

RESEARCH

Open Access



# Towards model-driven CIM-based data exchange for DSOs

Dominik Ascher<sup>1\*</sup> and Christoph Kondzialka<sup>2</sup>

From The 7th DACH+ Conference on Energy Informatics  
Oldenburg, Germany. 11-12 October 2018

\*Correspondence:

[ascher@in.tum.de](mailto:ascher@in.tum.de)

<sup>1</sup>Technische Universität München,  
Boltzmannstr. 3, 85748 Garching b.  
München, Germany

Full list of author information is  
available at the end of the article

## Abstract

Smart energy systems (SES) promote the transformation of the distribution grid towards more sustainable operation and planning strategies, but also impose a set of considerable technological and political challenges. In this, distribution system operators (DSOs) are faced with the necessity of adapting their information system landscapes to enable the efficient utilization of information within their internal structures. In this work, we propose a model-based approach to derive an open middleware platform supporting the integration of existing system landscapes of DSOs. For this, we shortly describe a domain analysis on the DSO domain, which we use to derive the requirements of our platform. The platform is then implemented utilizing the Common Information Model (CIM) and open standards. Finally, we demonstrate the applicability of our approach within a small case study for a single use-case.

**Keywords:** Model-driven engineering, Domain analysis, Distribution system operation, Smart energy systems, Common Information Model (CIM)

## Introduction

Smart energy systems (SES) promote the transformation of the distribution grid towards more sustainable operation and planning strategies, but also impose a set of complex challenges such as high numbers of decentralized and renewable energy sources (RES) as well as energy storages, high penetration of electric vehicles (EV) as well as changing load profiles as a result of self consumption. To address these emergent challenges, planning and operation strategies have to accommodate changes and intermittency of loads occurring as a result of aforementioned challenges (Ramchurn et al. 2012). For instance, this includes utilization of intelligent load management mechanisms (Palensky and Dietrich 2011), predicting, scheduling and storing intermittent power loads from RES (Lei et al. 2009; Sharma et al. 2011; Barton and Infield 2004), accurately predicting power consumption (Zhao and Magoulès 2012) as well as scheduling charging or discharging EVs efficiently (Kempton and Tomić 2005). To enable these new applications and methods, detailed knowledge about the current state of the low-voltage distribution grid is necessary. By utilizing the broad availability of data from modern sensing technology in electricity networks as well as using information encoded in heterogeneous DSO systems, more efficient operation and planning strategies can be achieved and costs for system

operation can be decreased. However, for this it is required that available data is made available to different systems and applications as well as can also be efficiently exchanged between them.

Because of this, standardization represents a key challenge towards enabling smart energy systems (Uslar et al. 2010; Fan et al. 2013; Kuzlu et al. 2017), promoting seamless data exchange between different applications from different vendors and avoiding vendor-specific solutions. The Common Information Model (CIM) represents a comprehensive data model for the electric utility domain, and is described in the International Electrotechnical Commission (IEC) standard IEC 61970 (for energy management systems (EMS)) and related standards IEC 61968 (for information exchanges between electrical distribution systems) and IEC 62325 (for energy market communications). Overall, the CIM is intended to support description of both distribution and transmission energy systems in terms of a shared data model, which encodes a domain ontology of entities and their relationships. Thus, the CIM proposes a standardized vocabulary for data exchange between different information systems and data within electricity networks.

#### **Problem statement**

Previously, DSOs integrated new information systems into their infrastructure on behalf of respective departments, which then accessed systems through proprietary or application-specific interfaces, while using proprietary data models. As a result, information from these systems can often not be easily accessed by other systems or exchanged between different departments. As a result, often complex point-to-point integrations between systems are then performed, which lead to high follow-up costs and time expenditures, while impeding day-to-day operations. In addition, it is often difficult to replace one system from a given vendor within a system landscape as every connection to accessing systems must be updated. Furthermore, DSO software systems are often highly tailored to the specific requirements of a given DSO, making according DSO system landscapes highly susceptible to vendor lock-in.

#### **Contribution**

Unlike proprietary solutions, compliance with applicable standards ensures that the communication infrastructure and data employed can be linked to as many other systems as possible in order to be used efficiently in the future as well. This also alleviates vendor lock-in by providing shared data models and interface definitions which are compliant to applicable and widespread standards. To address these challenges, this work presents a platform for DSOs to integrate their information system landscapes while utilizing open standards. For this, our approach firstly describes a domain analysis of the DSO domain to elicit requirements and establish a domain model. Using the domain model, we then derive parts of the implementation of our technical platform within a model-driven engineering approach. Based on this, we then describe the platform architecture, which employs open standards for data exchange between integrated systems, while maintaining independence of individual software systems. Finally, we demonstrate the applicability our approach towards current challenges of a real DSO within a small case study for a single use-case.

### Structure

This paper is structured as follows. The “[Approach](#)” section elaborates on our approach for establishing a middleware platform, based on current challenges for DSOs and their system landscapes. The “[Domain analysis](#)” section then describes a conducted domain analysis to elicit and generalize information as well as requirements about the domain of distribution system operators. Based on this, the “[Model-based adapters](#)” section describes how parts of our middleware platform can be derived within a model-based approach. The “[Middleware platform](#)” section explores the platform architecture, which is supported by the model-based approach. The “[Case study](#)” section then elaborates on a case study demonstrating a single use-case of our platform. Then, in the “[Related work](#)” section, we explore related work. Finally, in the “[Conclusion](#)” section, we conclude.

### Approach

To support migration of DSOs towards domain-wide standards, we propose a middleware platform for DSOs based on the Common Information Model, which is supported by a model-based approach. Initially, we conducted a domain analysis with two German DSOs, where information and requirements about the DSO domain was collected, analyzed and structured. Based on the information captured, a domain model was derived, which generalizes domain knowledge of the DSO domain. Furthermore, we then identified relevant use-cases for our middleware platform.

Using the domain model, we then derive parts of the implementation of the middleware platform within a model-driven engineering approach. The approach is intended to support efficient adaptation of parts of our platform to different distribution system operator system landscapes by utilizing the shared structure of data models and systems of DSOs and by employing model-to-model transformations.

Then, we present a modular software architecture that provides an efficient method to access interfaces of DSO data systems by utilizing lean micro-services and established web protocols. Here, based on a system landscape employed by a given DSO, we describe how single systems can be integrated into the middleware platform. The approach offers modularity with respect to offering step-by-step integration of different DSO systems, while not impeding ongoing business-relevant and day-to-day operations. Our concept aims at mapping non-CIM data to the standardized CIM data model for integrated systems. We identified the latter to represent a major keystone ability for a comprehensive DSO data exchange platform.

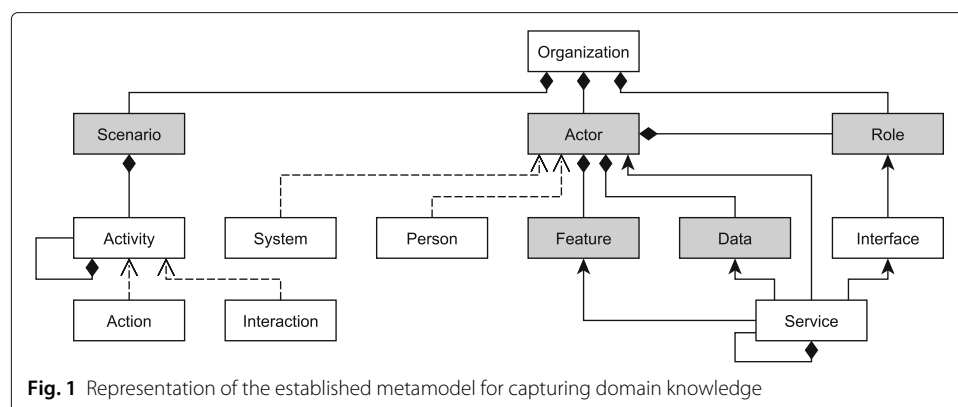
To apply our approach, we are implementing a vertical prototype for a single use-case scenario, which is limited in its functional scope. Functionally, the prototype demonstrates platform functionality for obtaining data necessary for network calculation, based on a data set representative of a medium-sized DSO’s grid planning department. In addition to functional requirements, additional functionalities such as a user interface and security requirements are addressed and evaluated within the prototype. For this, we followed a two-stage development methodology: Firstly, we integrated and tested the single features in our academic research environment. Then, secondly, we conducted a field test with a German DSO. The field test is performed by installing DSO data sources and systems on virtual machines equipped with real-world and representative DSO data. Note that direct integration into a DSO production environment would exceed available resources for implementation.

## Domain analysis

This section elaborates on a domain analysis for the distribution system operator domain. Here, to understand the domain and requirements of DSOs, a domain analysis was conducted with two German DSOs. Here, domain knowledge was captured in an interview process spanning multiple topic areas with different DSO departments intended to support capturing DSO requirements. For the interview process, we focused on internal DSO departments concerned with planning and operative asset management, system operation and network control, strategic asset management and regulation management, metering point operations, network documentation, accounting and purchasing as well as IT infrastructure management. For these topic areas, the domain analysis then specifically focused on eliciting requirements for DSOs with respect to challenges arising in the context of establishing smart energy systems. Based on the information elicited, we established a metamodel for capturing domain knowledge. The metamodel is visualized in Fig. 1.

Subsequently, we describe the classes of the metamodel in detail:

- *Organization*. An organization refers to an entity, which represents an organizational unit in terms of its departments or sub-organizations. Here, an organization realizes a set of *scenarios*, i.e. a set of activities which support the business processes of the DSO, and contains a set of *actors* and *roles*.
- *Scenario*. Scenarios represent sequences of *activities*, i.e. granular *actions* or *interactions*, which support the business processes of an organization. Activities are generally performed by a set of actors such as *persons* or *systems*.
- *Actor*. An actor refers to a *system* or *person*, which perform sequences of activities. Here, each activity is initiated and executed by at least one actor. Note that for this, an actor utilizes *data* and *features*, i.e. predefined functionality achieving a particular goal. Actors are specialized as follows:
  - *Person*. A person represents an actor, which performs an activity manually, i.e. without a software system supporting the activity.
  - *System*. A system represents an actor, which performs an activity automatically or interactively (i.e. together with another actor).
- *Interface*. An interface defines a formal description for accessing specific data. Here, an interface can be utilized for interaction between *services*. For this, an interface defines a set of *roles* which are allowed to access it.



**Fig. 1** Representation of the established metamodel for capturing domain knowledge

- *Data*. Data refers to domain-specific and business-specific structured information in terms of underlying data classes, types, attributes and the relationships between them. Note that data is managed by an accountable actor.
- *Role*. Roles refer to organizational units with a defined set of objectives such as internal departments or external business partners. Furthermore, we use roles to model access control, i.e. permissions on interfaces. Therefore, depending on the role, an interface can allow or deny access to specific resources.
- *Activity*. Activities represent tasks, which can be performed manually, automatically or as an interactive sequence.
  - *Action*. If initiation and execution are performed by the same single actor, we refer to this activity as an action.
  - *Interaction*. If initiation and execution are dependent on multiple different actors, we refer to this activity as an interaction.
- *Feature*. Features describe individual functionalities to transform data, which are realized by individual actors such as systems or persons, as well as the relationships between them.
- *Service*. A service represents a subset of the functionality, which can be performed by an actor. Here, a service can be responsible for data access, data transformation and validation and is accessed through a given interface.

Note that, based on the established metamodel, we then derive a general domain model for the domain of distribution system operators, which we described in detail in previous work (Ascher and Bytschkow 2018). Here, a domain model represents a model about the relevant parts of an application domain (Broy 2013). In this context, our domain model describes common domain elements about the distribution system operator domain, which are shared between different DSOs (e.g. common domain entities such as SCADA or GIS systems), but depending on the DSO, may occur in different representations such as SCADA systems from specific vendors or specific configurations. The goal of a domain model is to capture information about the problem domain itself, rather than the solution domain. In fact, requirements elicitation and design for a given solution can then be performed based on the domain model (See “[Model-based adapters](#)” and “[Middleware platform](#)” sections).

### **Model-based adapters**

This section describes model-based generation of specific parts of the implementation of our overall platform. More specifically, we describe code generation of so-called adapters by automatically deriving implementation code from an implementation model as well as the domain model.

Our approach for domain modeling defines an instrument to describe the problem domain of distribution system operators, which allows describing entities and relations of a specific problem domain. For this, we employ Multi-Level-Modeling (Atkinson and Kühne 2003), which allows us to model an arbitrary number of modeling levels, where each level describes a concretization of its parent level. More specifically, our approach defines different hierarchical modeling levels  $M(\cdot)$  for representation of domain knowledge. Here, the metamodel level ( $M3$ ) defines generic domain model entities and the relations between them (See “[Domain analysis](#)” section). Then, as an instance of the

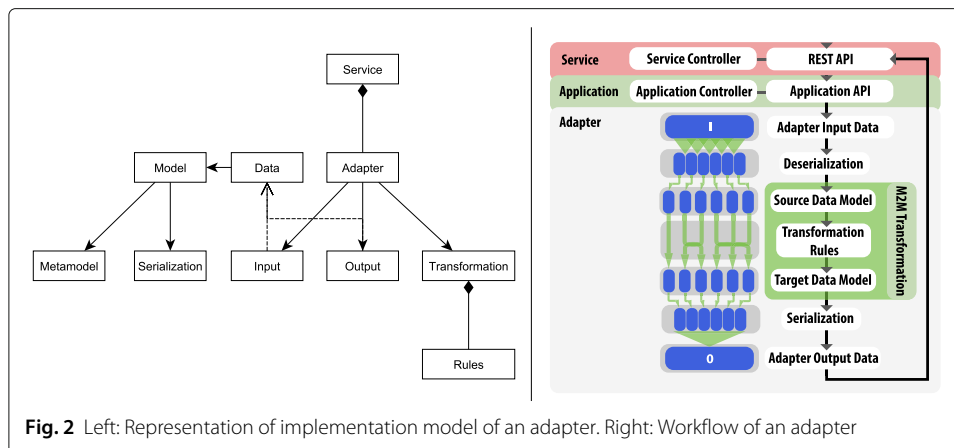
metamodel, the domain model ( $M2$ ) describes more specific domain entities, which occur in DSOs, such as SCADA or GIS systems of a specific vendor or data models employed by a specific metering application. Finally, as instances of the domain model, instance models ( $M1$ ) describe concrete instantiations of the domain model for specific DSOs such as a specific SCADA system of a given vendor configured for a single DSO.

Based on the information encoded within the domain model ( $M2$ ) and instance model ( $M1$ ) levels, we are able to perform data transformations from data encoded within one system to another data model. More specifically, Model-to-Model (M2M) transformations are defined to convert one data model to another, where the transformation of a source data model to a target data model utilizes a set of functional transformation rules. Thus, the transformation rules define the conversion from individual data entities in the source model to data entities in the target model. Note that transformation rules are defined on the domain model ( $M2$ ) level and are therefore independent of the specific instance data of a particular DSO. Then, based on the defined transformation rules, a transformation of specific instance data such as a specific network topology in an application-specific data model to another can be performed. In our approach, we employ M2M transformations to convert non-CIM data, which for example occurs in proprietary, application-specific data models to standardized CIM data, based on the CIM data model, and vice versa.

Conceptually, to then integrate these M2M transformation with a given system or application, we employ *adapters*, which are part of our platform. The core idea of adapters is to enable software systems to remain independent, where each system integrated within our platform is wrapped with its own specialized adapters. More specifically, adapters represent components of the proposed platform, which read the data input of an application, deserialize this input, map it into to a given, known source data model (i.e. the data model of a given system or application), perform a M2M transformation to a target data model, and then write the resulting output data after serializing it. Note that, in order to read data inputs from an application, we assume that the application produces this data in response to a specific query and that this data is made available to the adapter, for example as a local file. In fact, here *services* produce the query to a given application and execute the workflow of an adapter. An extensive description of services is provided in “[Middleware platform](#)” section. The workflow of an adapter is visualized in the right part of Fig. 2.

Technically, to implement an adapter within a given scenario (e.g. a sequence of activities with a given system), we define a generic implementation model. Here, as required conceptually, the implementation model of an adapter encodes the information necessary to read an input, perform a transformation, and write a specific output. Based on the system an adapter is implemented for, input and output data models are referenced using the information encoded within the domain model, i.e. the proprietary data model for the given system as well as the standardized CIM data model, the transformation should be performed to or from. The implementation model of an adapter is visualized in the left part of Fig. 2.

Based on this information, implementation code is then derived from the implementation model using code generation. To then tailor the implemented adapter to a specific DSO system configuration, we supplement the information encoded in the implementation code with information from the specific instance model (of the domain model)



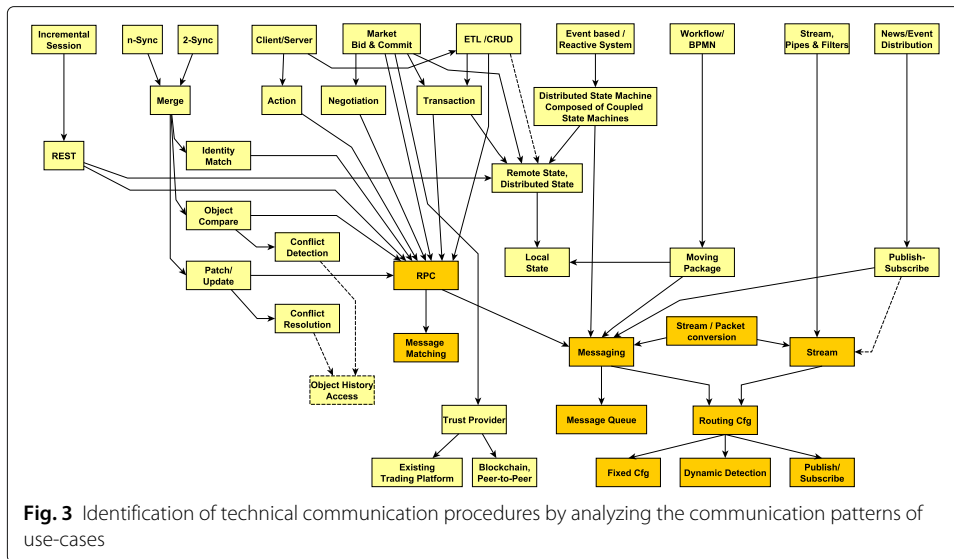
for the given DSO. Therefore, using the information encoded in the instance model, we adapt adapters dynamically to the specific system configuration of a DSO. Note that the approach is intended to support easy platform adaptation for heterogeneous DSOs by utilizing commonalities in data models and systems.

### Middleware platform

The technical part of our platform was aligned along the requirements and use-cases, which were identified during the domain analysis of the participating DSOs (see “[Domain analysis](#)” section). Over 100 use-cases were determined and described during this analysis, for the reason to mark the communication patterns behind it. We consider patterns as a sequence of events such as versioning and merge, market processes such as offer/bid/dealclose and underlying distribution of data streams.

Afterwards, an architectural style viable for the requirements of DSOs were chosen, based on correlation between communication patterns and corresponding communication methods (see Fig. 3). As it fully covers the communication patterns identified, the Representational State Transfer (REST) architectural style (Medvidovic and Taylor 2010) was selected. REST denotes a architectural style for distributed systems in general and web services in particular, and defines a stateless client-server protocol. In addition, is scalable with respect to queue-based methods, is suitable for integrating current safety requirements and technology as well as enables efficient system integration, control and maintainability. Furthermore, REST offers lightweight applicability and flexibility in industrial IT environments. By utilizing established and open technologies within our approach, system operators can choose to combine open-source and closed-sourced systems or components, which enables synergies between them. By means of integrating independent applications into the platform, a wide variety of different functionality can be accommodated. This includes the connection of simulation tools to grid calculation applications, the utilization and condition monitoring of the distribution grid, the prediction of energy consumption and weather, and addition of new energy systems as well as changing load profiles (Alberts et al. 2014).

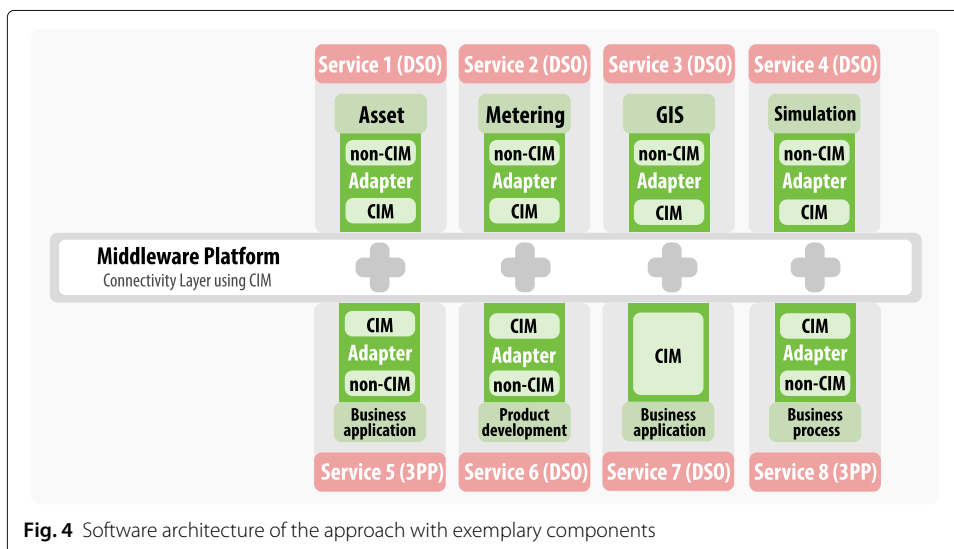
As shown in Fig. 4, the primary software systems remain independent, and are each wrapped with their own specialized adapters. Adapters then allow data exchange within



**Fig. 3** Identification of technical communication procedures by analyzing the communication patterns of use-cases

our platform. The bottom line shows a selection of further configurations when connecting new systems. Depending on the components involved, a wide variety of integration variants can be implemented.

Single use-cases are then realized using *services*. Two types of services are employed: *use-case services* and *application services*. Use-case services are responsible for the coordination of the interaction of individual components within the respective use-case, perform the control flow of involved data and provide a corresponding user interface. Instead, application services encapsulate either an existing specialized technical application or provide subject-specific functionality. Using this, application services only have to be implemented once and can be reused by services of different use-cases. As shown in Fig. 5, each service is consisted of a *server-module* and a *core-module*. Core-modules include specialized control logic for accessing data or processing subject-specific data. Note that implementation code for integrating subject-specific applications and code for



**Fig. 4** Software architecture of the approach with exemplary components



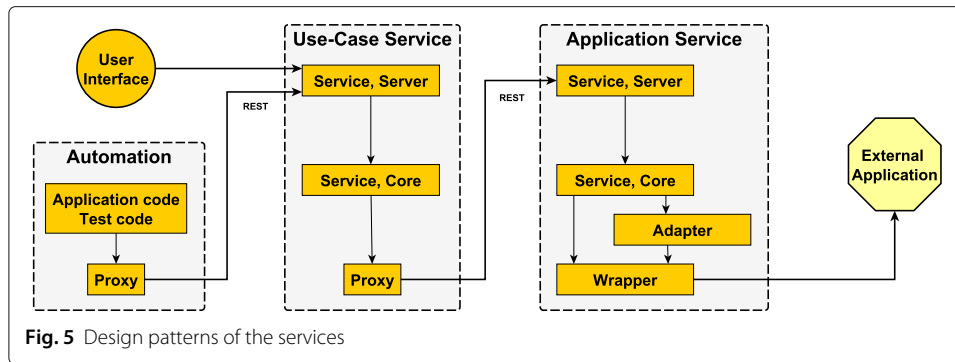


Fig. 5 Design patterns of the services

core-modules doesn't comprise a REST interface. Implementation is performed in the given target language such as Python or Java. In addition, the complete control flow of the realized use-case is implemented in core-modules. Then, finally, adapters for the conversion from and to the CIM are integrated within core-modules.

The server-module contains the main-loop and converts REST to procedure calls. For parallel calls, the server provides separate threads as process environments. Likewise, an exception forwarding mechanism is realized, where exceptions are translated by the server-module into a REST string and back into an exception by the calling proxy. Communication with other services is performed via the proxy-module, which converts service-internal function calls to REST for each service.

**Case study**

As part of the project, one scenario was selected from 100 different use cases identified as part of the performed domain analysis and was implemented within a case study. For this, the scenario "data acquisition for network calculation" was selected, which represents a vertical prototype covering all relevant features of the proposed platform. In Fig. 6, the scenario is visualized conceptually.

In terms of data, the scenario covers a wide range of links between data sources of different IT systems that are relevant to network calculations for DSOs. In terms of source systems, data from a GIS system, a metering system and an asset management system

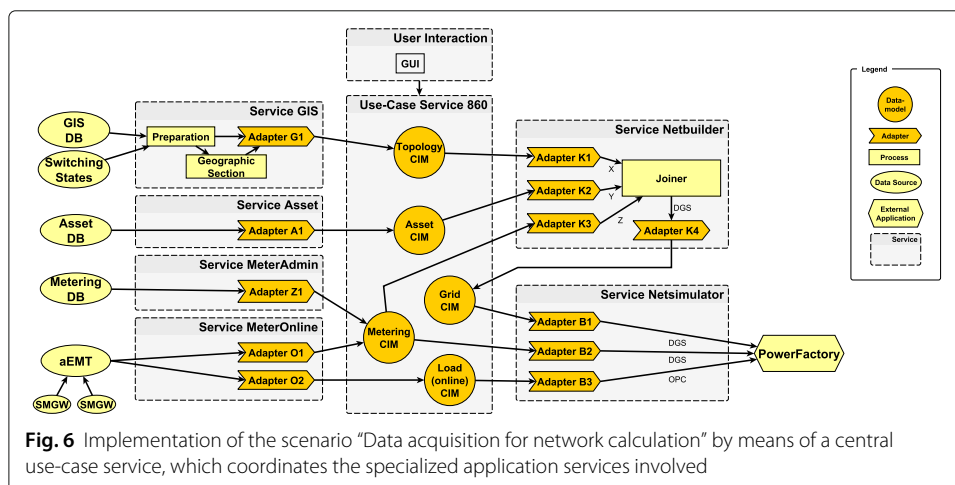


Fig. 6 Implementation of the scenario "Data acquisition for network calculation" by means of a central use-case service, which coordinates the specialized application services involved

is utilized. In this concrete scenario, the target system represents an application for network calculation. Additionally, a subject-specific service was employed between the data sources and target system. This service enriches the network data provided by the GIS system with asset and metering data in order to generate a low-voltage grid model that is consistent from medium voltage to every single low-voltage house connection. The goal of the prototype was to cover a scenario highly relevant for and benefiting to a participating DSO, while enabling application of the approach in a real-world setting.

### Related work

The Common Information Model (CIM) has been recognized as a key element to connect different DSO systems and applications using a standardized data model and enable interoperability for smart grid applications (Hargreaves et al. 2014; Ascher and Bytschkow 2018). In that, a substantial challenge consists in bridging the gap between application-specific data models and the application-independent CIM data model in order to make data accessible for applications requiring multiple data sources such as power system simulations. As a result, several approaches have focused on this challenge in the past.

The CIM2Matpower (CIM2Matpower) package supports transformation of a CIMv14 ENTSO-E profile transmission system network model to a Matpower case structure (Zimmerman et al. 2011). However, currently, the tool is only intended to support transmission system operator data structures. Armendariz et al. (2015) introduce a method to convert proprietary application data models to the MATLAB Matpower data model (Zimmerman et al. 2011) by utilizing the CIM. The developed tool is intended to support the conversion from utility data such as network topologies and equipment data into MATLAB Matpower to support simulation studies evaluating different technical options for future DSO operations. Gomez et al. (2017) propose an model-driven engineering approach which applies model-to-model transformations between CIM and the Modelica system modeling language (Mattsson et al. 1998). Here, the approach focuses on using M2M transformations to derive Modelica models for dynamic power system simulations only. In contrast, the scope of our approach aims to provide a holistic platform employing adapters between heterogeneous DSO applications and the CIM. Finally, Bredillet et al. (Bredillet et al. 2010) propose a smart grid service-oriented architecture, utilizing model-driven development and incorporating standards such as the Common Information Model (CIM) and the IEC 61850 standard. Goering et al. (2016) propose multi-layer service-oriented architecture utilizing the CIM. Finally, Haq et al. (2011) leverage the CIM to integrate systems from different application vendors within an Service Oriented Architecture.

### Conclusion

In this work, we proposed a modular middleware platform to promote the integration of heterogeneous DSO information systems. Initial results are promising with regard to the applicability of our model-based approach, with the proposed solution being suitable for reducing costs of utilizing information from different DSO systems by standardizing their data models and interfaces. As the platform employs the CIM, data exchange is independent from proprietary data models. Currently, we employ subsets of the CIM, i.e. CIM profiles, for source systems, and only utilize entities contained in the CIM standard. However, for future work, it has to be investigated to which degree the CIM can cover required

entities in order to describe the DSO domain sufficiently complete, without requiring proprietary extensions. In fact, as CIM development continues, efforts to establish a standardized CIM profile for DSOs (analogous to the CGMES profile of the ENTSO-E for transmission system operators which currently dominates application landscapes), which covers all requirements of the DSO domain, will be vital. Furthermore, currently, new CIM versions are provided without instructions for migrating models from one version to another, which complicates the maintenance of existing models. Here, future work has to investigate whether our model-based approach can support migration from different CIM versions.

#### Abbreviations

CIM: Common information model; DSO: Distribution system operator; EMS: Energy management systems; ENTSO-E: European network of transmission system operators for electricity; EV: Electric vehicles; GIS: Geographic information system; IEC: International electrotechnical commission; M2M: Model-to-model; RES: Renewable energy sources; REST: Representational state transfer; SES: Smart energy systems; SCADA: Supervisory control and data acquisition

#### Acknowledgements

We would like to sincerely thank our project partner, the software company SEKAS GmbH, for the cooperation. Furthermore, we would like to thank the participating municipal utilities of Stadtwerke Ulm/Neu-Ulm Netze GmbH and Stadtwerke Pfarrkirchen for the intensive cooperation.

#### Funding

Publication costs for this article were sponsored by the Smart Energy Showcases - Digital Agenda for the Energy Transition (SINTEG) programme. Furthermore, publication of this article was sponsored by the German Federal Ministry for Economic Affairs and Energy as part of the German federal research initiative Future-proof Power Grids within the research project ESOSEG (Grant 0325811A-C). The funding leading to the results presented in this article is gratefully acknowledged.

#### Availability of data and materials

The datasets generated during and/or analysed during the current study are not publicly available due to existing confidentiality agreements with third party partners but are available from the corresponding author on reasonable request and permission of partners.

#### About this supplement

This article has been published as part of *Energy Informatics* Volume 1 Supplement 1, 2018: Proceedings of the 7th DACH+ Conference on Energy Informatics. The full contents of the supplement are available online at <https://energyinformatics.springeropen.com/articles/supplements/volume-1-supplement-1>.

#### Authors' contributions

All authors participated in drafting and revising the final manuscript and contributed to this study. DA focused on domain analysis and the model-based approach in this manuscript. CK focused on the middleware platform and implementation of a practical field test scenario in this manuscript. Both of the authors have read and approved the final manuscript.

#### Competing interests

The authors declare that they have no competing interests.

#### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

#### Author details

<sup>1</sup>Technische Universität München, Boltzmannstr. 3, 85748 Garching b. München, Germany. <sup>2</sup>Hochschule Ulm, Eberhard-Finckh-Str. 11, 89075 Ulm, Germany.

Published: 10 October 2018

#### References

- Alberts I, Stauch V, Lee D, Hagedorn R (2014) Probabilistic weather information tailored to the needs of transmission system operators. In: AGU Fall Meeting Abstracts. American Geophysical Union, San Francisco
- Armendariz M, Saleem A, Nordström L, Brugeron M (2015) Facilitating distribution grid network simulation through automated common information model data conversion. In: PowerTech, 2015 IEEE Eindhoven. IEEE, Eindhoven. pp 1–6
- Ascher D, Bytschkow D (2018) Integrating distribution system operator system landscapes. *Comput Sci Res Dev* 33(1):169–175
- Atkinson C, Kühne T (2003) Model-driven development: a metamodeling foundation. *IEEE Softw* 20(5):36–41
- Barton JP, Infield DG (2004) Energy storage and its use with intermittent renewable energy. *IEEE Trans Energy Convers* 19(2):441–448

- Bredillet P, Lambert E, Schultz E (2010) CIM, 61850, COSEM standards used in a model driven integration approach to build the smart grid service oriented architecture. In: Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference On. IEEE, Gaithersburg. pp 467–471
- Broy M (2013) Domain modeling and domain engineering: Key tasks in requirements engineering. In: Perspectives on the Future of Software Engineering. Springer, Berlin Heidelberg. pp 15–30
- CIM2Matpower. <https://github.com/kkgerasimov/CIM2Matpower>. Accessed 30 May 2018
- Fan Z, Kulkarni P, Gormus S, Efthymiou C, Kalogridis G, Sooriyabandara M, Zhu Z, Lambotharan S, Chin WH (2013) Smart grid communications: Overview of research challenges, solutions, and standardization activities. *IEEE Commun Surv Tutorials* 15(1):21–38
- Goering A, Meister J, Lehnhoff S, Jung M, Rohr M, Herdt P (2016) Architecture and quality standards for the joint development of modular open source software for power grid distribution management systems. In: 5th DA-CH+ Energy Informatics Conference in Conjunction with 7th Symposium on Communications for Energy Systems (ComForEn), Vol. 36. Österreichischer Verband für Elektrotechnik, Wien
- Gómez F, Vanfretti L, Olsen SH (2017) CIM-Compliant Power System Dynamic Model-to-Model Transformation and Modelica Simulation. *IEEE Trans Ind Inform* 14(9):3989–3996
- Haq E, Haller D, Rahman KA, Iverson B (2011) Use of common information model (cim) in electricity market at California ISO. In: Power and Energy Society General Meeting, 2011 IEEE. IEEE, Detroit. pp 1–6
- Hargreaves N, Pantea SM, Taylor G (2014) Adopting the IEC Common Information Model to Enable Smart Grid Interoperability and Knowledge Representation Processes. Springer Singapore, Singapore. pp 439–462
- Kempton W, Tomić J. (2005) Vehicle-to-grid power implementation: From stabilizing the grid to supporting large-scale renewable energy. *J Power Sources* 144(1):280–294
- Kuzlu M, Pipattanasompom M, Rahman S (2017) A comprehensive review of smart grid related standards and protocols. In: Smart Grid and Cities Congress and Fair (ICSG), 2017 5th International Istanbul. IEEE, Istanbul. pp 12–16
- Lei M, Shiyun L, Chuanwen J, Hongling L, Yan Z (2009) A review on the forecasting of wind speed and generated power. *Renew Sust Energy Rev* 13(4):915–920
- Mattsson SE, Elmqvist H, Otter M (1998) Physical system modeling with Modelica. *Control Eng Pract* 6(4):501–510
- Medvidovic N, Taylor RN (2010) Software architecture: foundations, theory, and practice. In: Software Engineering, 2010 ACM/IEEE 32nd International Conference On. IEEE, Cape Town Vol. 2. pp 471–472
- Palensky P, Dietrich D (2011) Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE Trans Ind Inform* 7(3):381–388
- Ramchurn SD, Vytelingum P, Rogers A, Jennings NR (2012) Putting the ‘smarts’ into the smart grid: a grand challenge for artificial intelligence. *Commun ACM* 55(4):86–97
- Sharma N, Sharma P, Irwin D, Shenoy P (2011) Predicting solar generation from weather forecasts using machine learning. In: Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference On. IEEE, Brussels. pp 528–533
- Uslar M, Rohjans S, Bleiker R, González J, Specht M, Suding T, Weidelt T (2010) Survey of Smart Grid Standardization Studies and Recommendations–Part 2. In: Smart grid communications (SmartGridComm), 2010 first IEEE international conference on. IEEE, Gothenberg
- Zhao H-x, Magoulès F (2012) A review on the prediction of building energy consumption. *Renew Sust Energy Rev* 16(6):3586–3592
- Zimmerman RD, Murillo-Sánchez CE, Thomas RJ (2011) MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. *IEEE Trans Power Syst* 26(1):12–19

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](http://springeropen.com)

---