcap \exists Control^{-1}. PUC)$
Question 6: What are the variables that are manipulated by a service and are affected by a second use case?/C _{VI}	$OptimizationVar \sqcap \exists optimize^{-1}. PUC \sqcap \exists relatedTo. (State \sqcap \exists control^{-1}. PUC)$

3.5. OWL and SPARQL to Evaluate the EMS-Ontology

The previous section defined a set of questions to evaluate the efficacy of the EMS-ontology. Some of those questions—*Questions 3–4*—need to carry out automatically a sequence of queries and to deal with concrete values as well. Hence, it is suitable to implement the proposed methodology for conflicts identification by employing Web Ontology Language (OWL) and SPARQL query language approach [20].

OWL is formally founded on description logic principles, it handles concrete properties defined in DL using datatypes (e.g., xsd:float). Moreover, OWL is part of the W3C standardized ontology languages. SPARQL a graph-based query language is also an official W3C recommendation. It extracts information from an OWL ontology and introduces a large list of functions for querying data enabling a higher flexibility compared to DL queries (e.g., *GROUP*, *ORDER BY*). Thus, SPARQL notation is used to formulate *Questions 3–4* (for details see Appendix C).

4. Handling of Conflicts within ESS Applications

The previous section classifies conflicts and presents a mechanism to detect them by the usage of ontologies. This section is focused on the handling of those conflicts, thus, potential solutions for conflict's resolution are presented. Afterwards, the implementation of those solutions is addressed by extending the EMS-ontology and by using a MDE-based approach to derive an EMS controller implementation.

4.1. Handling of Solutions per Conflict Type

A multi-functional ESS setup is illustrated in Figure 6. A set of n services ($Service_i$ for $i = 1, \dots, n$) is provided by the ESS equipped with an EMS controller. Each service is identified with its corresponding active and/or reactive power set-points P_i and/or Q_i . For some services, such as local grid voltage support, the set-point evaluation is performed locally. On the other hand, set-points can be externally identified—hereinafter called remotely—and transferred to the EMS controller (e.g., for an energy market associated service [12]).

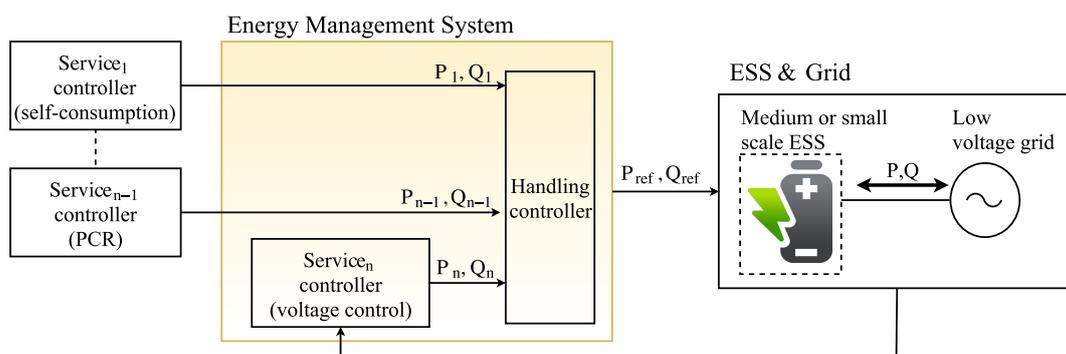


Figure 6. Multi-functional ESS equipped with an embedded EMS controller.

4.1.1. Set-Points Down-Regulation

To begin with, it is supposed that the set-point values P_i are identified and transmitted to the EMS controller. Furthermore, it is assumed that each signal is smaller than the ESS converter maximal active power limit P_{max} (i.e., $|P_i| < P_{max}$ for $i = 1, \dots, n$). As a result of independent set-point identification mechanisms, the total set-point P_{ref} can exceed the P_{max} value (i.e., $P_{ref} = \sum_{i=1}^n P_i > P_{max}$). For instance, a use case set-up is defined by a P_{max} that is limited to 100 kW and two services that intend to control the set-point P_{ref} by the values 40 kW and 80 kW respectively. Even if independently the services do not exceed the technical limitations (100 kW), the accumulation of them (40 kW + 80 kW) could do. Thus, the set-points must be down-regulated. To this end, for each

service a corresponding priority is defined. Assuming that the service n has the lowest priority if the maximal limit is exceeded, the set-point P_n is down-regulated as

$$P_{n-new} = \text{sgn}(P_n) \left| P_{max} - \sum_{i=1}^{n-1} |P_i| \right| \quad (1)$$

Similarly, if still the sum is greater than P_{max} , then P_n is set to zero and the next least-prior term is down-regulated. Hence, the corresponding services are partially provided. This handling solution corresponds to the conflict C_{III} .

4.1.2. Converter PQ Range Limiter

The handling solution proposed in the previous section makes it possible to define the reference ESS active and reactive power such that all the services are considered

$$P_{ref} = \sum_{i=1}^n P_{i-new}, \quad Q_{ref} = \sum_{i=1}^n Q_{i-new} \quad (2)$$

Consequently, the reference apparent power is defined by

$$S_{ref} = \sqrt{P_{ref}^2 + Q_{ref}^2} \quad (3)$$

This value is limited by the ESS apparent power limit (S_{max}). Although all the individual service set-points associated with active and/or reactive power provision are smaller than the maximal limits (i.e., $P_{i-new} < P_{max}$ and $Q_{i-new} < Q_{max}$ for $i = 1, \dots, n$) still the ESS apparent power limit can be violated i.e., $S_{ref} > S_{max}$ exemplifying a conflict C_{II} . Thus, the corresponding P_{ref} and Q_{ref} must be down-regulated such that $S_{ref} = S_{max}$. A potential solution is detailed in the literature [30].

4.1.3. SOC Estimation and Capacity Allocation

In order to illustrate type C_{IV} conflict occurrence in this application, let the service set-points be

$$P_i \neq 0 \quad \text{and} \quad P_j \neq 0 \quad \text{for} \quad i, j = 1, \dots, n \quad \text{such that for any} \quad (i, j), i \neq j \quad \text{and} \quad P_i \cdot P_j < 0 \quad (4)$$

This means that, the set-points P_i and P_j , targeting the active power value of an ESS, have opposite signs. This corresponds to the conflict C_{IV} . In such a conflicted scenario, it is important to virtually track the impact of each set-point. This is realized by virtually allocating a portion of the ESS capacity to each service. Furthermore, a state of charge estimator must be implemented for each capacity portion. State of charge estimator evaluates the SOC changes by considering the allocated capacity, battery voltage measurement and the power set-point defined by Equation (4). A detailed description of this handling solution is described in [30].

4.1.4. Use Case Specific Solutions

A general solution can not be encouraged for conflict C_{VI} . Hence, the handling solution must be employed depending on the use case specification/requirement. For instance, a multi-functional ESS provides services which define the P_{ref} . Additionally, the Point of Common Coupling (PCC) voltage support is also offered, see Figure 1. Voltage support has to be provided so that the PCC voltage V_{pcc} does not violate the limits (V_{upper} and V_{lower}) defined by the DSO. The steady state PCC voltage in terms of active/reactive power injected by the ESS is expressed by

$$V_{pcc} = V - \frac{RP_{ref} + j\omega LQ_{ref}}{V} \quad (5)$$

where the R, L are the approximated grid resistance and inductance and V is the grid voltage. Based on Equation (5) the active power provision affects the PCC voltage. Thus, the simultaneous provision of voltage support (i.e., $Q(V)$ control) and active power associated services are in conflict, this corresponds to a conflict C_{VI} . One solution to handle this conflict is based on a dynamic reactive power control [30]. In this approach, an integral controller with a proportional gain dynamically adjusts Q_{ref} such that the voltage always stays within its limits $V_{lower} \leq V_{pcc} \leq V_{upper}$.

A similar methodology suggesting dynamic regulation of PCC voltage by the means of both active and reactive power is proposed by [13]. Alternatively, the solution presented in [8] evaluates the compensating Q_{ref} according to a pre-set $Q - V$ characteristic curve defined by [4]. Comparing the aforementioned approaches shed lights on the fact that the choice of voltage regulation strategy mainly depends on the application requirements, network voltage level and ESS sizing.

4.2. Extended EMS-Ontology for Handling Conflicts

A main goal of the EMS-ontology, besides conflicts identification, is the handling of them. To this end, an extension of this ontology is required. This extension is based on the definition of patterns targeting the previously addressed handling solutions. A pattern is easily identifiable for the cases where a general solution is provided (e.g., “Converter PQ Range Limiter” as outlined above). Moreover, the information gathered from the current EMS-ontology is enough to identify and propose a corresponding general solution. However, in case of specific solutions the information modeled with the current ontology is not enough to precise a conflict resolution. It means extra information from the multi-functional ESS context is required (ESS capacity, network operator requirements, etc.). This paper proposes an extension of the actual EMS-ontology to implement the mentioned specific/general handling solutions. For simplicity, only the solution “Down-Regulation of Set-points” is discussed in detail in this article.

A graph illustrating new concepts and roles for implementing “Down-Regulation of Set-points” is shown in Figure 7. A validation of this ontology is based on the collection of enough information to carried out according to Equation (1). To this end, information provided by the EMS-ontology such as priority of services, physical device to be controlled (e.g., ESS), set-point values out of limit (e.g., $\sum_{i=1}^n P_i > P_{max}$, for $i = 1, \dots, n$) and set-point to be affected (P_{ref}) are evaluated. As a result, set-points are scaled-down in compliance with the maximal converter active power (P_{max}). An evaluation of the extended ontology is done by an use case implementation in Section 5.3.

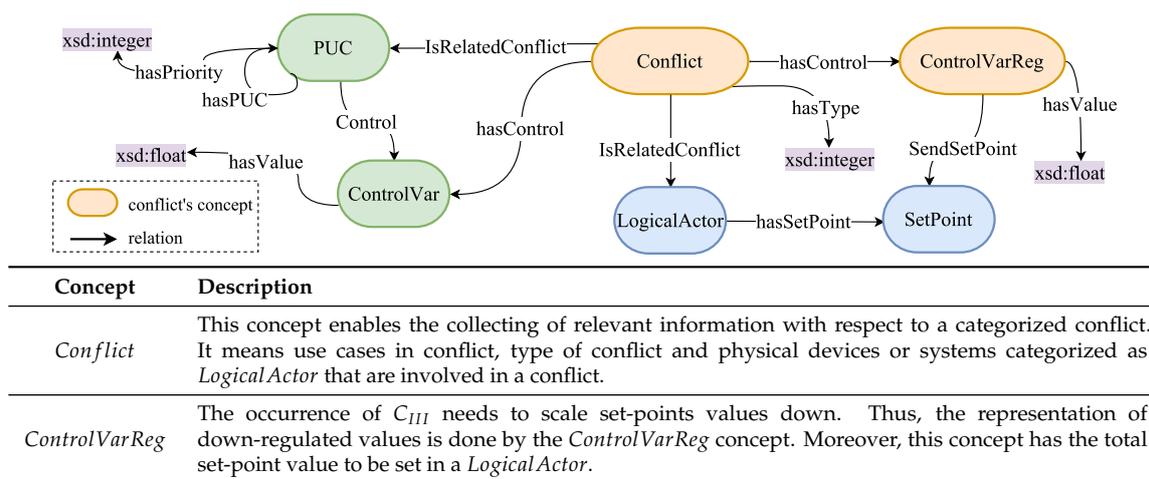


Figure 7. Extension of the EMS-ontology for the handling of controller conflicts.

4.3. EMS-Ontology Exploited by the MDE Approach

The MDE approach is focused on the modeling of applications. Once the model is accomplished the conduct of M2M and M2T takes place. To this end, a meta-model and a source model are required. A way to apply ontology engineering to the MDE approach is discussed in [31]. In this work, a meta-model is considered as an ontology (*TBox*) and a source model as an instance of the ontology (*ABox*). Previous sections described an EMS-ontology to detect and manage conflicts. The fact of seeing EMS-ontology as a meta-model and a multi-functional ESS use case as a source model brings the benefit of generating exploitable source code (e.g., C++, Matlab) and target models (e.g., Simulink models). The resulted model or text is meant to be deployed during the elaboration and validation of the ESS application. This attempts to demonstrate that the EMS-ontology supports during the rapid prototyping of ESS control applications. A proof-of-concept is addressed in the following section.

5. Proof of Concept

In this section a practical multi-functional ESS application is introduced. This example aims to highlight the conflicts occurrence and propose corresponding handling solutions by means of an ontology-based methodology support. Thus, a Smart Low Voltage Grid Controller (SLVGC) is implemented which integrates the handling solutions presented in Section 4. In the following, the application setup and ESS functionalities/services are described. In this control solution the ESS is required to provide self-consumption and market services in smart low voltage grid. Furthermore, from the power quality perspective the ESS has to provide local voltage support. In the following each service and the setup configuration are described.

5.1. ESS Services and Setup

In this setup, the ESS helps the household Photovoltaic (PV) systems in balancing the generation and consumption profiles. In other words, every household rents a portion of the storage capacity in order to store the excess PV generated power and consume it in high-load time intervals. The amount of power to be charged/discharged is equal to the generation-consumption difference and it is calculated by the smart metering devices installed at the household PCCs. Accordingly, these measured values are steadily sent to the SLVGC and all together form the total self-consumption set-point P_{sc} . The individual self-consumption signals and the sum P_{sc} are updated every minute. From the SLVGC point of view, P_{sc} is considered as an externally defined set-point. In this application, self-consumption is provided to 10 households renting 1/2 of the total ESS capacity. The set-point for this service P_{ms} is identified by the EMO. Hence, the storage owner can sell the stored energy in high-price time periods, and recharge the market allocated capacity in a low-price period. This service also rents 1/2 of the total ESS capacity. As a result of the resistive nature of the low voltage grid and multiple service provision, the PCC voltage may exceed the permitted limits. Hence, PCC voltage is regulated by means reactive power injection-absorption (Q_{gs}). PCC voltage controller is one of the SLVGC integrated schemes. Thus, the grid support control variables Q_{ref} is locally evaluated.

The setup configuration is demonstrated in Figure 8. The input signals P_{sc} and P_{ms} are sent to the SLVGC hardware via the communication layer (e.g., utilizing a Modbus TCP protocol). The SLVGC also receives measurements from battery (SOC_{bat} and V_{bat}), injected power by ESS (P and Q), and the PCC voltage (V_{pcc}).

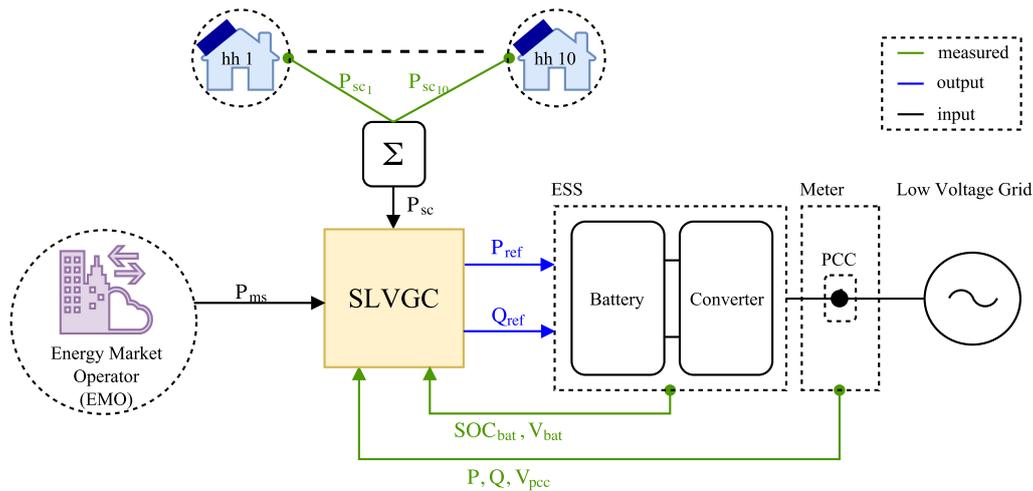


Figure 8. SLVGC control approach for a multi-functional usage of ESS.

5.2. Applying the Ontology-Based Methodology

An engineering support consisting of the above introduced ontology-based methodology for detecting controller conflicts and for handling corresponding solutions is in the following example applied and demonstrated. The detection of the overlapping of ESS services is based on the analysis of data provided by domain experts (e.g., control or power system engineer). This data defined as knowledge base of the SLVGC application (*ABox*) is evaluated by reasoning mechanism resulting in the derivation of additional truths (inferred data). As a sequel, inferred data is queried to identify conflicts. This process continues with the creation of a new instance of the EMS-ontology to manage conflicts (updated data). The whole cycle is shown in Figure 9.

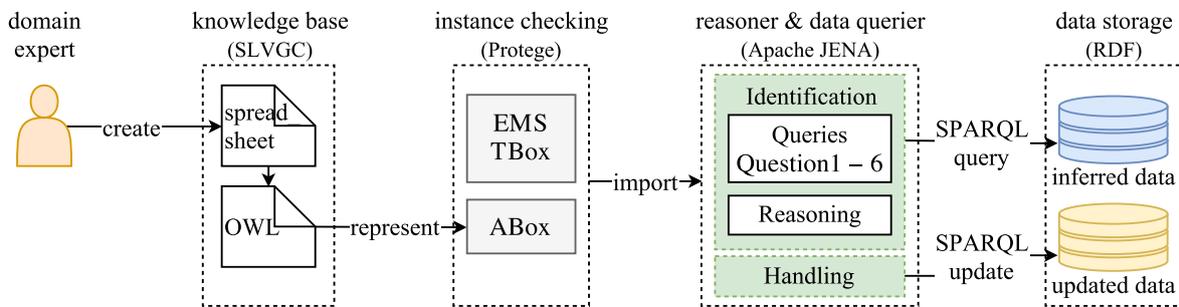


Figure 9. Ontology-based methodology applied to a selected use case example.

Domain specialists own the expertise about the services to be implemented within the SLVGC controller. That is why their knowledge related to the multi-functional usage of ESS need to be collected and analyzed. To achieve a comprehensive data collection, templates for spreadsheets are designed where relevant domain knowledge is stored in tabular and therefore understandable form. However data in that format is not ready to be analyzed, then a transformation from spreadsheets into an instance of OWL EMS-ontology is required. The knowledge resulted from this transformation is shown in Figure 10.

$$\begin{aligned}
 ABox = \{ & Application(SLVGC), HLUC(MarketService), HLUC(SelfConsumption), HLUC(VoltageControl), \\
 & OptimizationVar(V_{pcc}), ControlVar(P_{ms}), ControlVar(P_{sc}), ControlVar(Q_{gs}), \\
 & SetPoint(P_{ref}), SetPoint(Q_{ref}), LogicalActor(ESS), LogicalActor(Meter), \dots
 \end{aligned}$$

Figure 10. Extract of SLVGC knowledge base.

It shows that *SLVGC* is defined as an instance of the concept *Application*, besides of this the concept *HLUC* models the services provided to the EMO, the customers, and the DSO. The *ControlVar* concept gathers control variables. The *SLVGC* implements local voltage control, then the regulation of the voltage at the PCC point V_{pcc} is modeled by the concept *OptimizationVar*. The ESS and the meter that carries information of the low voltage grid are defined by the *LogicalActor* concept. The ESS device incorporates two set-points to set active and reactive power by external systems (i.e., P_{ref} , Q_{ref}). Those values are defined as *SetPoint* concepts. The exhaustive knowledge of the *SLVGC* application is presented in Appendix B.

As a first attempt to validate the EMS-ontology and the constancy of the *SLVGC* knowledge (i.e., *ABox*), the Protege toolkit—an open-source ontology editor—is used. This tool enables the integration of a variety of reasoners (Pellet, FaCT++, HermiT, etc.) and the evaluation of DL and SPARQL queries through the setup of plug-ins. Thus, Protege is convenient for validation of the EMS-ontology and consistency checking of the *SLVGC* knowledge base. However, it is limited in terms of implementation of extensive queries, it is the case for *Question 3–4* (see Table 3). To cover those gaps, JENA—a Java framework [32]—is employed for the implementation of queries as presented in Table 3. It supports the deployment of OWL ontology language, the connection to inference engines and the usage of a SPARQL processor. This query engine performs operations to create, update and remove data from a Resource Description Framework RDF store through SPARQL update [33], as depicted in Figure 9.

5.3. Exploiting Inferred Data

A benefit of an ontology-based application is the derivation of additional truths from the knowledge base. In this context, one assertion derived from *SLVGC* knowledge base (see *ABox* in Figure 10) is that *VoltageControl* service affects the active power value of the grid (P). This deduction is entailed from the axiom $Optimize \circ relatedTo \sqsubseteq Optimize$ and the statements $Optimize(VoltageControl, V_{pcc}), relatedTo(V_{pcc}, P)$. Many others deductions are automated inferred by the engine reasoner. The inferred data is queried by SPARQL and DL queries to achieve the detection of conflicts type. The SPARQL and DL notation used to establish the aforementioned queries is fully presented in Appendix C. Conflicts resolution mechanisms are employed according to the study presented in Section 4. This is shown in Table 4, moreover answers derived from queries are also exposed.

Handling solutions addressed in Table 4 are satisfactory implemented. The solution for conflict C_{III} is illustrated in Figure 11. Implementations of calculations introduced in Equation (1) are carried out in the JENA framework. The resulting values are used to create an instance of the EMS-ontology (i.e., *SLVGC_conflict*). This instance gathers active power values coming from market and self-consumption services (P_{ms} , P_{sc}). When the sum of those values exceeds the technical limitations of the ESS ($P_{max} = 100kW$) they need to be down-regulated. A way to exploit the instance *SLVGC_conflict* is by carrying out M2T transformations, resulting in the automatic generation of source code (e.g., Matlab code). This code is deployed into a power system simulator (e.g., Simulink) to execute off-line simulations of the *SLVGC* controller. A similar approach is employed for the other handling solutions mentioned in Table 4. Hence, the reduction of manual work during the controllers development process is achieved.

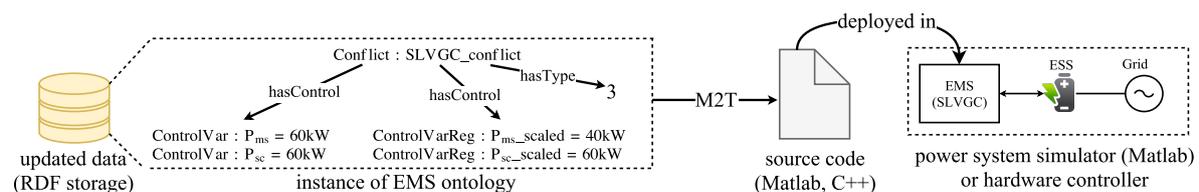


Figure 11. Data for handling conflict C_{III} and the corresponding M2T process.

Table 4. Querying the SLVGC ontology and handling solutions per conflict type.

Conflict/Type	Detected	Conclusion Derived from Queries	Handling Solution
Multi-objective optimization/ C_I	X	There is not any variable that is intended to be optimized or regulated by two different use cases then conflict C_I is not identified. Control strategies regarding self-consumption and market services are run externally, thus the SLVGC application receives set-points from household and EMO. It could be the case that those services are in conflict but as the SLVGC application has no further information regarding those services then C_I is dismissed.	Not required
Maximal limit dependency of control variables/ C_{II}	✓	The limit of Q is defined by the state $P(\sqrt{S_{max}^2 - P^2})$. This state P is controlled by the use cases self-consumption and market service through the control variables P_{sc} and P_{ms} . Additionally, the state Q is controlled by voltage control use case through Q_{gs} . A conflict C_{II} involving the control variables $\{P_{sc}, Q_{gs}\}$ and $\{P_{ms}, Q_{gs}\}$ is detected. Then a coordination between the services in conflict is required to avoid a saturation of P and Q .	Converter PQ range limiter
Set-point out of limits/ C_{III}	✓	The set-point P_{ref} is set by two control variables: P_{ms} and P_{sc} . The total value to be set exceeds the active power limit P_{max} , then a conflict C_{III} is identified	Down-regulation of set-points
Set-point sign conflicts/ C_{IV}	X	The values of control variables P_{ms} and P_{sc} have the same sign then a conflict C_{IV} is dismissed. However those values evolve over time, thus the EMS-ontology cannot predict the conflict C_{IV} . Identification of conflict C_{IV} should take place during real-time operation or simulation of the SLVGC application.	SOC estimation and capacity allocation
Set-point set by at least two use cases/ C_V	✓	The controllers market service and self-consumption have the intention of setting the set-point P_{ref} of the ESS, then conflict C_V is detected. This conflict is not considered harmful by the domain expert. Thus, no handling solutions are executed.	Not required
Interrelated manipulated variables/ C_{VI}	✓	The use cases market service and self-consumption control the state P . Additionally the value of this state affects the PCC voltage ($V_{pcc} = V - \frac{RP+j\omega LQ}{V}$). Thus, V_{pcc} is affected by market service and self-consumption use cases. On the other hand, V_{pcc} is manipulated by voltage control to regulate the PCC voltage, then C_{VI} is identified.	Reactive power voltage controller

5.4. Simulation Result

In this section the simulation results of the multi-functional ESS application introduced above is presented. The ESS and distribution feeder are modeled based on the approach suggested by [30,34,35]. The simulation is performed in Matlab/Simulink. The total simulation time is set to 2 h and the model parameters are described in Appendix D. As previously explained above, the SLVGC receives self-consumption and market service set-points, P_{sc} and P_{ms} respectively. The implemented SLVGC integrates the handling solutions introduced in Section 4.

The achieved results are depicted in Figure 12. In this illustration, the power, voltage and SOC values are respectively normalized with S_{max} , V and 100% (see Appendix D). Furthermore, the measurement reference frame is chosen such that positive/negative power values correspond to charge/discharge operation. According to Equation (5), charge/discharge state results in PCC voltage drop/increase. In Figure 12, the two plots (I) and (II) are associated self-consumption and market service provision. In these plots the original set-points are shown in blue (i.e., P_{sc} and P_{ms}). The time resolution for P_{sc} and P_{ms} are 1 and 15 min respectively. P_{sc} exhibits a period of high PV generation followed by an oscillating profile caused by a sunny-cloudy weather condition. For the sake of compactness, the pure consumption state ($P_{sc} < 0$) is neglected. However, the original P_{ms} shown in plot (II) requires both charge and discharge operations. In these plots, the green signals correspond to the down-regulated signals if the ESS P_{max} limit is exceeded (see Section 4.1.1). In this application, P_{sc} has higher priority, thus the down-regulation is applied to P_{ms} , resulting in

$$P_{sc-new} = P_{sc}, \quad P_{ms-new} = \text{sgn}(P_{ms}) \left| P_{max} - |P_{sc}| \right| \quad (6)$$

This explains the intervals in plot (II) where the green signal does not follow the blue signal. The regulation of the voltage, active and reactive power are shown in Figure 13 by means of plots (III) and (IV). In plot (III), the active, reactive and apparent power profiles are shown. Based on Equations (2) and (3) the ESS set-points are defined. Following the argument presented in Section 4.1.2, if the S_{max}

limit is violated the ESS apparent S_{ref} is limited. This results in down-regulating P_{ref} and Q_{ref} accordingly, hence

$$S_{ref-lim} = S_{max}, \quad P_{ref-lim} = \left(\frac{S_{max}}{S_{ref}}\right)P_{ref}, \quad Q_{ref-lim} = \left(\frac{S_{max}}{S_{ref}}\right)Q_{ref} \quad (7)$$

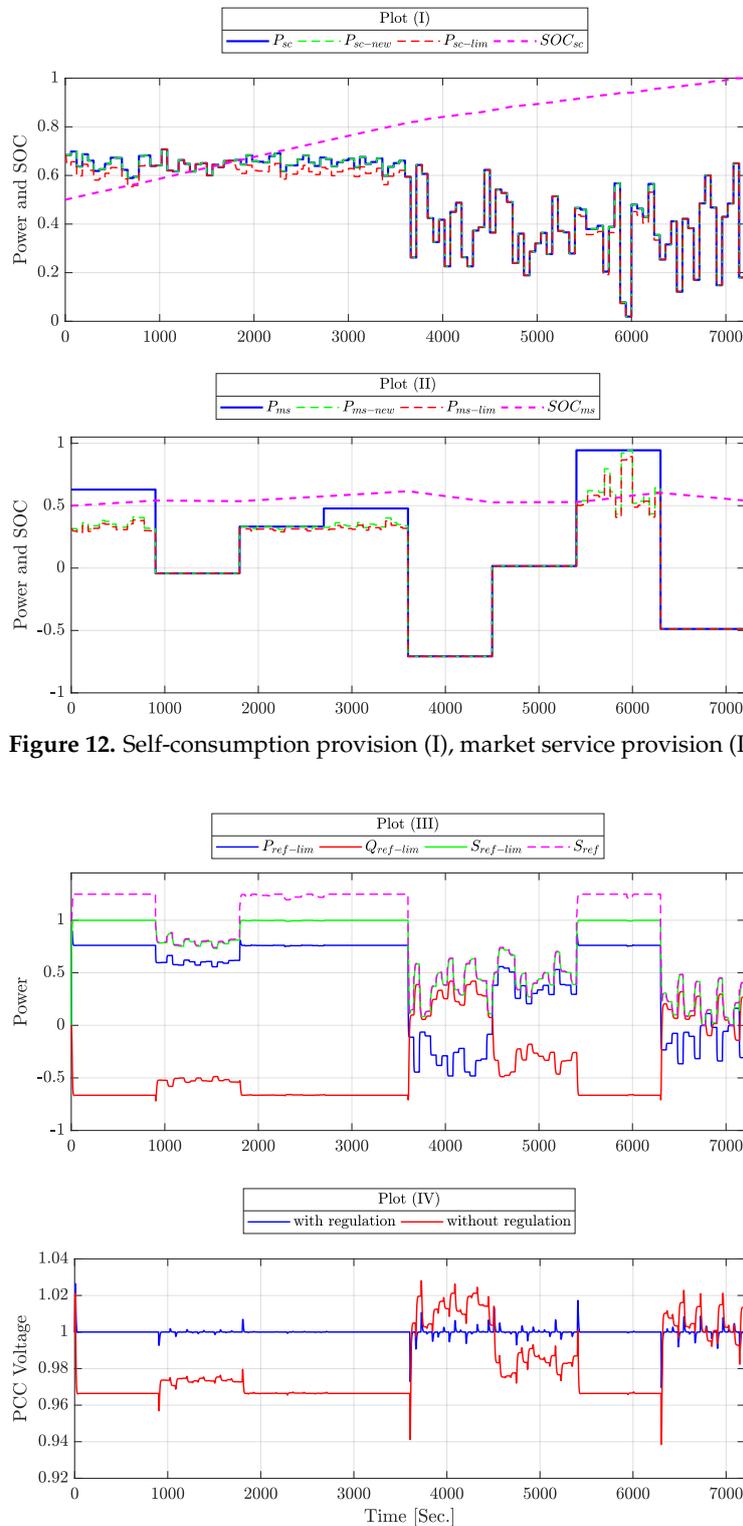


Figure 12. Self-consumption provision (I), market service provision (II).

Figure 13. ESS power profiles (III), PCC voltage control (IV).

This corresponds to the intervals where the green signal is limited to 1 in plot (III). As a consequence of down-regulating P_{ref} to $P_{ref-lim}$, the associated market and self-consumption signals are also scaled down with the same ratio as in Equation (7), resulting in P_{sc-lim} and P_{ms-lim} . These signals are shown in red in plots (I) and (II). Sequentially, these final services set-points are feedbacked to state of charge estimation schemes mentioned in Section 4.1.3. The estimated state of charge evolution is depicted in plots (I) and (II) reflecting the availability of allocated capacity for each service. Last but not least, the reactive power set-point Q_{ref} is evaluated by the PCC voltage controller which is implemented based on the strategy in [30]. By construction, Q_{ref} always has an opposite sign compared to P_{ref} . The PCC voltage controller impact can be observed in plot (IV). For instance, when $P_{ref} > 0$ the ESS is charging. Thus, the PCC voltage drops from its nominal value. Consequently, the controller reacts by adjusting $Q_{ref} < 0$ such that PCC voltage regains the nominal value as it's steady state.

6. Conclusions and Future Work

With the large-scale integration DER into power distribution grids new services provided by ESS are required, thus the development of a corresponding multi-functional usage of them is now becoming a reality. This trend faces a difficult issue derived from the overlapping between control targets. Thus, an engineering support for an early detection and handling of conflicts during the requirements analysis of control structure is necessary. This methodology will enable a correct planning and management of ESS control applications.

The main basis of this solution lays on the definition of an ontology (i.e., EMS-ontology), overly simplified to identify and resolve conflicts. The presented EMS-ontology based approach demonstrates that an early requirements analysis of the control structure enables the identification and handling of conflicts. Moreover, a convenient and user-friendly way to gather knowledge from domain experts is also tackled by proposing a spreadsheet template. This knowledge is modeled by means of ontologies, hence reasoning mechanisms enable the inference of new facts. This resulted information is queried by predefined queries targeting conflicts classified by types (see Tables 2 and 3). Additionally, a set of handling solutions per conflicts type is encouraged, this motivates a semi-automatic generation of code to support conflicts resolution and to be implemented during the development of the system. This enables the rapid prototyping of ESS applications and a preventive correction of handling of conflicts before any practical implementation.

Model-based strategies to design the requirements of power systems are addressed by different standards and approaches. However, information gathered by them are not enough for a full conflict identification. A well-known approach to specify and structure the requirements of smart grid applications is defined by SGAM. This approach is being used in divers smart grids projects [26]. Despite of this, a full conflict detection is not totally ensured [11]. Additionally, the use case methodology introduced by IEC 62559 [10] gives the basis to specify requirements for energy systems control applications, but the identification of conflicts is not really addressed. In contrast with those methodologies, the engineering support approach defined by this paper does not provide a model for designing all the aspects of power systems applications. This motivates to concentrate future work on the alignment of the proposed approach with power system modeling methodologies such as SGAM and IEC 62559. This would improve the development chain of control applications, enabling a full description of power system application as well as an automatic conflict identification from the very beginning of the development process. Furthermore, the support of interoperability between the EMS with external systems such as TSO and EMO to import knowledge about controllers implemented remotely to the EMS should be assured. This entails the alignment of the EMS-ontology with other smart grid information models such as IEC 61850 and Common Information Model (IEC 61970), both are standards to improve the interoperability in electric power systems [36].

The literature shows only first preliminary ideas that addresses conflicts within a multi-use ESS context. An approach that targets the identification of conflicts within electric energy systems is

studied in [37], this study recommends the use of the language Multi-level Flow Modeling (MFM) to model the requirements of control structures. Additionally, this approach is aligned to the IEC 62559 in the scope of the ELECTRA IRP project [38]. However, because of the large range of power system applications covered by this solution, a correct effectiveness and also its limitations are unknown. Compared with this approach, not only the identification but the handling of conflicts are addressed by the EMS-ontology. Moreover, this ontology targets a restricted domain: multi-use ESS, then the objective design criteria is very well defined by a set of selected questions, as outlined in Table 3. On the other hand, a correct design of ontologies depends on the initial knowledge base, the EMS-ontology is built from use cases presented in Table 1, hence is not exhaustive for all the conflicts derived from multi-functional storage systems. Thus an efficient EMS-ontology would require the knowledge of a larger list of use cases. This may entail the extension of handling solutions resulting in an evolution of the ontology as well. Hence, the designing of the EMS-ontology is an ongoing and non-exhaustive process.

Acknowledgments: This work is partly supported by the Austrian Ministry for Transport, Innovation and Technology (bmvit) and the Austrian Research Promotion Agency (FFG) under the ICT of the Future Programme in the OpenNES project (FFG No. 845632).

Author Contributions: All the authors contributed to the main idea of the paper. Claudia Zanabria and Tayyebi Ali wrote the paper. Thomas I. Strasser supervised the overall work. Filip Prösl Andrén, Johannes Kathan, and Thomas I. Strasser proofread the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

Acronyms

CHIL	Control-Hardware-in-the-Loop
DER	Distributed Energy Resources
DL	Description Logic
DSO	Distribution System Operator
EERA	European Energy Research Alliance
EMO	Energy Market Operator
EMS	Energy Management System
ESS	Energy Storage System
HLUC	High Level Use Case
IEC	International Electrotechnical Commission
IRP	International Reporting Project
M2M	Model-to-Model
M2T	Model-to-Text
MDD	Model-Driven Design
MDE	Model Driven Engineering
OWL	Web Ontology Language
PCC	Point of Common Coupling
PUC	Primary Use Case
PV	Photovoltaic
RDF	Resource Description Framework
SCADA	Supervisory Control and Data Acquisition
SGAM	Smart Grid Architecture Model
SLVGC	Smart Low Voltage Controller
SOC	State-of-charge
TSO	Transmission System Operator
UC	Use Case

Nomenclature

P_i	Active power set by the service i
Q_i	Reactive power set by the service i
S_{ref}	Apparent power set-point of a battery inverter
P_{ref}	Active power set-point of a battery inverter

Q_{ref}	Reactive power set-point of a battery inverter
Q_{max}	Maximum reactive power limitation of a battery inverter
P_{max}	Maximum active power limitation of a battery inverter
S_{max}	Maximum apparent power limitation of a battery inverter
P_{i_new}	Down-regulated value of P adopted by the service i
Q_{i_new}	Down-regulated value of Q adopted by the service i
V	Voltage of the grid
P	Active power injected/consumed at the PCC node
Q	Reactive power injected/consumed at the PCC node
V_{pcc}	Voltage at the point of common coupling
R	Grid resistance
L	Grid inductance
P_{sc}	Active power set by the self-consumption service
Q_{gs}	Reactive power set by the grid support controller
P_{ms}	Active power set by the market service
P_{sc_new}	Down-regulated value of P_{sc} according to Set-point Down-Regulation strategy
P_{ms_new}	Down-regulated value of P_{ms} according to Set-point Down-Regulation strategy
P_{ms_limit}	Down-regulated value of P_{ms} according to Converter PQ Range Limiter strategy
P_{sc_limit}	Down-regulated value of P_{sc} according to Converter PQ Range Limiter strategy
S_{ref_limit}	Down-regulated value of S_{ref} according to Converter PQ Range Limiter strategy
P_{ref_limit}	Down-regulated value of P_{ref} according to Converter PQ Range Limiter strategy
Q_{ref_limit}	Down-regulated value of Q_{ref} according to Converter PQ Range Limiter strategy
SOC_{sc}	State-of-charge of the self-consumption service
SOC_{ms}	State-of-charge of the market service

Appendix A. TBox of the EMS-Ontology

$$\begin{aligned}
TBox = \{ & Optimize \circ relatedTo \sqsubseteq Optimize \\
& relatedTo \circ hasState^- \sqsubseteq OptimizeLA \\
& SendSetPoint \circ relatedTo \sqsubseteq SendSetPoint \\
& SendSetPoint \circ hasSetPoint^- \sqsubseteq SendSetPointToLA \\
& Control \circ SendSetPoint \circ SetPointSetState \sqsubseteq Control \\
& hasSetPoint^{-1} \circ hasState \circ hasLimit \sqsubseteq hasLimit \\
ControlVar \sqsubseteq & ControlVar \sqcap \forall SendSetPoint.SetPoint \sqcap \forall SendSetPointToLA.LogicalActor \\
& hasValue \text{ keyfor } (ControlVar \sqcup SetPoint) \\
& hasPriority \text{ keyfor } (HLUC \sqcup PUC) \\
& (hasMax \sqcup hasMin) \text{ keyfor } Limit \\
PUC \sqsubseteq & PUC \sqcap \forall Optimize.OptimizationVar \sqcap \forall hasSetPoint.SetPoint \\
& HLUC \sqsubseteq HLUC \sqcap \exists hasPUC.PUC \\
Application \sqsubseteq & Application \sqcap \exists hasHLUC.HLUC \\
SetPoint \sqsubseteq & SetPoint \sqcap \exists hasLimit.Limit \sqcap \exists SetPointSetState.State \\
& State \sqsubseteq State \sqcap \exists hasLimit.Limit \\
& Limit \sqsubseteq Limit \sqcap \forall relatedTo.State \\
OptimizationVar \sqsubseteq & OptimizationVar \sqcap \exists OptimizeLA.LogicalActor \\
LogicalActor \sqsubseteq & LogicalActor \sqcap \forall hasSetPoint.SetPoint \sqcap \exists hasState.State \\
& trans(relatedTo), Sym(relatedTo) \}
\end{aligned}$$

Appendix B. ABox of the SLVGC Application

$$\begin{aligned}
 ABox = \{ & Application(SLGVC), HLUC(MarketService), HLUC(SelfConsumption), HLUC(VoltageControl), \\
 & PUC(MarketService), PUC(SelfConsumption), PUC(VoltageControl), \\
 & OptimizationVar(V_{pcc}), ControlVar(P_{ms}), ControlVar(P_{sc}), ControlVar(Q_{gs}), \\
 & SetPoint(P_{ref}), SetPoint(Q_{ref}), LogicalActor(ESS), LogicalActor(Meter), \\
 & Limit(P_{max}), Limit(Q_{max}), hasLimit(P, P_{max}), hasLimit(Q, Q_{max}), State(P), State(Q), State(V_{pcc}), \\
 & hasHLUC(CEMS, MarketService), hasHLUC(CEMS, SelfConsumption), hasHLUC(CEMS, VoltageControl) \\
 & hasPUC(MarketService, MarketService), hasPUC(VoltageControl, VoltageControl), \\
 & hasPUC(SelfConsumption, SelfConsumption), \\
 & Control(MarketService, P_{ms}), Control(SelfConsumption, P_{sc}), Control(VoltageControl, Q_{gs}), \\
 & SendSetPoint(P_{ms}, P_{ref}), SendSetPoint(P_{sc}, P_{ref}), SendSetPoint(Q_{gs}, Q_{ref}), \\
 & relatedTo(V_{pcc}, P), relatedTo(V_{pcc}, Q), relatedTo(P_{max}, Q), relatedTo(Q_{max}, P), \\
 & hasState(ESS, P), hasState(ESS, Q), hasState(Meter, P), hasState(Meter, Q), \\
 & hasState(Meter, V_{pcc}), hasSetPoint(ESS, P_{ref}), hasSetPoint(ESS, Q_{ref}), \\
 & SetPointSetState(P_{ref}, P), SetPointSetState(Q_{ref}, Q), Optimize(VoltageControl, V_{pcc}) \\
 & hasMax(P_{max}, 100), hasMax(Q_{max}, 100), hasMin(P_{min}, -100), hasMin(Q_{min}, -100), \\
 & haspriority(MarketService, 2), haspriority(SelfConsumption, 3), haspriority(VoltageControl, 1), \\
 & hasValue(P_{ms}, 60), hasValue(P_{sc}, 60), hasValue(Q_{gs}, 60), \}
 \end{aligned}$$

Appendix C. Querying the Ontology of the SLVGC Controller

Table A1. Implementation of Question 1.

Multi-Objective Optimization (Conflict C_I)	
DL Query	Query Result
(1) Query to know if there is an optimization variable that is optimized by at least two different use case: <i>OptimizationVar</i> and (<i>InvOptimize min 2 (PUC)</i>)	{}

Table A2. Implementation of Question 2.

Maximal Limit Dependency of Control Variables (Conflict C_{II})	
DL Query	Query Result
(1) Search limits of states that depends on other states: <i>Limit</i> and <i>relatedTo min 1 State</i>	{ P_{max}, Q_{max} }
(2) Search state that owns the limit Q_{max} : <i>State</i> and <i>hasLimit value Qlimit</i>	{ Q }
(3) Search control variables that control the state Q : <i>ControlVar</i> and <i>Control value Q</i>	{ Q_{gs} }
(4) Investigate the PUC that controls the state Q : <i>PUC</i> and <i>Control value Q</i>	{ <i>VoltageControl</i> }
(5) Search state that is related to the limit Q_{max} : <i>State</i> and <i>relatedTo value Qmax</i>	{ P }
(6) Search control variables that control the state P : <i>ControlVar</i> and <i>Control value P</i>	{ P_{sc}, P_{ms} }
(7) Search PUC that controls the state P : <i>PUC</i> and <i>Control value P</i>	{ <i>MarketService, SelfConsumption</i> }

Table A3. Implementation of Question 3 and 4.

Set-Point out of Limits (Conflict C_{III}) and Set-Point Sign Conflicts (Conflict C_{IV})	
SPARQL Query	Query Result
(1) Search a set-point that is set by at least two control variables: <i>SELECT DISTINCT ?setpoint ?ControlVar WHERE {?setpoint onto:InvSendSetPoint ?ControlVar.}</i>	$\{P_{ms}, P_{ref}$ P_{sc}, P_{ref} $Q_{gs}, Q_{ref}\}$
(2) Investigate the control variables that sets P_{ref} : <i>SELECT DISTINCT ?controlvar WHERE {?controlvar onto:SendSetPoint ?value.} FILTER (?value=Pset)}</i>	$\{P_{ms}, P_{sc}\}$
(3) Get the value of control variables that set the set-point P_{ref} : <i>SELECT DISTINCT ?controlvar ?controlvalue WHERE { ?controlvar onto:SendSetPoint ?setpoint ?controlvar onto:hasValue ?controlvalue. FILTER (?setpoint=st:Pset)}</i>	$\{P_{ms}, 60$ $P_{sc}, 60\}$
(4) Get the sum of values of control variables P_{sc} and P_{ms} : <i>SELECT (SUM(?controlvalue) as ?TotalSetpoint) WHERE {?controlvar onto:SendSetPoint ?setpoint.?controlvar onto:hasValue ?controlvalue. FILTER (?setpoint=st:Pset)} GROUP BY ?setpoint</i>	{120}
(5) Find the limitations of the set-point P_{ref} : <i>SELECT ?Max WHERE {?setpoint onto:hasLimit ?limit.?limit onto:hasMax ?Max. FILTER (?setpoint=st:Pset)}</i>	{100}

Table A4. Implementation of Question 5.

Set-Point Set by at Least Two Use Cases (Conflict C_V)	
DL Query	Query Result
(1) Search set-point that is set by at least two control variables: <i>SetPoint and InvSendSetPoint min 2 ControlVar</i>	$\{P_{ref}\}$
(2) Identify the control variables that set the SetPoint P_{ref} : <i>ControlVar and SendSetPoint value Pset</i>	$\{P_{ms}, P_{sc}\}$
(3) Search the PUC that controls P_{ms} and P_{sc} : <i>PUC and Control value Pms, PUC and Control value Psc</i>	$\{SelfConsumption, MarketService\}$

Table A5. Implementation of Question 6.

Interrelated Manipulated Variables (Conflict C_{VI})	
DL Query	Query Result
(1) Search if there are some variables to be regulated or optimized: <i>OptimizationVar and InvOptimize min 1 PUC</i>	$\{V_{pcc}\}$
(2) Search the PUC that manipulates V_{pcc} : <i>PUC and Optimize value Vpcc</i>	$\{VoltageControl\}$
(3) Search the state that is related to V_{pcc} : <i>State and relatedTo value Vpcc</i>	$\{P, Q\}$
(4) Search the use cases that controls P : <i>PUC and Control value P</i>	$\{SelfConsumption, MarketService\}$

Appendix D. Parameters of the SLVGC Use Case

Table A6. Use case parameters.

Parameter	Value/Setting	Description
R	0.07 Ohm	resistance of the low voltage line
L	0.255 mH	inductance of the low voltage line
V	230 V rms	low voltage grid
f	50 Hz	nominal frequency of the grid
P_{max}	100 kW	maximum active power of the ESS converter
Q_{max}	100 kVAR	maximum reactive power of the ESS converter
S_{max}	100 VA	maximum apparent power of the ESS converter
V_{bat}	660 V	nominal voltage of the battery
C_{bat}	600 Ah	capacity of the battery
$SoC_{bat-init}$	50%	initial SoC of the battery
C_{sc}	50%	capacity allocated to the self-consumption use case
C_{ms}	50%	capacity allocated to the market service use case

References

- Liserre, M.; Sauter, T.; Hung, J. Future Energy Systems: Integrating Renewable Energy Sources into the Smart Power Grid Through Industrial Electronics. *IEEE Ind. Electron. Mag.* **2010**, *4*, 18–37.
- Li, X.; Hui, D.; Lai, X. Battery Energy Storage Station (BESS)-Based Smoothing Control of Photovoltaic (PV) and Wind Power Generation Fluctuations. *IEEE Trans. Sustain. Energy* **2013**, *4*, 464–473.
- Hollinger, R.; Diazgranados, L.M.; Braam, F.; Erge, T.; Bopp, G.; Engel, B. Distributed solar battery systems providing primary control reserve. *IET Renew. Power Gener.* **2016**, *10*, 63–70.
- IEC. *Draft IEC 61850-90-7 TR Communication Networks and Systems for Power Utility Automation*; Technical Report; International Electrotechnical Commission (IEC): Geneva, Switzerland, 2011.
- Büdenbender, K.; Braun, M.; Stetz, T.; Strauss, P. Multifunctional PV Systems offering additional functionalities and improving grid integration. *Int. J. Distrib. Energy Resour. Smart Grids* **2011**, *7*, 109–128.
- Hänninen, S.; Kyritsis, A.; Abdulhadi, I.; Schwalbe, R.; Strasser, T.; Kosmecki, M.; Sobczak, B.; Rink, R.; Kedra, B.; Wilk, M.; et al. WP 6 Controllable Flexibility Detailed Requirements and Constraints for the Control of Flexibility; Technical Report; ELECTRA Internal Report R6.1; ELECTRA IRP, 2015. Available online: http://orbit.dtu.dk/files/125919193/20150116_R6.1_DetailedRequirementsandConstraintsfortheControlofFlexibility_V1.02.pdf (accessed on 28 August 2017).
- EERA. *D4.3 Integration of Storage Resources to Smart Grids: Possible Services, D4.4 Control Algorithms for Storage Applications in Smart Grid*; EERA Joint Programme on Smart Grids Sub-Programme 4 Electrical Energy Technologies; Technical Report; EERA: Brussels, Belgium, 2014.
- Von Appen, J.; Stetz, T.; Braun, M.; Schmiegel, A. Local Voltage Control Strategies for PV Storage Systems in Distribution Grids. *IEEE Trans. Smart Grid* **2014**, *5*, 1002–1009.
- CEN-CENELEC-ETSI Smart Grid Coordination Group. *Reference Architecture for the Smart Grid*; Technical Report; CEN-CENELEC-ETSI: Brussels, Belgium, 2012.
- IEC. *IEC 62559: Use Case Methodology*; Technical Report; International Electrotechnical Commission (IEC): Geneva, Switzerland, 2015.
- Zanabria, C.; Pröbstl Andrén, F.; Kathan, J.; Strasser, T. An approach for the handling of controller conflicts within multi-functional energy storage systems. In Proceedings of the 24th International Conference on Electricity Distribution (CIRED), Glasgow, UK, 12–15 June 2017.
- Bletterie, B.; Tayyebi, A.; Kadam, S.; Le Baut, J.; Stöckl, J.; Kathan, J.; Einfalt, A. A novel concept for combining distribution network and system support services for storage systems. In Proceedings of the 2017 IEEE Manchester PowerTech, Manchester, UK, 18–22 June 2017; pp. 1–6.
- Perera, B.; Ciufu, P.; Perera, S. Advanced point of common coupling voltage controllers for grid-connected solar photovoltaic (PV) systems. *Renew. Energy* **2016**, *86*, 1037–1044.
- Consentec GmbH. *Description of Load-Frequency Control Concept and Market for Control Reserves*; Study Commissioned by the German TSOs; Consentec GmbH: Aachen, Germany, 2014.

15. Braam, F.; Hollinger, R.; Engesser, M.L.; Müller, S.; Kohrs, R.; Wittwer, C. Peak shaving with photovoltaic-battery systems. In Proceedings of the Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Istanbul, Turkey, 12–15 October 2014; pp. 1–5.
16. Riffonneau, Y.; Bacha, S.; Barruel, F.; Ploix, S. Optimal Power Flow Management for Grid Connected PV Systems With Batteries. *IEEE Trans. Sustain. Energy* **2011**, *2*, 309–320.
17. Andrén, F.; Lehfuss, F.; Strasser, T. A development and validation environment for real-time controller-hardware-in-the-loop experiments in Smart Grids. *Int. J. Distrib. Energy Resour. Smart Grids* **2013**, *9*, 27–50.
18. Zanabria, C.; Andrén, F.P.; Kathan, J.; Strasser, T. Towards an integrated development of control applications for multi-functional energy storages. In Proceedings of the IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, Germany, 6–9 September 2016; pp. 1–4.
19. Faschang, M.; Schwalbe, R.; Einfalt, A.; Mosshammer, R. Controller hardware in the loop approaches supporting rapid prototyping of smart low voltage grid control. In Proceedings of the 2014 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Istanbul, Turkey, 12–15 October 2014; pp. 1–5.
20. Hitzler, P.; Krötzsch, M.; Rudolph, S. *Foundations of Semantic Web Technologies*; Chapman & Hall/CRC: London, UK, 2009.
21. Schachinger, D.; Kastner, W.; Gaida, S. Ontology-based abstraction layer for smart grid interaction in building energy management systems. In Proceedings of the IEEE International Energy Conference (ENERGYCON), Leuven, Belgium, 4–8 April 2016; pp. 1–6.
22. Samirmi, F.D.; Tang, W.; Wu, Q. Fuzzy Ontology Reasoning for Power Transformer Fault Diagnosis. *Adv. Electr. Comput. Eng.* **2015**, *15*, 107–114.
23. Baader, F. *The Description Logic Handbook: Theory, Implementation and Applications*; Cambridge University Press: Cambridge, UK, 2003.
24. Brambilla, M.; Cabot, J.; Wimmer, M. Model-Driven Software Engineering in Practice. *Synth. Lect. Softw. Eng.* **2012**, *1*, 1–182.
25. Siegel, J. *Developing in OMG's New Model-Driven Architecture*; Object Management Group (OMG): Needham, MA, USA, 2001.
26. Dänekas, C.; Neureiter, C.; Rohjans, S.; Uslar, M.; Engel, D. Towards a model-driven-architecture process for smart grid projects. In *Digital Enterprise Design & Management*; Springer: Warsaw, Poland, 2014; pp. 47–58.
27. Andrén, F.; Strasser, T.; Kastner, W. Engineering Smart Grids: Applying Model-Driven Development from Use Case Design to Deployment. *Energies* **2017**, *10*, 374.
28. Working Group Sustainable Processes (SG-CG/SP). *CEN-CENELEC-ETSI Smart Grid Coordination Group—Sustainable Processes*; Technical Report; CEN-CENELEC-ETSI: Brussels, Belgium, 2012.
29. Kalibatiene, D.; Vasilecas, O. Survey on Ontology Languages. In *Perspectives in Business Informatics Research*; Grabis, J., Kirikova, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; Volume 90, pp. 124–141.
30. Tayyebi, A.; Bletterie, B.; Kupzog, F. Primary Control Reserve and Self-Sufficiency Provision with Central Battery Energy Storage System. In Proceedings of the NEIS Conference, Hamburg, Germany, 21–22 September 2017; in press.
31. Hua, Y.; Zander, S.; Bordignon, M.; Hein, B. From AutomationML to ROS: A model-driven approach for software engineering of industrial robotics using ontological reasoning. In Proceedings of the 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, Germany, 6–9 September 2016; pp. 1–8.
32. Apache Jena. Ontology API, 2012. Available online: <https://jena.apache.org/documentation/ontology/> (accessed on 12 March 2017).
33. World Wide Web Consortium. SPARQL 1.1 Query Language, 2008. Available online: <https://www.w3.org/TR/sparql11-query/> (accessed on 11 February 2017).
34. Tayyebi, A. Modelling and Simulation of a Multifunctional PV Electrochemical Storage System. Master's Thesis, University of Oviedo, Oviedo, Spain, 2016.
35. Yazdani, A.; Iravani, R. *Voltage-Sourced Converters in Power Systems: Modeling, Control, and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2010.

36. Bredillet, P.; Lambert, E.; Schultz, E. CIM, 61850, COSEM standards used in a model driven integration approach to build the smart grid service oriented architecture. In Proceedings of the First IEEE International Conference on Smart Grid Communications (SmartGridComm), Gaithersburg, MD, USA, 4–6 October 2010; pp. 467–471.
37. Heussen, K.; Gehrke, O.; Niemann, H. On early conflict identification by requirements modeling of energy system control structures. In Proceedings of the IEEE 20th Conference on Emerging Technologies & Factory Automation (ETFA), Luxembourg City, Luxembourg, 8–11 September 2015; pp. 1–8.
38. Uslar, M.; Heussen, K. Towards modeling future energy infrastructures—The ELECTRA system engineering approach. In Proceedings of the PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Ljubljana, Slovenia, 9–12 October 2016; pp. 1–6.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).