

A Toolchain for the Development of Agent-Based Smart Grid Control Solutions

Sebastian Törsleff*, Christian Derksen**, Marcel Ludwig***, Tobias Linnenberg*, Erik Wassermann*, Nils Loose**,
Alexander Fay*

* *Institute of Automation Technology, Helmut-Schmidt-University, Hamburg, Germany, (e-mail: sebastian.toersleff@hsu-hh.de)*

** *Research Group for Business Informatics and Software Engineering, University of Duisburg-Essen, Essen, Germany, (e-mail: christian.derksen@uni-due.de)*

*** *Institute of Power Systems Engineering, University of Wuppertal, Wuppertal, Germany, (e-mail: marcel.ludwig@uni-wuppertal.de)*

Abstract: The energy transition necessitates a variety of control solutions to integrate renewables-based energy generation and unlock demand-side flexibility. To contribute to solving this challenge, the authors built a toolchain for developing smart grid control solutions. This toolchain consists of methodological, conceptual and technological components. In this contribution, we detail the way in which the toolchain facilitates the continuous utilization of control software artifacts across simulations, testbeds and deployment. This capability has the potential to significantly reduce engineering effort and build trust in agent-based control solutions for smart grids.

Keywords: smart grid, grid operation, multi-agent systems, distributed control, active network management, agent-oriented software engineering.

1. INTRODUCTION

The advent of small-scale distributed energy resources, specifically those based on renewables, poses a significant challenge to the operation of electricity grids. In this context, the intelligent control of grid-connected assets, referred to as smart grid solutions throughout this paper, presents a viable and cost-effective alternative to conventional grid expansion measures (Harnisch et al., 2016; Ochoa et al., 2010).

Our future energy system will be hugely diverse across regions, not only from a geographical perspective, but also from a temporal and regulatory point of view. The geography of a region determines the availability of primary energy, e.g. solar radiation. This availability is also determined by temporality, i.e. depending on the time of the day or the season. Furthermore, different policy frameworks, e.g. central versus peer-to-peer marketplaces, will entail a further variety in requirements that smart grid solutions have to satisfy (Tuballa and Abundo, 2016). To meet these diverse requirements, a multitude of smart grid solutions will be required across the globe.

A lot of research is dedicated to satisfying this demand (cf. section 2). However, many scientific approaches in this field have not been implemented in real smart grids. To contribute

to closing this gap, a toolchain that facilitates the efficient development of smart grid solutions has been developed in the research project Agent.HyGrid¹. This toolchain comprises an engineering methodology and several software tools. The methodology, called 2DECS (Development approach for DEcentralised Control Systems), is tailored to the domain-specific needs of smart grid solutions, e.g. aspects concerning regulation and standardization, and covers the entire lifecycle of a smart grid control system, from design to implementation and later adaptation (Linnenberg and Fay, 2018). The software tools support the development and validation phase and help to reduce the development effort through the provision of reusable software components and the facilitation of software artifact continuity, e.g. using the software code for a control algorithm across simulation, testbed and field deployment. In this contribution we focus on the components and mechanisms that enable this continuity.

Section 2 discusses the related research. In section 3, the methodological and technological components of the toolchain are introduced. Afterwards, the continuous utilization of software artifacts along the project phases is presented in section 4. Section 5 shows how the toolchain is applied to building an active network management solution for the campus grid of the University of Wuppertal. The conclusion and an outlook are provided in section 6.

¹ www.agent-hygrid.net

2. RELATED RESEARCH

The area of research dealing with the development smart grid control solutions combines elements from classical energy system analysis and operational grid control to analyze operational implications of changes to the grid, e.g. integration of distributed energy sources such as electric vehicle charging stations. Simulation tools used in this research domain are, e.g., GridLAB-D, μ GRiDs, GridSim OpSim and pandaPower (Chassin et al., 2014; Nguyen et al., 2017; Sawant and Doan, 2017; Thurner et al., 2017). These tools are, however, primarily used for simulation purposes and do not facilitate a seamless field deployment of developed algorithms. Furthermore, these models are limited to simulating electricity grids. One approach to mitigate the latter limitation are co-simulation tools as presented in (Godfrey et al., 2010) or as implemented in *mosaik* (Lehnhoff et al., 2015). These tools can simulate the behavior of smart grid solutions across energy carrier domains. However, just like previously mentioned simulation tools, developed control software cannot be seamlessly deployed in the field.

Regarding actual implementations of smart grid solutions, numerous approaches were published in recent years. One example for this is the agent based PowerMatcher (Kok et al., 2005) a market-based control concept to balance supply and demand. In (Hashemi et al., 2015) and (Schwalbe et al., 2015) the authors present field-tested control approaches that increase the hosting capacity in low voltage grids using active transformers. (Luo et al., 2015) developed a distributed control technique for avoiding thermal overloads. An agent-based virtual power plant consisting of wind power generators and electric vehicles is presented in (Vasirani et al., 2013). A congestion management control approach based on agents is introduced in (Hu et al., 2015). The coordinated control of distributed generators using a virtual power plant is addressed in (Gan et al., 2013). Surveys of agent-based implementations are provided in (Ghribi et al., 2014; Shawon et al., 2019; Vrba et al., 2014). In general, these implementations are problem specific approaches, few of which have been tested in the field. Hence, there is a lack regarding tools that facilitate the development of smart grid solutions across the entire lifecycle which we intend to close.

3. TOOLCHAIN COMPONENTS

The toolchain comprises an engineering process and software tools. While the focus of this contribution lies on the continuous utilization of software artifacts along the project phases enabled by the toolchain, this section introduces the toolchain's main components to set the stage for the focus of this contribution.

3.1 2DECS

The main value proposition of the "Development approach for DEcentralised Control Systems" (2DECS) is to build trust in control solutions on the customer and operator side of a development project by ensuring that all relevant standards and requirements are taken into account (Linnenberg and Fay,

2018). Derived from established iterative processes such as SCRUM, VModell XT and Xtreme Programming, as well as agent-oriented approaches like GAIA, TROPOS and O-MaSE, it incorporates organizational as well as technical aspects of development projects. Considering the high coverage of engineering phases of the aforementioned processes they can be regarded as a good foundation for the development of a new methodology, tailored to the needs of software engineering in energy grid control solution projects. Even though each of the existing methodologies displays individual advantages, the definition of a new approach was found necessary as none of them supports all requirements that software engineering of agent-based control solutions in the energy domain imposes. Namely those requirements are: regulatory compliance of the solution; support of a generic communication architecture within the product; promotion of safety and security awareness on all levels of the development team; strong customer orientation; model, syntax and format consistency throughout all stages of the process. 2DECS is designed to meet all those requirements in order to support the development of decentralized control solutions in the energy domain.

The process meta-model is made up of five elements: artifacts, activities, phases, roles and tools. The process leaves it to the engineers to choose the appropriate tools, however, the utilization of Agent.Workbench (see section 3.2) and the Energy Option Model (see section 3.3) is recommended. In the following, only the phase model is discussed in detail as it

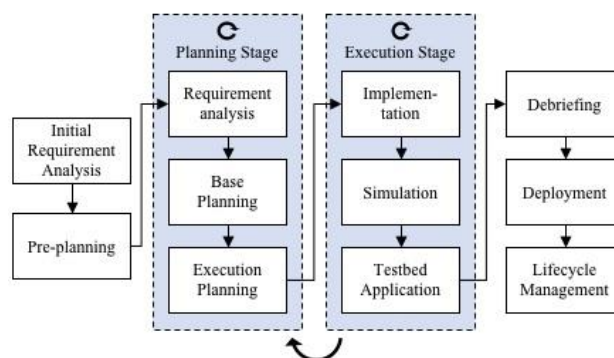


Fig. 1: Simplified 2DECS Phase Model.

serves to contextualize deployment stages, which are the focus of this contribution in section 4. A simplified and reduced version of the phase model is depicted in Fig. 1. Some of the phases are grouped into stages. These stages are characterized by the fact that they are iterative in themselves but also in-between one another, thus allowing to code, test and repeat in multiple cycles.

After the initial *Requirements Analysis* and *Pre-planning* phases, in which the project lead and the customer define the high-level project goals (i.e. *Definition of Done* and initial *User Stories*), the *Planning Stage* is entered. It encompasses *Requirements Analysis*, *Base Planning*, and *Execution Planning*. The *Planning Stage* deals with formalizing requirements and the prioritization and scheduling of work packages. This forms the basis for the *Execution Stage*. Multiple iterations of this stage are proposed. Defined features

are developed and tested iteratively. After passing the *Simulation* phase, individual features, or an aggregation of several features, is tested in a *Testbed Application*. If one of the tests fails, a fallback into the *Implementation* phase is required in order to eliminate errors.

After passing the *Execution Stage*, a *Debriefing* of all stakeholders is arranged in order to give feedback on the progress made and eventually collect new *User Stories*. From this phase it is possible to either continue to the *Deployment Phase* or return to the *Planning Stage*. In the first case the software is deployed in the customers' control system and therewith the *Increment* is turned into a *Release*. After deployment, a return into the *Planning Stage* allows to implement and provide further features as the product gains substance. Alternatively, the project can be closed, and the *Lifecycle Management* phase is entered.

3.2 Agent.Workbench

Formally named Agent.GUI (Derksen et al., 2012), Agent.Workbench was initially created as a general-purpose tool that enables developers to address end user needs for the utilization of agent-based systems through a visual representation. For this purpose, the Agent.Workbench offers both, a predefined end user application and a general extensibility that enables to integrate any kind of application based on Java and the widely used agent platform JADE².

With the predefined visual application, end users can define agent projects and configure them for different aspects, like additional Java resources, so called Project-Plugins, the ontologies used in an agent system, the required JADE platform settings, as well as parameters for a static or dynamic load balancing for cases in which the developed multi-agent system is used in a distributed manner. To execute the same multi-agent system in different scenarios, setups can be defined. Herein, the agents can be visually selected, configured and finally started and monitored using the application, too. If required, agents may share an environment (model) in which they operate and interact. For this purpose, a time model in combination with any data model can be defined. As a predefined environment model, Agent.Workbench offers a generic graph or network model that is especially useful for the visual configuration of topologies in the context of energy applications (e.g. for electricity, heat and gas networks).

Focusing on the execution of in-field Energy Agents, Agent.Workbench may be executed in different ways. Summarized by the notion "Embedded System Agent", it allows the execution of selected agents without any visual representation. Thus, it enables the execution of agents on, e.g., industrial PCs or Raspberry PI's. In Agent.HyGrid and with respect to the Energy Agent approach, Agent.Workbench serves as base layer. It is open source and freely available on GitHub³.

3.3 Energy Option Model

The Energy Option Model (EOM) (Derksen and Unland, 2016), is a modelling framework that enables the description of the operational flexibility of energy conversion systems. Integrated into Agent.Workbench as an additional feature, that enables a visual definition and the development of individual, system dependent software fragments (e.g. static data models for ramp curves or methods to calculate energy flows). On that basis, self-containing OSGI-bundles can be created that provide a detailed description of single systems, which can be used in a unified way and in different contexts - as for example by a single Energy Agent or in system aggregations. A single energy systems base model comprises several aspects, which are outlined in the following.

IO-List: This list contains all information that can be exchanged with the corresponding energy system. In particular, measurements and set points are defined here, but also individual static data models that help to parametrize a model with respect to the actual system (e.g. the positioning of a photovoltaic plant or the ramp curve of a combined heat and power plant)

System Usage: Since electrical vehicles can be charged in different ways or washing machines run with different programs, the EOM differentiates in that respect and provides the modelling abilities for these specific purposes. Thus, different system usages can be defined, but also different levels of details (e.g. for different timescale, such as real time control or long-term planning models)

Technical Interfaces: With respect to the physical system boundaries, connecting elements to an energy network can be considered here. Based on specific domain models, different energy carriers can be handled in detail (e.g. reactive power components in electrical grids or the gas compositions in gas networks), while the actual energy flows and amounts are always handled as SI energy units (e.g. kWh, MW etc.). This enables the utilization of the EOM in cross-domain scenarios (sector coupling).

System States: By means of a state machine, the operating states of system can be modeled. Thus, batch processes or continuous system operations can be modelled. The durations of individual states can be defined to discretize the model.

Energy Flows: Based on the state machine and the defined technical interfaces, the definition of energy flow calculations concludes the definition of an EOM-based model. Here, static energy flows, empirical data or complex calculations that consider measurements and set points can be defined as the basis for an energy flow determination.

In the further course of the usage of the EOM, the described base model serves as unified base for decision making processes that are either used for a real time control or for longer planning processes (e.g. day ahead).

² <http://jade.tilab.com>

³ <http://github.com/EnFlexIT/AgentWorkbench>

Handling aggregations is the second important capability of the EOM. Aggregations are synonymous to system of systems, which means that parent systems can be described and operated in the same manner as subordinate systems (recursivity). Based on the bundled knowledge about single technical systems, system descriptions can be distributed or exchanged between Energy Agents.

3.4 Energy Agents

Within the overall toolchain, Energy Agents are a conceptual element. In principle, they are software agents whose internal reasoning is based on an EOM-based system description, with an internal data model and control strategies as additional elements. The Energy Agent approach aims to cover the whole life cycle of a system, from simulations over testbed and field

3.5 Power Flow Calculation

To enable the calculation of electrical distribution grids, the toolchain comprises a power flow calculation core (PFC). Aligned with the modular architecture, different power flow algorithms can be integrated in this core, depending on the requirements of a development project. By default, a complex PFC is used, which is based on the Newton-Raphson method. This algorithm calculates all nodal voltages and branch currents of the selected grid in an efficient manner. It should be highlighted that the three electrical phases can be calculated in parallel by means of multi-threading. This capability facilitates performant testbed simulations that run in real-time. The network topology can be imported from CSV files, optionally including GIS data for geographical visualizations. The PFC takes active and reactive power as an input for each

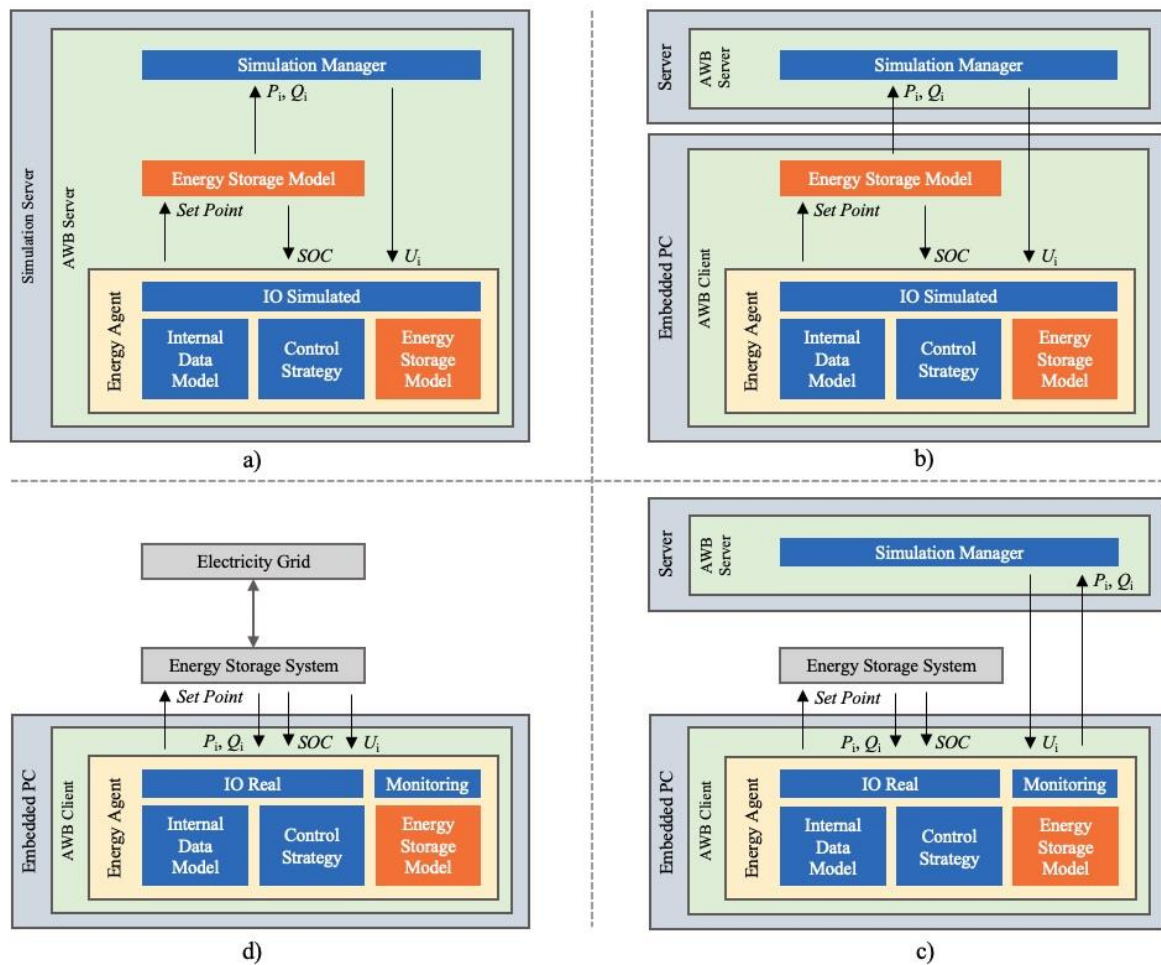


Fig. 2: System Configurations for a) Simulation, b) Distributed Simulation, c) Testbed Application and d) Field Deployment.

test scenarios to real field applications. In the field. Energy Agents are not supposed to replace the low-level control logic of technical systems, e.g. the droop control of a battery storage inverter. Instead, they act as an additional layer on top of it, providing planning and interaction capabilities for coordination with other Energy Agents and integration into control center applications (e.g. grid operation or virtual power plants).

node. These are streamed either from time series or from dynamic system models that are based on the EOM. The graphical user interface of Agent.Workbench shows a dynamic representation of the grid state with differently colored nodes and branches based on parameterized thresholds for congestions and voltage, e.g. congested power lines are displayed in red.

4. CONTINUOUS USE OF SOFTWARE ARTIFACTS

One of the key features of the toolchain is the continuous use of software artifacts, e.g. control strategies, in the development of decentralized control solutions. This is enabled by the modular architecture of the software tools, which allows the utilization of the same software artifacts within the phases *Simulation*, *Testbed Application* and *Deployment* of the 2DECS phase model. We differentiate four different system configurations for utilizing software artifacts during development, which are shown in Fig. 2. Configuration a) and b) are used in the *Simulation* phase, while c) and d) are used in *Testbed Application* and *Deployment*, respectively.

To illustrate the configurations, we make use of a simple running example. The system (hypothetically) under development is an Energy Agent that controls an energy storage system based on the observed grid voltage at the point of common coupling, i.e. it provides an autonomous grid service. While the configurations will be more complex for other applications, e.g. when multiple Energy Agents are involved, this running example covers most of the basic principles. In the following we will discuss the specifics of each configuration.

4.1 Simulation

In this configuration, which is depicted in Fig. 2 a), all components are executed on one simulation server encapsulated in a single instance of Agent.Workbench (AWB (Server)). The Energy Agent comprises four main elements. First, a model of the system it controls (in this case an energy storage), based on the EOM. This model enables the Energy Agent to evaluate different control strategies and how they affect the state and behavior of the controlled system. Second, control strategies that utilize the system model. Third, an Internal Data Model that enables parameterization of the Energy Agent. In the running example this could be the voltage limits, which trigger a change in set points. Fourth, the IO behavior for simulation environments (IOSimulated), which serves as the interface to the simulated technical system. Via this interface it sends set points to the Energy Storage Model, that actually influences the PFC contained in the Simulation Manager by generating values for active and reactive power (P_i , Q_i) for each time increment of the simulation. This model is also based on the EOM, thus illustrating the dual purpose of the EOM, i.e. internal reasoning of the Energy Agent as well as simulation of the physical system. On a side note, this separation enables analyses of simplified models and in how far they can be used for controlling complex systems. The Simulation Manager calculates the state of the grid using the previously introduced PFC. In the running example, we assume all other nodes in the simulated grid are represented by power flow time series. Via the IOSimulated the Energy Agent receives the resulting nodal voltage at the point of common coupling from the Simulation Manager.

4.2 Distributed Simulation

This configuration also corresponds to the *Simulation* phase. In this configuration, the simulation is distributed onto multiple computing devices, i.e. a server and an embedded PC. As can be seen, the Energy Agent contains the same software artifacts as in the previous configuration, however, it now runs on dedicated hardware. This configuration enables performance testing and is obviously most valuable if the Energy Agent runs on the hardware that will be deployed in the field as well. When a control solution comprises multiple entities, e.g. several Energy Agents and a control center they report to, this configuration facilitates additional analyses. For instance, by physically distributing the individual hardware entities, the effects of an actual communication network can be studied with regard to agent-to-agent and agent-to-control center communication. The communication can be either routed through the Agent.Workbench Server or via a point-to-point connection, depending on the requirements of the control solution.

4.3 Testbed Application

This configuration incorporates real systems into the simulation. It is also known as hardware-in-the-loop testing and corresponds with the Testbed Application phase of 2DECS. As with the Distributed Simulation, the Energy Agent runs on an embedded PC. However, now it sends the set points to an actual energy storage system, i.e. the energy storage is not simulated anymore. For this purpose, the IOSimulated is replaced by the IOReal behavior for interaction with real hardware. As the real storage system is not directly connected to the simulation server, the active and reactive power produced/consumed by are reported to the simulation by the Energy Agent, more specifically its monitoring component. The nodal voltage is still provided by the Simulation Manager, as we are typically interested in testing the Energy Agent against critical situations, which we can more easily realize within simulation scenarios using time series or dynamic models. The monitoring component furthermore enables the Energy Agent to write sensor values and control decision into a database.

4.4 Deployment

This configuration is applied when we actually deploy a smart grid solution. Again, there are little changes from the perspective of the Energy Agent. It now receives the nodal voltage directly from the energy storage system, which is connected to the actual electricity grid. Also, it does not forward active and reactive power of the energy storage to the Simulation Manager anymore. However, if required, such reporting mechanism to a control center can be put in place. Accordingly, only few parameter-based adjustments have to be made to the monitoring and IOReal components.

It can be seen that the core components of the Energy Agent – Internal Data Model, Control Strategies and Energy Storage Model – persist in all deployment phases. This significantly

improves engineering efficiency as it reduces redundancy, which is particularly valuable in agile development. Notably, the same principles apply to more complex control solutions as well.

5. EVALUATION

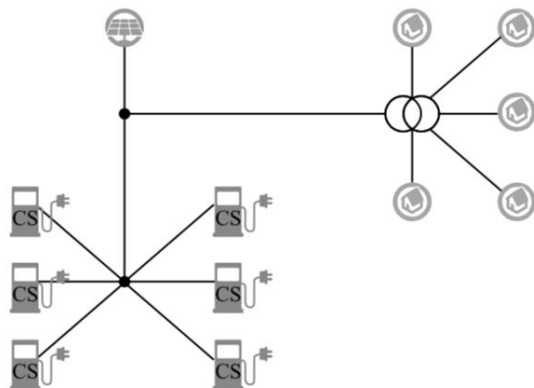


Fig. 3: Topology of the low-voltage grid.

As a means of evaluating the toolchain, it has been applied to the development of an active network management (ANM) solution for a low-voltage grid on the University of Wuppertal campus. The functional scope of the ANM solution comprises grid state estimation and prediction and the prevention of line overloads and voltage violation. The topology of the campus grid, shown in Fig. 3, includes several electric vehicle charging stations, a photovoltaic plant, a transformer station and multiple passive loads (household). To enable grid estimation, sensors (not depicted) were placed at suitable positions in the grid. The collection of sensor values and subsequent PFC is carried out by a dedicated Energy Agent. Furthermore, each of the controllable systems was equipped with its own Energy Agent that possesses a digital twin of the respective system based on the EOM. To prevent congestions and voltage violations a hybrid approach, combining decentralized coordination with centralized control in emergency situations, was used (Törsleff et al., 2017). For example, Energy Agents coordinate the curtailment of electric vehicle charging based on customer preferences. The Energy Agents were initially developed and tested in simulations. In this development stage all major issues concerning control strategies and coordination between agents were solved. Subsequently, as per the methodology, the Energy Agents were deployed in distributed simulations, testbeds and in the field on industrial PCs (Intel Atom E3845 1.91GHz CPU, 4GB of RAM, 64 GB SSD, 1504MFlops. In addition to the Ubuntu operating system, the industrial PCs were equipped with Java 1.8 and AWB, allowing for multiple Energy Agents being deployed on a single industrial PC. We iterated through multiple cycles of the 2DECS deployment phases, increasing the functional scope with each cycle. In this way code improvements from the testbed phase could easily be used in the simulation phase of the next cycle. While no systematic A-B comparison with other methodologies and development tools was performed, the project members reported considerable (subjective) gains in efficiency compared to previous development efforts in

other research projects where different development tools and methodologies were utilized. Further gains were attributed to the graphical user interface of AWB which facilitates quick modeling of different grid topologies and scenarios. Additionally, since the physical component models generated using the EOM are independent from the control logic and hence re-usable, engineering effort will decrease over time as the model library grows and increases in quality in the course of future projects. In order to validate efficiency gains resulting specifically from the 2DECS approach, it was assessed in further case studies and expert interviews. Over a period of several months 2DECS was used by two teams in three projects. Furthermore, ten experts in the field of software engineering and energy systems originating from industry and academia were interviewed. The positive results of the case studies as well as the affirmative replies of the experts support the applicability and advantages of the approach .

6. CONCLUSION AND OUTLOOK

In this contribution, we presented a toolchain for the development of agent-based smart grid solutions with a focus on how it facilitates the continuous utilization of software artifacts across the development phases. Furthermore, we gave an overview of its evaluation in the context of an active network management solution for a low-voltage grid. The successful implementation and overall potential facilitated the acquisition of new research and industry projects in which the toolchain is currently being applied and improved, ranging from sector coupling initiatives to determining hydrogen injection potentials of gas grids.

REFERENCES

- Chassin, D.P., Fuller, J.C., Djilali, N., 2014. GridLAB-D : An Agent-Based Simulation Framework for Smart Grids. *J. Appl. Math.*
- Derksen, C., Branki, C., Unland, R., 2012. A Framework for Agent-Based Simulations of Hybrid Energy Infrastructures, in: *Federated Conference on Computer Science and Information Systems (FedCSIS 2012)*. pp. 1293–1299.
- Derksen, C., Unland, R., 2016. The EOM: An adaptive energy option, state and assessment model for open hybrid energy systems, in: *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*. pp. 1507–1515.
- Gan, D., Dai, C., Li, H., Xin, H., Li, N., 2013. Virtual power plant-based distributed control strategy for multiple distributed generators. *IET Control Theory Appl.* 7, 90–98. <https://doi.org/10.1049/iet-cta.2012.0141>
- Ghribi, K., Sevestre, S., Guessoum, Z., Gil-Quijano, J., Malouche, D., Youssef, A., 2014. A survey on multi-agent management approaches in the context of intelligent energy systems, in: *International Conference on Electrical Sciences and Technologies in Maghreb (CISTEM)*.

- Godfrey, T., Mullen, S., Griffith, D.W., Golmie, N., Dugan, R.C., Rodine, C., 2010. Modeling Smart Grid Applications with Co-Simulation, in: First IEEE International Conference on Smart Grid Communications (SmartGridComm). pp. 291–296.
- Harnisch, S., Steffens, P., Thies, H.H., Cibis, K., Zdrallek, M., Lehde, B., 2016. New planning principles for low voltage networks with a high share of decentralized generation, in: CIRED Workshop 2016.
- Hashemi, S., Østergaard, J., Degner, T., Ron BrandlandWolfram Heckmann, 2015. Efficient Control of Energy Storage for Increasing the PV Hosting Capacity of LV Grids. *IEEE Trans. Smart Grid* 13, 1–9. <https://doi.org/10.1109/TSG.2016.2609892>
- Hu, J., Saleem, A., You, S., Nordström, L., Lind, M., Ostergaard, J., 2015. A multi-agent system for distribution grid congestion management with electric vehicles. *Eng. Appl. Artif. Intell.* 38, 45–58. <https://doi.org/10.1016/j.engappai.2014.10.017>
- Kok, J.K., Warmer, C.J., Kamphuis, I.G., 2005. PowerMatcher: Multi Agent Control in the Electricity Infrastructure, in: International Conference on Autonomous Agents and Multiagent Systems. pp. 75–82.
- Lehnhoff, S., Nannen, O., Rohjans, S., Schlogl, F., Dalhues, S., Robitzky, L., Hager, U., Rehtanz, C., 2015. Exchangeability of power flow simulators in smart grid co-simulations with mosaik, in: Workshop on Modeling and Simulation of Cyber-Physical Energy Systems, Held as Part of CPS Week.
- Linnenberg, T., Fay, A., 2018. Software Engineering for Agent Based Energy Systems, in: 14th IEEE International Conference on Automation Science (CASE).
- Luo, T., Dolan, M.J., Davidson, E.M., Ault, G.W., 2015. Assessment of a new constraint satisfaction-based hybrid distributed control technique for power flow management in distribution networks with generation and demand response. *IEEE Trans. Smart Grid* 6, 271–278. <https://doi.org/10.1109/TSG.2014.2327482>
- Nguyen, V.H., Besanger, Y., Tran, Q.T., Nguyen, T.L., Boudinet, C., Brandl, R., Marten, F., Markou, A., Kotsampopoulos, P., van der Meer, A.A., Guillo-Sansano, E., Lauss, G., Strasser, T.I., Heussen, K., 2017. Real-Time Simulation and Hardware-in-the-Loop Approaches for Integrating Renewable Energy Sources into Smart Grids: Challenges & Actions, in: IEEE PES Innovative Smart Grid Technologies (ISGT).
- Ochoa, L.F., Dent, C.J., Harrison, G.P., 2010. Distribution Network Capacity Assessment: Variable DG and Active Networks. *IEEE Trans. Power Syst.* 25, 87–95.
- Sawant, P., Doan, D., 2017. Modelling and Simulation of Microscale Trigeneration Systems Based on Real-Life Experimental Data. *IFAC-PapersOnLine* 50, 3238–3243.
- Schwalbe, R., Brunner, H., Stifter, M., Abart, A., Traxler, E., Radauer, M., Niederhuemer, W., 2015. DG-demonet smart LV grid-increasing hosting capacity of LV grids by extended planning and voltage control. *Proc. - 2015 Int. Symp. Smart Electr. Distrib. Syst. Technol. EDST 2015* 63–69. <https://doi.org/10.1109/SEDST.2015.7315184>
- Shawon, M.H., Muyeen, S.M., Ghosh, A., Islam, S.M., Baptista, M.S., 2019. Multi-Agent Systems in ICT Enabled Smart Grid: A Status Update on Technology Framework and Applications. *IEEE Access* 7. <https://doi.org/10.1109/access.2019.2929577>
- Thurner, L., Scheidler, A., Schäfer, F., Menke, J.-H., Dollichon, J., Meier, F., Meinecke, S., Braun, M., 2017. pandapower - an Open Source Python Tool for Convenient Modeling, Analysis and Optimization of Electric Power Systems. *IEEE Trans. Power Syst.* 33, 6510–6521.
- Törsleff, S., Linnenberg, T., Wassermann, E., Derksen, C., Loose, N., Fay, A., 2017. An Algorithm for the Temporary Acquisition of Control over Third Party Assets in Active Network Management, in: Conference on Sustainable Energy Supply and Energy Storage Systems (NEIS).
- Tuballa, M.L., Abundo, M.L., 2016. A review of the development of Smart Grid technologies. *Renew. Sustain. Energy Rev.* 59, 710–725. <https://doi.org/10.1016/j.rser.2016.01.011>
- Vasirani, M., Kota, R., Cavalcante, R.L.G., Ossowski, S., Jennings, N.R., 2013. An agent-based approach to virtual power plants of wind power generators and electric vehicles. *Smart Grid, IEEE Trans.* 4, 1314–1322.
- Vrba, P., Marik, V., Siano, P., Leitao, P., Zhabelova, G., Vyatkin, V., Strasser, T., 2014. A review of agent and service-oriented concepts applied to intelligent energy systems. *IEEE Trans. Ind. Informatics* 10, 1890–1903.