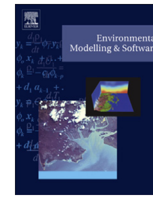
Environmental Modelling & Software 56 (2014) 37–51

Contents lists available at ScienceDirect

Environmental Modelling & Software

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



Designing dependable and sustainable Smart Grids – How to apply Algorithm Engineering to distributed control in power systems☆

Astrid Nieße [a,*], Martin Tröschel [a], Michael Sonnenschein [b]

[a] OFFIS – Institute for Information Technology, Division Energy, Escherweg 2, D-26121 Oldenburg, Germany
[b] Department of Environmental Computer Science, University of Oldenburg, D-26111 Oldenburg, Germany

ARTICLE INFO

Article history:
Received 21 May 2013
Received in revised form
2 December 2013
Accepted 2 December 2013
Available online 6 January 2014

Keywords:
Smart grid
Distributed algorithms
Smart grid simulation
Scenario design
Information technology
Process model

ABSTRACT

In this work, we present the Smart Grid Algorithm Engineering (SGAE) process model for application-oriented research and development in information and communication technology (ICT) for power systems. The SGAE process model is motivated by the main objective of contributing application-oriented research results for distributed control concepts on a sound methodological background. With this process model, we strive for an engineering aspiration within the domain of Smart Grids. The process model is set up with an initial conceptualisation phase followed by an iterable cycle of five phases with both analytical and experimental parts, giving detailed information on inputs and results for each phase and identifying the needed actors for each phase. Simulation of large-scale Smart Grid scenarios is a core component of SGAE. We therefore elaborate on tooling and techniques needed in that context and illustrate the whole process model using an application example from a finished research and development project.

1. Software availability

The Smart Grid simulation framework mosaik is available for researchers within their PhD projects on request. More details are given at http://mosaik.offis.de.

2. Introduction

Smart Grids are expected to enable flexible, accessible, reliable and economically attractive electricity networks (SmartGrids European Technology Platform, 2010). Following this definition, control systems for Smart Grids should additionally enable the integration of high shares of renewable energy resources and thus contribute to a sustainable transformation of the power system, taking into account the effects of information and communication technology (ICT) itself (Hilty et al., 2006). The transformation of the existing power generation to renewable, distributed generation implicates an increase in complexity for the control of the overall system, as control methods have to cope with many individually configured, distributed, small generation units as well as with fluctuation in their feed-in depending on meteorological conditions.

Distributed control methods, such as self-organizing multi-agent systems, are a very promising approach to address both technical (flexibility, accessibility, reliability) and economic requirements for Smart Grids relying on distributed generation and demand-side management (Kok et al., 2005), (Penya and Jennings, 2008), (Lehnhoff, 2010), (Ramchurn et al., 2011), (Nieβe et al., 2012). Although some approaches are already under test in field trials, distributed control methods for Smart Grids are still subject to application-oriented research, not (commercial) software development. To facilitate the transition to the field however, a methodologically sound engineering process model that guides this way from the beginning is essential.

Smart Grids up to now lack such a standardised process model, but developments regarding standardisation are under way: with the Smart Grid Architectural Model (SGAM) a first conceptual framework has been defined in 2008, including a use case based methodology to manage the requirements engineering process (International Electrotechnical Commission, 2008), (Trefke et al., 2013). Use cases as a well established instrument from software engineering describe a system's intended behaviour, defining actors and system interactions as sequences of actions with the required intermediate results. The use case based methodology

☆ Thematic issue on Modelling and evaluating the sustainability of smart solutions.

* Corresponding author.
E-mail addresses: niesse@offis.de (A. Nieße), troeschel@offis.de (M. Tröschel), sonnenschein@informatik.uni-oldenburg.de (M. Sonnenschein).

1364-8152/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.envsoft.2013.12.003

maps a use case to the SGAM in several steps starting from a functional view, yielding refined requirements on the system to be developed for each layer. This model focuses on application development, with the involved parties being aware of the system's intended behaviour. As this does not hold for application-oriented research, this methodology is not applicable: for complex systems like Smart Grids, a process model is needed in order to actually define the systems' intended behaviour in a preliminary phase.

Within application-oriented research as a preliminary step to (commercial) software development, process models are less common than in commercial software development. A widespread process model in information technology is design science (Hevner et al., 2004). Design science is a general framework for the design and evaluation of any artefact in an arbitrary application domain. It introduces explicit steps for the description of

- the artefact with respect to a given research question,
- a design process (search heuristics),
- a grounding in a knowledge base,
- the type of evaluation,
- the kind of introduction of the artefact into the real world,
- the input to the knowledge base.

For specific artefacts at least the actual design process as well as the intended type of evaluation have to be refined. If the artefact is an algorithm, Algorithm Engineering (Sanders, 2009) as a process model for scientific algorithm development can be applied. Algorithm Engineering is a general model for the theoretical research on and development of algorithms, focussing on performance and complexity of algorithms. It is explicitely invariant to the application domain and therefore does not yield application-specific performance information. We therefore claim the need for a domain-specific extension of Algorithm Engineering for researching on and developing of Smart Grid control concepts – Smart Grid Algorithm Engineering. We derived the relevant topics to be considered for this extension from our experiences during many years of work in this field. While we do not imply any claim for comprehensiveness, we aim at developing a process model that solves the most relevant issues regarding application-oriented research on Smart Grid control algorithms. Therefore, we understand a later refinement of the process model itself as an iterative process. The list of topics considered for an extension of Algorithm Engineering comprises:

- *Smart Grid knowledge:* We aim at developing control concepts for the field and therefore have to be aware of the relevant legal and technical constraints regarding the real-world implementation from the beginning.
- *Dependability analysis*: The Smart Grid is a mission-critical system. We therefore have to consider dependable system behaviour, e.g. defined runtime behaviour, a priori.
- *Simulation models*: As we cannot perform experiments in the field for mission-critical systems like Smart Grids, the evaluation of control concepts under development can be performed only based on (large-scale) simulation studies.
- *Scenario design and usage*: We want to consider the applicability and domain-specific performance of control concepts in different contexts, represented by different scenario designs during simulation.
- *Knowledge management*: With personnel in research institutes and universities usually leaving to academia or industry after some years and new scientists following from universities (usually with disciplinary knowledge from the application domain), knowledge management is a big issue to achieve continuity.

In this paper, we therefore propose a domain-specific extension and refinement of Algorithm Engineering – a methodical combination of algorithmic research and engineering – as an engineering approach for the design of ICT-based control in power systems. First, we derive some key requirements for a suitable engineering approach regarding the above defined topics using the Smart Grid Architecture Model (SGAM). Following that, we discuss simulation as a key component especially for an experimental assessment of Smart Grid control systems, and we give some methodical background on the design of both Smart Grid scenarios and simulation-based experiments. We then introduce the concept of Algorithm Engineering and elaborate on our domain-specific extension, Smart Grid Algorithm Engineering. We conclude this paper with a discussion on the benefits of our approach, problems encountered in using this model and necessary extensions.

3. Requirements for a distributed Smart Grid control algorithms engineering approach

We rely on the Smart Grid Architecture Model (SGAM) as a reference design for Smart Grid systems to derive general requirements on how to adapt Algorithm Engineering to the Smart Grid domain (see Fig. 1). We first give an overview on the SGAM, after that map relevant topics within the research area of Smart Grid control algorithms to the layered view of this conceptual model and then derive requirements on the process model from this view.

The main issue addressed by the SGAM is interoperability of software and automation systems from business applications down to components in the field, each with specific interface requirements. The SGAM has been introduced by three European standardisation organisations (CEN, CENELEC, ETSI) as a result from the European standardisation mandate M/490. Volunteers from industry and manufacturers have been heavily involved in the process of defining the SGAM. Their common interest is to setup a reference architecture that facilitates the development of interoperable component interfaces, thus reducing costs and engineering overhead after deployment to the field. As already pointed out in the introduction, the SGAM can be regarded as a first step towards a standardised process model for system development in the Smart Grid.

On the domain dimension in the SGAM, the energy conversion chain from bulk generation down to the customer premises is depicted (see Fig. 1). The domain of distributed energy resources (DER) on the distribution level is integrated in this dimension. The management systems for each level form the second dimension of different zones, emphasising the different hardware, IT systems and actors involved from market down to field and process zone. In the station zone for the distribution grid domain for example, automation and protection systems in substations would be allocated. The plain formed by these two dimensions, domains and zones, is combined with the different abstraction levels from business level down to the communication and component layer as interoperability dimension. The different layers represent rising interoperability requirements for applications crossing the layer boundaries: communication with components in the field is a basic functionality for Smart Grid applications, thus it can be found on the lower component and communication layers. Modelling of information using standardised data models is a precondition of higher business functionality depicted in the upper levels, therefore the information layer can be found inbetween these layers (International Electrotechnical Commission, 2008). IT-based voltage control at a substation may serve as an example: depending on the current operational state of the underlying power grid, measurements in the connected higher voltage level grid and prognoses for both, a

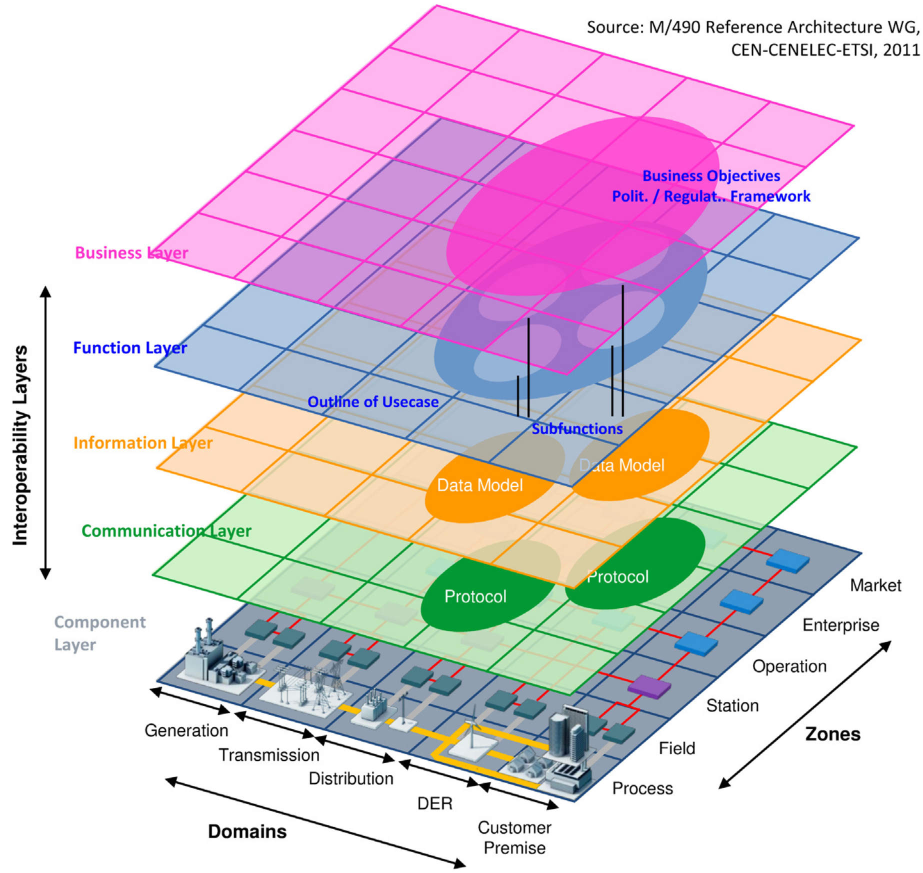


Fig. 1. Smart grid architectural model (SGAM), Source: International Electrotechnical Commission, 2008.

tap changer may be controlled to influence the voltage level in a proactive manner. This kind of functionality depends on data exchange and interoperability including common semantics. Therefore it would be placed in the function layer of the SGAM. Smart Grid applications may span all domains and zones. Distributed control algorithms are typically developed for the domains from customer premises to distribution, thus including the DER domain, and are allocated at either the function or the business layer.

In Table 1, we map relevant topics for the engineering of Smart Grid control algorithms to the layered view of the SGAM conceptual model, thus deriving general requirements on the process model.

4. Modular Smart Grid simulation

Simulation is an important tool for understanding the complex interactions between the interconnected elements of Smart Grids and the evaluation of efficient coordination and control strategies. In addition to real-world experimentation and theoretical analysis, simulation offers the opportunity to study the behaviour of future power systems under varying conditions and in numerous scenarios. Here, we understand a scenario as a specific instance of the SGAM component layer (a given power grid infrastructure with a certain share of distributed generators, consumers and storage systems), and an experiment as a parameter assignment for all components within the scenario followed by a computational execution of this scenario using simulation tools. In the following, we will discuss the modular Smart Grid simulation framework mosaik as one tooling example for the simulation-based evaluation of Smart Grid control approaches. It should be noted, however, that there are many possible tools available that could be used in the context of Smart Grid simulation (European Energy Research Alliance, 2013), (Pöchacker et al., 2013), e.g. GridLab-D (Chassin and Widergren, 2009), IPSYS (Bindner et al., 2004) or HOMER (HOMER Energy LLC, 2013). An appropriate tooling decision always depends on the specific requirements in a given research context – our engineering approach is not limited to the usage of mosaik.

4.1. Mosaik

In the context of Smart Grids, a lot of effort is put into the modelling of (electro)technical system components such as photovoltaics, wind energy converters or entire power grids. Depending on the purpose of the simulation, the models' properties such as temporal resolution, technical fidelity, computational complexity etc. can vary greatly. Often, models are evaluated at large costs with data from laboratory experiments and/or field tests. In this context validation means that the model's output fits the measured data – usually voltage, real and reactive power – for a given range of input parameters and a given temporal resolution. For components of Smart Grids the latter is highly important, because physical units can be modelled and observed with respect to different electrical effects like short-term oscillations or mean power production depending on the temporal resolution. Thus a component model is validated independently from a scenario for a certain application area depending on the modelled effects.

Ideally, a publicly available library of such validated and trusted component models would be established in a similar way as it exists e.g. for ecological models by the web-based ecobas library (Benz et al., 2001). De facto, detailed models of specific components as wind energy converters or CHPs of a specific type are not

Table 1
Deducing process model requirements from SGAM.

Topic	**Smart grid knowledge**
SGAM view	ICT-based Smart Grid control concepts span several zones in the SGAM. The usage of data between zones may be restricted to specific actors, as described in the ENTSO-E harmonized electricity role model (ENTSO-E, 2011). The development of algorithms has to cope with these restrictions. If, for example, the system operation of power networks should include state information of units connected to the grid, the data access might be restricted for unbundling reasons.
Requirement 1	Ensure domain knowledge in algorithm development: The process model should help to avoid conflicts of developed algorithms with regulatory or legal requirements.
Topic	**Dependability analysis**
SGAM view	All zones from operation to process zone are mission-critical for a secure operation of the electrical system regarding operational constraints of automation equipment, generators and controllable loads and grid components like transformers or tap changers. For example, the development of a new ICT-based configuration concept for protection equipment in low-voltage grids would require a systematical check if the real-time constraints of this application area are still met.
Requirement 2	Reflect dependable system behaviour: The process model should include a dedicated phase where checks regarding dependable systems behaviour and real-time constraints may be performed.
Topic	**Simulation models**
SGAM view	In an experimental evaluation, simulation models represent real hardware on the component layer. In the Smart Grid domain, this is of tremendous importance, as for financial and safety reasons tests in the field can be performed only very late in the development process. If co-simulation is performed, simulation models to represent the communication layer are needed as well, e.g. to compare the performance of different communication protocols.
Requirement 3	Use validated simulation models: Validated simulation models (e.g. established power flow analysis models) have to be used. The process model should stipulate the documentation of developed models to facilitate a well-grounded choice of validated models regarding e.g. temporal resolution and interfaces.
Topic	**Scenario design and usage**
SGAM view	A concrete simulation scenario can be considered an instantiation of the component dimension. For co-simulation experiments, this scenario includes the communication layer as well. Many assumptions have to be made to narrow down the huge amount of possible parameter settings to specified and ready-for-evaluation scenarios. Therefore, one concrete scenario cannot represent the component dimension in it's full variety. As an example the variety of grid topologies should be mentioned: to set up a scenario, this variety can be narrowed down to typical structures like radial distribution systems and ring topologies, but concrete instances have to define all attributes down to line capacities. The experimental results derived with this grid instance thus loose generality regarding all possible grid topologies. Therefore a variety of scenarios have to be checked to expand the validity of the experimental results.
Requirement 4	Provide scenario interchangeability: The simulation framework used within the process model has to support a convenient approach for scenario interchangeability to avoid the pitfalls of overfitting the algorithms to a specified scenario.
Topic	**Knowledge management**
SGAM view	The development of Smart Grid algorithms spans all layers in the SGAM from the function to the component layer. The domain-specific performance of algorithms usually cannot be estimated in advance, but comparison with related problems may help to classify the problem and identify first solutions. For example, many energy scheduling problems can be considered as combinatorial optimisation problems. Some may be solved by branch-and-bound algorithms, whereas others will show performance issues following this approach. The documentation of the results regarding a concrete application might help researchers to identify appropriate algorithms for related problems.
Requirement 5	Document evaluation results including meanders: In research projects, especially the process of meandering to reach an appropriate algorithm for a defined problem is usually not documented, whereas the final results (algorithms showing good performance and interesting domain-specific results) are published in scientific journals or conference proceedings. For the scientific personnel though, the meandering process is most important to gain knowledge for future problems. Therefore the process model should facilitate the documentation of intermediate results and meanders.

disclosed by the manufacturer, so generic models often have to be used instead. For the evaluation of generally applicable smart grid control algorithms this is already a common approach. Each model – whether specific or generic – has to be documented along with its intended application area, information on its validation by data from real components, temporal resolution, and information on its composability, as specified in the next paragraph. Thus, scenarios composed from these models rely on a continuously expanding foundation of evaluated and trusted simulation models.

In Smart Grid simulation, modularity – the approach to build complex systems from single, often standardized building blocks or sub-systems – is a key feature to allow the creation of complex, large-scale simulations by reusing evaluated and trusted component models. The simulation framework mosaik relies on formal descriptions of both the syntactic and semantic properties of single simulation models and the Smart Grid scenario to compose large-scale simulations. The semantic information on single simulation models includes, in particular, the step size of the simulator, available parameters, data types and units of the parameters, allowed parameter ranges, as well as information on required and offered data flow. The mosaik API automatically checks if component models are used consistently to this information in a scenario. As depicted in Fig. 2, a scenario model (1) comprising semantic self-descriptions of different simulation models and their runtime environments (simulators) has to be specified in a domain-specific language using Eclipse Xtext (Schütte and Sonnenschein, 2012). Control strategies realised e.g. as multi-agent systems (MAS) are taken into account by adding their self-description as part of the scenario model. During runtime, the formal scenario model is compiled into a simulation study description (2) that specifies all simulated system components and their input/output-interrelations (Schütte et al., 2011). The mosaik composition and

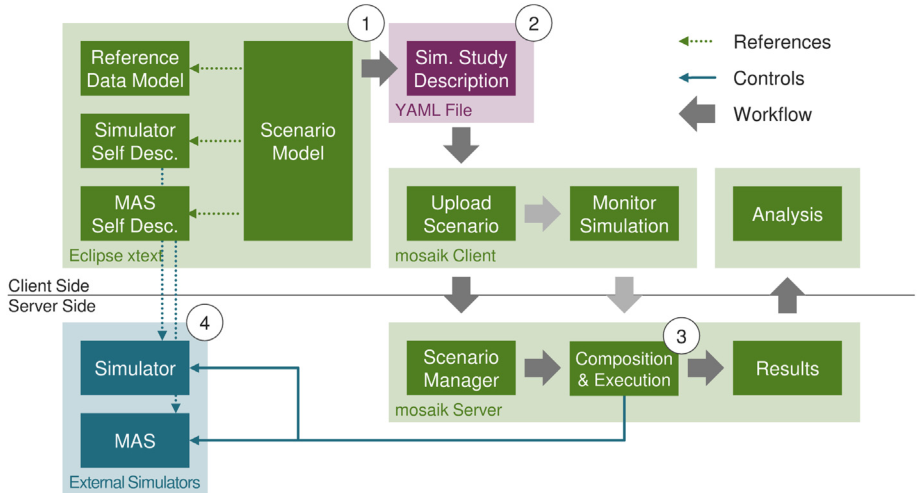


Fig. 2. Architecture overview of mosaik (Scherfke and Schütte, 2013).

execution engine (3) then uses this information in order to instantiate, parameterize and execute the different simulators (4), to synchronize their simulation progress and to exchange input/output data as necessary. In our current projects, we focus on studies in the frequency domain, that is we restrict the usage of mosaik and the corresponding simulation models research questions regarding phenomena with a temporal resolution larger or equal to 1s.

Using formal scenario models for the specification of Smart Grid simulations has the additional advantage of reproducibility of results and ease of scenario variation. As all used simulation models, their interrelations and their scenario specific parameters are documented in a formal way. By this means, simulation experiments can be executed automatically as often as desired to generate reliable data for empirical analyses. Also, varying single scenario parameters – e.g. the share of energy storages in a given distribution grid – is inexpensive and allows for the comparison of different instances of a common base scenario.

4.2. Scenario design

Scenarios in the domain of distributed Smart Grid control are needed to strengthen or falsify hypotheses regarding the effectiveness or optimality of control concepts in current and future grids. Thus, they serve as input for technology assessment or integrated assessment and modelling (IAM) (Harris, 2002). Scenarios therefore are a key element within the inductive-experimental approach of application-oriented research (Gibbons et al., 1994). In Smart Grid research and development projects, scenarios comprise the power grid with all elements needed to evaluate the developed algorithms: the energy generation units with their grid connecting node, their electrical feed-in behaviour over the examined time span and all electrical loads within the system – thus they can be understood as an instantiation of the component layer within the SGAM. For reference scenarios representing a current system development state, statistical data can be retrieved from statistical offices and industry information. For future projections, data retrieval is much more challenging and quality issues have to be taken into account. In Smart Grid research, complete scenario projects, e.g. following (Alcamo and Ribeiro, 2001), (Gausemeier et al., 2009) or (Spath et al., 2011) are often not feasible due to the high amount of relevant technologies from grid components in the relevant voltage levels to automation and ICT equipment. Current studies may instead deliver appropriate inputs for future scenario design. Assumptions regarding future trends may conflict; therefore these studies have to be combined carefully during scenario design. The decision of the appropriate scenario approach should depend on the technological scope and data availability for the addressed topics: for projects with a limited technological scope and missing surveys on future projections, scenario projects following (Alcamo and Ribeiro, 2001) or (Gausemeier et al., 2009) might be feasible and/or necessary. For projects with a broader Smart Grid context the use of publicly available data sources and surveys is advisable. In Table 2 an excerpt of relevant data sources for current and future Smart Grid scenarios for Germany is given. We used these data sources in the GridSurfer project and several other projects, carefully evaluating assumptions and updates.

4.3. Design of experiments

A statistically sound experimental design is crucial for the interpretation of data resulting from simulation, especially regarding the avoidance of optimistic or pessimistic special cases in the context of empirical analyses. As experimental design is a vast methodological field, a comprehensive discussion is not possible at this point (see

Table 2
Data sources for reference and future scenarios (Germany).

	Reference scenarios	Future scenarios
Power grids	Example grids from TSO/DSO partners, benchmark grids from CIGRE or IEEE	Grid development plan (50Hertz Transmission GmbH, 2013), survey on distribution grids (Deutsche Energie-Agentur GmbH, 2012)
Renewables	Germany renewable energy database (50Hertz Transmission GmbH, 2011)	Future scenario of the German Federal Ministry for the Environment, Nature Conversation and Nuclear Safety (BMU, 2010), energy concept of the German Federal Government (Bundesregierung, 2010)
Electrical storage	Survey on storage technologies and needs 2(Energietechnische Gesellschaft im VDE, 2009)	Surveys on future electrical storage needs and technologies (Energietechnische Gesellschaft im VDE, 2012; Hollinger et al., 2013)
Controllable loads	SmartA survey (Stamminger et al., 2008)	See reference scenarios
ICT connectivity	Survey on future emerging Smart Grid with ICT focus (Appelrath et al., 2012)	See reference scenarios

(Kleijnen, 2001), (Kelton, 2000), and (Oh et al., 2009), for an introduction to this field). We will, however, outline the fundamental methodical approach for the design of experiments following the discussion in Siebertz et al. (2010). An important goal of designing simulation experiments is the definition of a set of test series that maximizes the information value of the data obtained by a minimized number of typically time-consuming simulation runs. Here, a test series is a fixed assignment of values to input parameters for a simulation run, and an experimental design is a set of test series, accordingly. The methodical approach for the creation of sound experimental designs comprises the following four steps (adapted following (Siebertz et al., 2010)):

1. Model analysis: The inputs and outputs of the system under consideration have to be determined, documented and – in case of input parameters – classified by relevance. Screening and, if applicable, sensitivity analysis can support this step.
2. Define parameters and range of values: The relevant input parameters and the according range of values have to be defined. If necessary, the possible ranges of values have to be discretized. It should be noted that the complexity of the resulting experimental design increases significantly with the number of allowed values and the number of input parameters.
3. Create experimental design: Simulation experiments are defined by assigning discrete values to input parameters. Depending on the characteristics of the system under consideration, several different approaches are possible: uniformly distributed experimental designs require that all discrete values of an input parameter are used equally often in the test series, orthogonal designs facilitate the efficient analysis of systems with independent inputs, and random-based design such as Latin Hypercubes or Monte-Carlo-simulations are based on randomised value assignments.
4. Optimise experimental design: After the initial setup, an experimental design's quality can be evaluated and if necessary improved. Possible quality measures are for instance entropy as a measure of the amount of information in a set of test series, or discrepancy as a measure related to the probability distribution of the discrete values in a set of test series. Again, there are several approaches for optimisation in this step.

These steps should lead to a significant reduction of the exploration space of data necessary for evaluation purposes.

5. Algorithm engineering

Since Popper presented a scientific method combined of deductive analyses and inductive experimental evaluation of theories (Popper, 1959), many disciplines have adopted his idea within their concept of analytics and experiments. For algorithmics, Sanders proposed Algorithm Engineering as a cycle of design, analysis, implementation and experiments as illustrated in Fig. 3 (Sanders, 2009). The main purpose of Algorithm Engineering is to yield application-independent insights in the general performance of algorithms, depicted at the lower left of Fig. 3 as performance guarantees. Although Algorithm Engineering claims to be application-independent, input from real-world problems is used to set up realistic models and experimental inputs. By this means, relevant algorithms for specific application domains should be developed.

Sanders' general methodology shows an innovative path for profound algorithmic development inspired by application-specific problems. For the development of distributed control concepts in the Smart Grid though, general performance of algorithms regarding runtime behaviour and storage requirements is of minor relevance: to give an answer to real-world problems within power grid operation and energy unit coordination, insight into Smart Grid specific performance indicators (PI: performance indicator) has to be won. Therefore, we propose an application-specific extension of Algorithm Engineering for the development of distributed control concepts in the Smart Grid.

6. Smart Grid Algorithm Engineering (SGAE) – an integrated approach

When Smart Grid research is performed with an engineering aspiration, people and processes matter. In the following, we first describe relevant roles in our concept of Smart Grid Algorithm Engineering (SGAE) and then give more details on the proposed process.

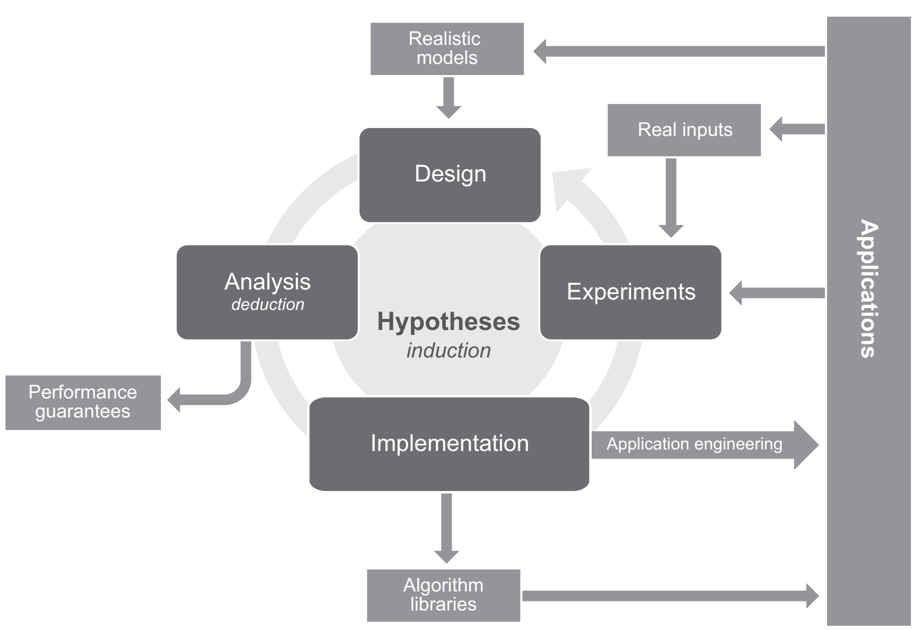


Fig. 3. Algorithm Engineering, adapted from (Sanders, 2009).

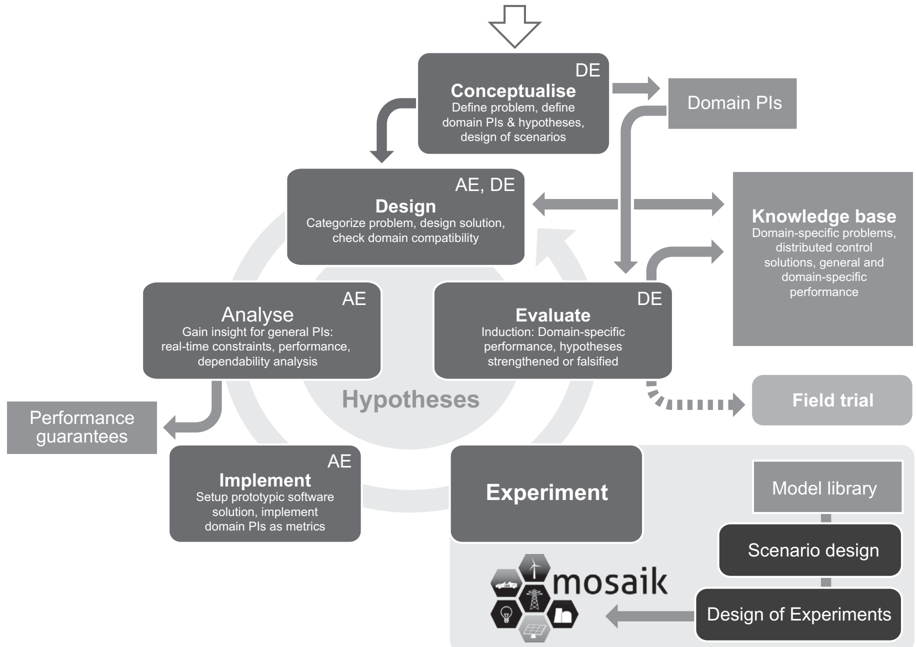


Fig. 4. Smart Grid Algorithm Engineering process model (SGAE).

6.1. Roles in SGAE

As an application-oriented process model for research and development, we have to take a closer look at the roles needed within the process, and the premises (e.g. academic background) under which the scientific personnel can fulfil them. As the main purpose is to develop distributed control concepts, we need personnel with a profound background in algorithms and the ability to detect an algorithmic categorisation of real-world problems. As this is work within a practical application context, we call these people **algorithm engineers** (AE); most of them probably with a background in computer science.[1] AEs are not expected to fully understand the application domain: the **domain experts** (DE) come with an expertise in the context of the problem to be solved. The SGAM presented in Section 3 (see Fig. 1) helps to identify the needed background: if our solution crosses all zones from market to process in the field and all domains from bulk generation to customer premises, it will be nearly impossible to find a DE with sufficient knowledge in all the areas.[2] DEs may be energy market experts, DER experts, power system engineers, electrical engineers, etc. Additionally we claim the need for **experts in experimental engineering** (EE) with a background in model building, simulation, experimental design and statistics. In the following sections we will point out how the roles defined here interact in the process model of SGAE.

6.2. Overview

To develop a process model for Smart Grid control concept research and development, we modify the Algorithm Engineering process cycle following the requirements defined from the SGAM conceptual model (see page 5) in the following manner:

1. We add a conceptualisation phase strongly related to the application domain, where we understand and describe the problem to be solved, thus yielding the realistic models used in Algorithm Engineering. Furthermore, we add Smart Grid background to the deductive phase of design and analysis (requirement 1: *ensure domain knowledge in algorithm development*).
2. With Smart Grid knowledge inserted into the analysis phase, we add an early check for admissibility of the designed solution to the application domain and gain insight in performance guarantees relevant to the application area (requirement 2: *reflect dependable system behaviour*).
3. We set up a model library for component simulation models to improve both model quality and comparability of experimental results (requirement 3: *use validated simulation models*).
4. We make extensive use of Smart Grid simulation frameworks and models to reduce the repeated overhead of simulation and scenario setup (requirement 4: *provide scenario interchangeability*) and split the experimental phase from algorithm engineering into an experimental and separate evaluation phase, stressing the different roles within the process.
5. In the evaluation phase, we demand an appropriate documentation of results including those experimental efforts that did not show the expected outcome, as an important aspect of knowledge management (requirement 5: *document evaluation results including meanders*).

Fig. 4 gives an overview of the resulting process model of Smart Grid Algorithm Engineering (SGAE). We explain the details for each phase in the following sections. To illustrate the process model we refer to the project GridSurfer, a research and development project funded by the German Federal Ministry of Economics and Technology (BMWi) from 2009 to 2011 within the context of e-mobility (Tröschel et al., 2011). This is a convenient example, as the

[1] Unlike in general software development processes, we do not differentiate requirements engineers, software developers and so an, but when we iterate from prototype to practice this has to be done.

[2] This is also a plea for a thematic restriction of the problem to be solved – if we do not have the domain experts needed to understand and conceptualise the problem, we will not be able to generate and evaluate an appropriate solution.

Table 3
SGAE phase conceptualisation.

Objective	The objective of this phase is to define the problem to be solved within the application domain and thereby prepare the topic for non-domain experts. The hypotheses are specified and the scenarios to evaluate the performance regarding application-specific performance indicators are chosen.
Input	Industrial partners problem definition (if applicable/possible)
Roles	DE, AE (for review of problem definition)
Procedure	*Problem definition:* In this initial step, the DEs have to define a problem relevant to the application area in such a way that the AEs are able to classify the algorithmic properties of the problem. We learnt in several projects, that this step is both time-consuming and demanding for researchers. Algorithm engineers have to be involved to review the problem definition to ensure that the results are unambiguous within their understanding. *Define domain performance indicators:* With precise knowledge of the problem to be solved the domain experts define which domain-specific performance indicators have to be taken into account to measure the performance of the solution later in the process. *Identify hard constraints:* If the algorithm to be designed is related to system critical aspects within the Smart Grid, some constraints like operational constraints regarding voltage or frequency have to be regarded as hard constraints. An experimental approach can not be used to guarantee that these will be within the desired bounds. Therefore, these hard constraints are marked for examination within the analytical phase of the SGAE. *Define hypotheses:* From problem definition and domain-specific performance indicators, first hypotheses regarding the expected outcome of the solution are defined. They represent the backbone of the whole SGAE cycle, thus a careful definition and refinement in later phases or repeated cycle runs is crucial. Additionally it is checked if the hypotheses can be falsified or strengthened by the domain PIs. Otherwise either domain PIs or hypotheses (or both) have to be refined. *Design scenarios:* To evaluate the performance of the solution regarding the application problem, instantiations of the component layer are needed, defining the power grid details down to the lines and transformers with all energy components connected. Further, assumptions regarding the controllability and expected growth rates of DER for future scenarios have to be made. Details on scenario design are given in Section 4.2.[a] Referring to requirement 4 it is an important issue to define an adequately broad spectrum of scenarios to avoid an overfitting of control algorithms (cf. Section 4.2). This is supported by our mosaik simulation framework offering a Smart Grid specific scenario definition language allowing to file scenario definitions and share them between different projects (cf. Section 4.1).
Output	• Problem definition • Domain PIs • Hard constraints for analytical examination • Set of hypotheses • Set of scenarios

[a] In Fig. 4 the scenario design is shown as part of the experimental phase to stress the importance of the activities closely related to Smart Grid simulation. The initial scenario design has to be performed within the conceptualisation phase. In the experimental phase, the scenarios have to be implemented within the chosen simulation framework.

technological setting and the designed algorithm itself is quite simple.[3] Currently, we apply the full process of Smart Grid Algorithm Engineering within the much broader project group Smart Nord (Sonnenschein et al., 2012).

6.3. Conceptualization and problem definition

We have already emphasised the role of domain experts: they are needed to determine current and expected problems in the power systems domain. Thus, they enable algorithm engineers to design a solution for real-world problems with the right assumptions. Therefore we add a conceptualisation phase to stress the importance of a thorough problem definition and hypotheses generation, thus reflecting requirement 1 (*ensure domain knowledge in algorithm development*). The domain experts have to define under which assumptions (manifested in the scenarios) the solution designed later on should perform well and which application-specific performance metrics are needed to measure this performance.

6.3.1. Details

See Table 3.

6.3.2. Example

With a high share of photovoltaic units (PV) in the distribution grid, problems regarding line capacities and transformer usage are expected. With a high share of electric vehicles, additionally a high simultaneity of charging might for example result from commuters plugging their cars when returning home from work. This is expected to lead to severe load peaks during the typical peak hours in the evening. We derived the following general hypotheses from the problem description and domain knowledge[4]:

- With a high share of both PV and EV a smart charging strategy can help to use the PV feed-in in the local grid, thus reducing problems in the grid.
- More PV feed-in will be used locally and less grid-operational constraints are violated when both charging and discharging is controlled.

From these hypotheses the following domain performance indicators where identified, thus representing requirements for the experimental phase to generate data sets reflecting these performance indicators.

- Probability density function of the transformer load states
- Overall share of PV feed-in in local sinks
- PV share used for charging the electric vehicles

6.3.3. Extensions

Up to now, we have not followed a formal process in defining the problem but relied on discussion and review within the project team. The quality of the outcome therefore strongly depends on the ability of the scientists involved to overview and structure the problem. We already showed that the SGAM is a useful

[3] It should not be underestimated that a restricted technological setting does not simplify the process of scenario design, experimental design and evaluation.

[4] These hypotheses were refined within the process. For reasons of clarity only the initial hypotheses are presented.

conceptualisation for ICT related Smart Grid topics. In the IntelliGrid methodology for developing requirements for energy systems (International Electrotechnical Commission, 2008) a use case methodology based on the SGAM is presented. This approach can be applied to requirements engineering in the development process of distributed Smart Grid algorithms (Lehnhoff et al., 2013). We will follow this concept in upcoming research projects. The results of the problem definition are currently documented in a project-specific non-formal document style. The use of ontologies as a formal representation of problem definitions has been proposed (Janssen et al., 2009). We will check this concept for applicability within SGAE.

6.4. Design of (distributed) SG control schemes

In this second phase, algorithm engineers are challenged to find an adequate distributed control algorithm for the specified problem. This is the most creative part of the process, but here creativity does not mean to invent anything from scratch, but to find the right, tailor-made solution for specific requirements based on a pool of known general algorithmic solutions. Compared to the original AE approach, this phase again contains a collaboration with domain experts as the demanded control algorithms have to fit into the application domain.

6.4.1. Details

See Table 4.

6.4.2. Example

For the smart charging of EVs we decided to allocate the intelligence at the local network substation, i.e. the local transformer. This reduces the complexity of the problem, because the number of vehicles connected in the low voltage grid below the local transformer station is quite small (typically <100). Additionally, the hierarchical topology of the power grid allows a quite efficient coordination between different voltage levels. The local decision on the optimal charging and possible discharging interval for a vehicle is based on the one hand on the expected charging level and parking duration, both specified by the driver, and on the other hand on the expected residual load in the local grid. For the calculation of the residual load, a supply forecast is given every 24 h by the PV systems in the local grid. The calculation of charging intervals uses this information to minimise the difference between local supply and electrical loads (demand of EVs and other consumers) for all time slots (Vornberger et al., 2011). A review of this approach with experts from the domain, in this case a distribution grid operator and owners of the EVs, came to the result that the requirements of this approach were realistic – particularly a 24 h-prognosis of the residual load seemed to be acceptable.

We compared three different charging strategies:

- *Uncontrolled charging*: Directly after being connected to the charging point, EVs charge until their batteries are fully loaded.
- *Controlled charging*: Using a smart charging approach discussed in Vornberger et al. (2011), EVs receive individual charging times in order to distribute their power demand over time, thus reducing simultaneity of charging processes.
- *Vehicle-to-grid*: In addition to controlled charging, EVs were used to feed electric power back to the grid in times of high demand, thus acting as a mobile storage system for electric energy (Tröschel et al., 2011).

6.4.3. Extensions

The use case methodology already presented in Section 3 could be used to check whether the algorithm fits the problem. If the proposed algorithm is a self-organizing method, general criteria of self-organizing methods (autonomy, emergence, global state awareness, target orientation, adaptivity, resilience) should be identified (Holzer et al., 2008), (Holzer and de Meer, 2009). In addition to the domain-specific PI, such criteria could be used to characterise the self-organisation properties of the algorithms.

6.5. Analysis

In the original AE approach, this phase is dedicated to the formal runtime analysis of the algorithm. In the domain-specific SGAE approach, specific PIs indicate whether the algorithm fulfills its

Table 4
SGAE phase design.

Objective	This phase has several objectives: • Identify the algorithmic complexity (e.g. NP-hardness) and the proper class (e.g. DCOP) of the specified problem, • Develop a possible algorithmic solution for the problem, • Check whether the proposed algorithm is adequate for the application domain and whether it is based on the right assumptions on its technical environment.
Input	• Problem definition (conceptualisation phase) • Set of scenarios (conceptualisation phase)
Roles	AE, DE (for reviewing)
Procedure	*Identification:* For a proper identification of the problem class, similarity to problems in other domains, abstract problem definitions (Garey and Johnson, 1979), as well as application-specific problem classification schemes (known e.g. for some operational research problems) can be helpful, but personal experience of AEs is indispensable. *Development:* In many cases, new algorithms can be based on known solutions for similar problems of the proper class, or on generalised algorithm schemes. Generalised algorithm schemes are well known in self-organisation methods (e.g. ant colony algorithms, schooling algorithms) or in computational intelligence methods (e.g. evolutionary algorithms) for optimisation purposes. Several of these methods are explained in Gendreau and Potvin (2010). Additionally, a number of basic distributed algorithms (e.g. termination detection, snapshot) (Lynch, 1996) can be used as part of a specific solution in distributed control. For a domain-specific algorithm, intended applications have to be kept in mind. *Check for applicability:* Domain experts have to assess whether the proposed solution fits the problem definition and could be implemented in practice. In particular, they assess whether the requirements of the algorithm on the availability of metered data, communication paths and computational power at devices could be realised with reasonable efforts. Furthermore, they check information availability regarding the respective roles within the energy system (ENTSO-E, 2011). If this check fails, the algorithm has to be redesigned, so the design phase will not end before this check is successful.
Output	• Algorithmic classification of the regarded problem within the Smart Grid domain • (Distributed) control algorithm

requirements given by the problem definition. These requirements might also include hard real time restrictions for the runtime behaviour of the algorithm, but in principle, the analysis phase of SGAE tends to be more general than the analysis phase in AE. In our former projects this phase was not carried out in detail, because a simulation-based validation of the algorithms appeared to be adequate for the application. But formal analysis will become considerably more important when we start to develop algorithmic concepts for system critical ancillary services like frequency stabilization in the grid. Adding this phase to Smart Grid Algorithm Engineering fulfills requirement 2 (*reflect dependable systems behaviour*).

6.5.1. Details

See Table 5.

6.5.2. Example

In our GridSurfer example, a possible hard constraint could have been a limitation on the simultaneity factor, i.e. a restriction on the number of EVs that can be charged in parallel due to limitations of the grid, e.g. the thermal load capacity of power lines. It has to be proven formally that the optimisation algorithm respects this limitation in every case. A real-time restriction could have been a fixed maximum specified for the length of the time interval between plugging an EV to the local grid and reception of the calculated charging interval. The length of this time interval restricts the available computation time for the optimisation algorithm and vice versa limits the number of EVs that can be plugged simultaneously to the grid at a local transformer station. But this limit has to be derived formally by a worst-case analysis of the behaviour of the algorithm.

6.5.3. Extensions

For other applications domains such as the automotive domain, complete workflows from the specification of a system to its implementation as a formally proven distributed embedded system have been proposed (Büker et al., 2013). It is a challenging task to transfer such ideas to the smart grid domain, because there are at least three significant differences between the domains:

- The number of controllable devices in a power grid is very large compared to a car or an airplane.
- The number and type of devices to be controlled in a part of the grid, e.g. a medium voltage grid, is changing and the control system has to adapt automatically to these changes.
- The future behaviour of devices such as PV systems can only be forecast but not certainly be determined, so the control system has to adapt to unexpected behaviour of devices as a normal case.

Thus, it has to be carefully analysed which formal techniques can be applied or adapted to the analysis of what type of hard constraints – especially hard real-time constraints – of smart grid control algorithms, and which conditions have to be satisfied by the algorithms for such a formal analysis. It seems to be a promising approach to start with comparatively simple control algorithms like the autonomous contribution of photovoltaic systems to frequency control (Hermanns and Hartmanns, 2013); these algorithms have the potential for unintentional synergetic behaviour.

6.6. Implementation

The objective of this phase is to implement prototypic software needed to evaluate the application-specific performance of algorithms designed for the specified problem within the Smart Grid domain. Therefore, criteria other than those used in industrial software development are needed regarding software quality. When we move from research to practice though, appropriate software development process models should be applied (see extensions, Section 6.6.2).

6.6.1. Details

See Table 6.

6.6.2. Extensions

As soon as experimental evaluation of control concepts shows promising results, application-oriented researchers should work together with industrial partners for further industrial exploitation. In this process, established software development process models should be chosen. In the Smart Grid domain, a first start would be the use case based methodology based on the SGAM as defined by Trefke et al. (2013). Therefore it would be important to extend the implementation phase to define the transition of SGAE to the use case based methodology. By this means the foundation to a seamless transition from application oriented research to application development would be laid.

6.7. Experiment

The experimental phase is – in addition to the design phase – of major importance to our approach. We already discussed several aspects regarding simulation-based experimentation in Section 4.1, especially focussing on the design of statistically sound experiments and suitable Smart Grid scenarios. In the procedure outlined below, we refer to this methodical background. It is also important

Table 5
SGAE phase analysis.

Objective	The objective of this phase is to analyse whether the proposed solution fulfills hard constraints that might be part of the problem definition.
Input	• Problem definition (conceptualisation phase) • Domain PIs (conceptualisation phase) • Hard constraints (conceptualisation phase) • (Distributed) control algorithm (design phase)
Roles	AE
Procedure	A promising approach to formally analyse whether a control algorithm fulfills given hard constraints is based on model checking (Baier et al., 2008). This method requires that the algorithm is defined as an automaton (e.g. a hybrid automaton) and the constraints are given in a specific temporal logic. If the state space of the automaton can be represented finitely, the formula of the specified constraints can be checked relating to the state space. Of course, this method tends to be very complex and time-consuming for complex systems, so there is a lot of research going on to speed up the model checking approach for specific models and types of constrains. Additionally, dependability analysis methods for static systems cannot directly be applied to self-adaptive systems, so more specific methods have to be developed and applied for this case (Güdemann et al., 2006).
Output	Algorithm with formally proven characteristics

Table 6
SGAE phase implementation.

Objective	The objective of the implementation phase is to generate a prototypic implementation of the algorithms designed including code to generate data sets needed for the identified domain performance indicators.
Input	• Designed algorithm (design phase) • Domain performance indicators (conceptualisation phase) • Existing software systems (if applicable)
Roles	AE
Procedure	The procedure within the implementation phase can be very different, depending not only on rules, guidelines and existing software systems, but also on the requirements regarding the transferability of the code into practice. Therefore, we hint to typical aspects relevant for the design of distributed self-organisation algorithms: *Agent (system) architecture:* Depending on the focus of the research question, specific agent architectures might be very useful or even obligatory.[a] *Framework decision:* Within energy management, many researchers use JADE (Telecom Italia S.p.A., 2013) as a framework for agent-based systems (Beer and Tröschel, 2009), (Linnenberg et al., 2011). Researchers might as well implement their own system for distributed systems, e.g. to overcome problems like non-determinism in JADE. If real-time requirements are given, they have to be taken into account with the framework decision as well. *Integration strategy:* If researchers work together on distributed systems, they have to integrate the different software parts; at the same time, they have to ensure the result quality of the single algorithms. Current software development trends give useful hints on how to ensure this (e.g. by setting up an automated test suites early in the implementation phase).
Output	• Prototypic implementation of algorithms • Software code for the creation of data sets needed for the calculation of domain performance indicators

[a] In Wooldridge (2009) an introduction to different agent architecture concepts like e.g. BDI, InteRRaP is given.

to emphasise that the experimental phase of SGAE must not be mixed with the implementation phase: in the debugging processes within implementation, iterative prototyping may be needed depending on the underlying software development process, but the system under test must not be modified within the experimental and evaluation phase. Instead, the insights gained especially in the evaluation phase should be used as an input for a reiteration of the overall process.

6.7.1. Details

See Table 7.

6.7.2. Example

In the research project GridSurfer, we developed a simulation model of electric vehicles and combined it with an already implemented simulation model of photovoltaic units and power flow analyses. Thus, we were able to simulate different scenarios regarding smart charging of electric vehicles in low voltage grids. The simulation results were stored in an HDF5-database. Since they included very detailed information on the internal states of all simulated entities they were consolidated for the analysis of the domain-specific PIs we employed in the evaluation phase (see Section 6.3.2).

6.7.3. Extensions

Depending on how close the development cycle is in regard to a transfer to field deployment, it may be necessary or preferable to include real power systems in a software-in-the-loop approach. In order to allow the simulation-based evaluation of Smart Grid control systems in a realistic setting, we propose and currently work on the integration of standard–compliant interfaces, e.g. based on IEC 61850 or the OPC Unified Architecture, for the supervision and control of both simulated and real system components (Schütte et al., 2013). Thus, the field deployment of evaluated control systems is expected to be fast and inexpensive as reimplementation becomes (partially) unnecessary.

Table 7
SGAE phase experiment.

Objective	The objective of this phase is the generation of data with a maximum of informational content regarding the evaluation of the control system under test using a minimum of (time-consuming) simulation runs.
Input	• Implementation of the control system (implementation phase) • Scenario definitions, domain-specific PIs and corresponding metrics (conceptualisation phase) • Implemented and evaluated simulation models (especially from model library)
Roles	EE, DE
Procedure	*Development and/or adaption of simulation models:* Ideally, all required simulation models for components should already be available in the model library mentioned in Section 4.1. In our experience, however, some adaptations to existing component models or even the development of new models are usually necessary regarding the specific requirements of the evaluation of the control system under test and the domain-specific PIs and their corresponding metrics. This requires the technical expertise of domain experts (DE) to check whether an existing model can be adapted to a new application domain or when a new model validation has to be performed. After the component model has been modifying and validating it has to be added to the model library along with its documentation. This process step reflects requirement 3 (*use validated simulation models*). *Design of scenarios:* Experimental engineers (EE) implement the scenarios designed in the conceptualisation phase for execution in a simulation framework. Using mosaik, we rely on a domain-specific language for an efficient and computer-executable formal specification of scenarios (Schütte and Sonnenschein, 2012). With this convenient approach for scenario interchangeability, requirement 4 is fulfilled (*provide scenario interchangeability*). *Design of experiments:* In view of the objective of this phase, a methodically sound design of simulation experiments is crucial. As discussed in Section 4.3, we strongly suggest that experimental engineers (EE) employ an appropriate methodology for this step. *Model composition:* Taking the designed scenarios and experiments into account, experimental engineers combine the simulation models in order to yield an overall simulation of the physical topology of the system under test. *Pre-processing of simulation results:* As simulation experiments usually generate large amounts of data, experimental engineers and domain experts pre-process the simulation results for simplified use in the evaluation phase. This may include aggregation, consolidation or any other kind of data processing.
Output	Pre-processed data from simulation runs

6.8. Evaluate

In this last phase of Smart Grid Algorithm Engineering, we check the experimental results against our initial hypotheses using the domain performance indicators specified early in the process. The insights obtained from this are manifested in a knowledge base and represent the key result of the whole SGAE process, thus reflecting requirement 5 (*document evaluation results including meanders*). In this knowledge base, typical Smart Grid problems (from the conceptualisation phase), their algorithmic classification (from the design phase), algorithms (from the design and implementation phase), domain performance indicators and hypotheses (from the conceptualisation phase) with indication if they were strengthened or weakened (from the analytical and experimental phase) should be manifested. These results can only be generated by the domain experts, who evaluate the domain-specific performance using the data sets from simulation. In our understanding, the creation and maintenance of such a knowledge base is the key idea of knowledge production in the sense of Gibbons et al. (1994).

6.8.1. Details

See Table 8.

6.8.2. Example

Taking the hypotheses formulated in the example in Section 6.3 into account, we analysed the simulation results from the project GridSurfer with regard to (a) the share of PV feed-in in a given power grid that was consumed by electric vehicles (EV), and (b) the effects on power grid assets, especially on the power transformer in the local substation of the given power grid.

Fig. 5 shows the results regarding the local consumption of electric feed-in from photovoltaic plants. Using the vehicle-to-grid approach, a substantial amount of the PV feed-in was consumed locally, while both uncontrolled and controlled charging resulted in significantly smaller consumption rates. This especially follows from the fact that the EVs' batteries were discharged solely by driving, while the vehicle-to-grid approach additionally used the batteries to provide households with power in times of high demand.

Regarding the effects on the power grid's assets, Fig. 6 shows the different probability density functions of the power transformer's load states over the course of one (simulated) year. It is noteworthy that the vehicle-to-grid approach yields a much more balanced degree of utilisation, thus implicitly raising the potential capacity for further integration of renewable energy in the considered low voltage power grid.

6.8.3. Extensions

Up to now we do not evaluate metrics regarding self-organisation, e.g. global state awareness or autonomy that are in parts only assessable using quantitative measures (Holzer and de Meer, 2009). It would be interesting to include the metrics into the knowledge base to find out in the long term, if there is a correlation between the degree of distribution of the algorithm and the domain-specific performance.

6.9. SGAE Iterations

Once the six phases of SGAE are finished, several reiterations of the cycle may be needed, stepping further from a broad problem description to an algorithm with proven characteristics regarding hard constraints and experimentally shown characteristics regarding domain-specific performance indicators. The more often the cycle is repeated, the closer we come to an application in the field. Especially the non-functional requirements may change in this process because of the feedback from the domain experts. This may lead to different implementations and technological settings of the initial algorithm.

7. Conclusion and discussion

In this paper, we outlined a process model for the engineering of (distributed) Smart Grid algorithms. Using the Smart Grid Architecture Model (SGAM), we derived a set of minimal requirements that a suitable implementation of such a process model has to comply with. On the basis of these requirements, we applied and adapted Algorithm Engineering (Sanders, 2009) for the Smart Grid application domain. The resulting Smart Grid Algorithm

Table 8
SGAE phase evaluate.

Objective	The objective of this phase is to evaluate the designed algorithms against the hypotheses defined early in the process (probably refined within the other steps and/or several cycles of SGAE). The results of this evaluation are manifested in a knowledge base, giving input for new projects and research questions and delivering information for researchers new to the application domain.
Input	• Problem definition (conceptualisation phase) • Algorithmic categorisation (design phase) • Domain PIs (conceptualisation phase) • Hypotheses (conceptualisation phase) • Algorithm design (design phase) • Simulation results (data sets reflecting domain PIs) (experiment phase)
Roles	DE
Procedure	*Evaluation:* In this step the hypotheses from the initial conceptualisation – probably further refined within one or repeated cycles of SGAE – are checked using the simulation results reflecting the domain PIs. Then domain experts either decide that a hypothesis is strengthened or weakened, maybe even falsified. Statistical background is needed for this task to avoid typical errors like accounting coincidence for correlation and so on. *Transfer to knowledge base:* This step can be very easy to realise, if the results of the evaluation have been published with the needed level of detail. It can also be quite time-consuming, if the results were not as expected or published only in an inappropriate manner. Especially for algorithms that performed worse than expected, this step is very useful, though. In our experience, the documentation of bad results in that sense is not done very enthusiastic, not even in project deliverables needed for funding reasons. Therefore we have to include this process step within the internal project plan as a key activity to finish a project. This can be compared to the post-mortems known from software development projects. With this extension to our process, requirement 5 is fulfilled (*document evaluation results including meanders*).
Output	Knowledge base with information regarding the algorithms and their domain-specific performance

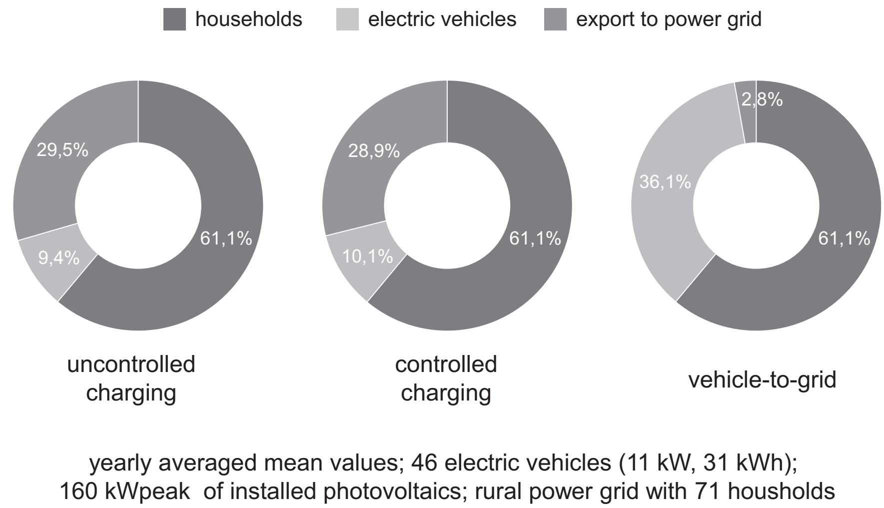


Fig. 5. Sinks of electric feed-in from photovoltaic plants for different charging strategies (Tröschel et al., 2011).

Engineering (SGAE) process model comprises an initial conceptualisation phase and five iteratively repeatable phases:

Initial phase

0. *Conceptualize*: In close discussion with experts from the application domain, the problem to be solved is understood and described, and domain-specific performance indicators and hypotheses regarding the solution's performance are defined.

Iterative phases

1. *Design*: On the basis of a formal definition and characterisation of a given problem, an algorithmic solution is designed and checked for domain compatibility.
2. *Analyse*: The designed algorithm is analysed with regard to real-time capability, performance and dependability.
3. *Implement*: A prototypical software implementation of the designed algorithm is realised, and the metrics for important performance indicators as defined during the conceptualisation phase are implemented.
4. *Experiment*: The prototype is integrated into varying Smart Grid scenarios in order to generate statistically sound experimentation data on the algorithm's run-time behaviour using simulation frameworks.

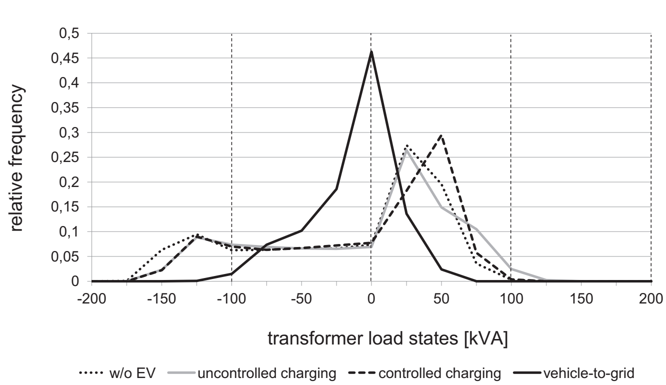


Fig. 6. Probability density functions of transformer load states for different charging strategies (Tröschel et al., 2011).

5. *Evaluate*: Using the implemented metrics and the data gained from experimentation, the domain-specific performance of the algorithmic solution is evaluated, thus strengthening or falsifying a set of initially formulated hypotheses.

In the context of Smart Grid research it is often not feasible to conduct real-world experiments due to financial and safety-related reasons. Therefore, experimentation in varying and large-scale scenarios in a laboratory environment is required to evaluate an algorithmic solution regarding its applicability to different Smart Grid environments. Thus, Smart Grid simulation is a core element for experimental analysis and evaluation purposes in SGAE.

SGAE is, of course, a living artefact. We currently employ the process model in different research projects, especially in the context of the project group Smart Nord. In addition, the requirements we initially derived using the SGAM were motivated by research experience and problems we encountered in our research projects. We therefore do not claim comprehensiveness in regard to possible requirements resulting from different experiences. Thus, there is (and most certainly will continue to be) the need to improve and refine the process model continuously. Resulting from our employment of SGAE so far, there are at least three distinct extensions we will have to address in future work:

- *Dependability analysis*: For some use cases, it will be necessary to prove certain characteristics or behavioural traits of an algorithmic solution formally. Thus, the support of formal dependability analyses should be improved. Basic concepts as described in (Avizienis et al., 2004) should be checked for applicability. This is crucial for research in the field of automation and protection concepts.
- *Model library*: We explained the importance of a model library to reduce the modelling overhead during experimentation setup. For such a model library, a documentation regarding application area of the model, validation status, temporal resolution and composibility with other models is needed. Therefore an appropriate documentation template and accompanying process is needed.
- *Transfer to field application*: In order to reduce the overhead for a transfer of an evaluated algorithmic solution to field application, SGAE should allow for an alignment with the use case based methodology of the SGAM (Trefke et al., 2013). We expect

further developments in that area, with standardisation and certification being a growing topic. We therefore intend to sketch the transition of SGAE's iterative cycles to the use case based methodology in the near future.

Acknowledgements

We thank all scientists that contributed to the development of the work presented here in the projects GridSurfer and Smart Nord. Parts of this work have been funded by the German Federal Ministry of Economics and Technology (BMWi) (grant 01ME09017) and the Lower Saxony Ministry of Science and Culture through the 'Niedersächsisches Vorab' grant programme (grant ZN 2764). Additionally, we thank the anonymous reviewers for their helpful comments that helped to improve this contribution.

References

50Hertz Transmission GmbH, 2011. EEG Anlagenstammdaten. URL. http://www.eeg-kwk.net/de/Anlagenstammdaten.htm (accessed 22.08.13.).
50Hertz Transmission GmbH, 2013. Netzentwickungsplan Strom 2013. URL. http://www.netzentwicklungsplan.de (accessed 22.08.13.).
Alcamo, J., Ribeiro, T., 2001. Scenarios as Tools for International Environmental Assessments.
Appelrath, H.-J., Kagermann, H., Mayer, C. (Eds.), 2012. Future Energy Grid. Acatech STUDIE. Springer, Berlin, Heidelberg.
Avizienis, A., Laprie, J.-C., Randell, B., Landwehr, C., 2004. Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Depend. Sec. Comput. 1, 11–33.
Baier, C., Katoen, J.-K., Larson, K.G., 2008. Principles of Model Checking. MIT Press.
Beer, S., Tröschel, M., 2009. MACE – multiagent control for energy infrastructures. In: Athanasiadis, I.N., Mitkas, P.A., Rizzoli, A.E., Gómez, J.M. (Eds.), Proceedings of the 4th International Symposium on Information Technologies in Environmental Engineering. Springer Verlag.
Benz, J., Hoch, R., Legovic, T., 2001. ECOBAS – modelling and documentation. Ecol. Model. 138, 3–15.
Bindner, H., Gehrke, O., Lundsager, P., Hansen, J.C., Cronin, T., 2004. IPSYS – a tool for performance assessment and supervisory controller development of integrated power systems with distributed renewable energy. In: Solar 2004: Life, the Universe and Renewables. Perth.
BMU, 2010. Leitstudie 2010—Langfristszenarien und Strategien für den Ausbau der erneuerbaren Energien in Deutschland bei Berücksichtigung der Entwicklung in Europa und global. Tech. rep.. BMU.
Büker, M., Damm, W., Ehmen, G., Henkler, S., Janssen, D., Stierand, I., Thaden, E., 2013. From specification models to distributed embedded applications: a holistic user-guided approach. In: Published Online at SAE 2013 World Congress.
Bundesregierung, 2010. Energiekonzept der Bundesregierung. Tech. rep.. Bundesregierung.
Chassin, D.P., Widergren, S.E., 2009. Simulating demand participation in market operations. In: 2009 IEEE Power Energy Society General Meeting. IEEE, pp. 1–5. URL. http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=5275369 (accessed 22.08.13.).
Deutsche Energie-Agentur GmbH, 2012. dena-Verteilnetzstudie. Ausbau- und Innovationsbedarf der Stromverteilnetze in Deutschland bis 2030 (Tech. rep).
Energietechnische Gesellschaft im VDE, 2009. Energiespeicher in Stromversorgungssystemen mit hohem Anteil erneuerbarer Energieträger (Tech. rep).
Energietechnische Gesellschaft im VDE, 2012. VDE-Studie Energiespeicher für die Energiewende (Tech. rep).
ENTSO-E, 2011. The Harmonized Electricity Market Role Model. Tech. rep.. ENTSO-E.
European Energy Research Alliance, 2013. D2.1 Overview of Simulation Tools for Smart Grids. Tech. rep. URL. http://www.eera-set.eu/ (accessed 30.08.13.).
Garey, M.R., Johnson, D.S., 1979. Computers and Intractability: a Guide to the Theory of NP-completeness (Series of Books in the Mathematical Sciences). W. H. Freeman.
Gausemeier, J., Wenzelmann, C., Plass, C., 2009. Zukunftsorientierte Unternehmensgestaltung: Strategien, Geschäftsprozesse und IT-Systeme für die Produktion von morgen. Hanser Verlag.
Gendreau, M., Potvin, J.-Y., 2010. Handbook of Metaheuristics. Springer-Verlag. Springer.
Gibbons, M., Limoges, C., Nowotny, H., Schwartzman, S., Scott, P., Trow, M., 1994. The New Production of Knowledge. Sage Publications.
Güdemann, M., Ortmeier, F., Reif, W., 2006. Safety and dependability analysis of self-adaptive systems. In: Second International Symposium on Leveraging Applications of Formal Methods Verification and Validation Isola 2006. 2006 2nd International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2006). IEEE, pp. 177–184.
Harris, G., 2002. Integrated assessment and modelling: an essential way of doing science 1. Environ. Model. Softw. 17, 201–207.
Hermanns, H., Hartmanns, A., 2013. An Internet inspired approach to power grid stability. IT – Inf. Technol. 55 (2), 45–51.
Hevner, A., March, S., Park, J., Ram, S., 2004. Design science in information system research. MIS Quart. 28 (1), 75–105.
Hilty, L.M., Arnfalk, P., Erdmann, L., Goodman, J., Lehmann, M., Wäger, P. a, 2006. The relevance of information and communication technologies for environmental sustainability – a prospective simulation study. Environ. Model. Softw. 21 (11), 1618–1629.
Hollinger, R., Wille-Haussmann, B., Erge, T., Sönnichsen, J., Stillahn, T., Kreifels, N., 2013. Speicherstudie 2013. Tech. rep.. Fraunhofer ISE.
Holzer, R., de Meer, H.D., 2009. Quantitative modeling of self-organizing properties. In: Proc. of the 4th Int'l Workshop on Self-organizing Systems (IWSOS 2009).
Holzer, R., de Meer, H.D., Bettstetter, C., 2008. On autonomy and emergence in self-organizing systems. Selforganizing Systems 51 (1), 157–169.
HOMER Energy LLC, 2013. HOMER Energy. URL. http://homerenergy.com/ (accessed 22.08.13.).
International Electrotechnical Commission, 2008. Publicly Available Specification (PAS) 62559 IntelliGrid Methodology for Developing Requirements for Energy Systems (Tech. rep).
Janssen, S., Ewert, F., Li, H., Athanasiadis, I., Wien, J., Thérond, O., Knapen, M., Bezlepkina, I., Alkan-Olsson, J., Rizzoli, a.E., Belhouchette, H., Svensson, M., van Ittersum, M., Dec. 2009. Defining assessment projects and scenarios for policy support: use of ontology in integrated assessment and Modelling. Environ. Model. Softw. 24 (12), 1491–1500.
Kelton, W.D., 2000. In: Joines, J.A., Barton, R.R., Kang, K., Fishwick, P.A. (Eds.), Proceedings of the 2000 Winter Simulation Conference, Experimental Design For Simulation, pp. 32–38.
Kleijnen, J., 2001. Experimental Designs for Sensitivity Analysis of Simulation Models (Tech. rep).
Kok, K., Warmer, C., Kamphuis, R., Mellstrand, P., Gustavsson, R., 2005. Distributed control in the electricity infrastructure. In: Proceedings of the International Conference on Future Power Systems, 2005.
Lehnhoff, S., 2010. Dezentrales vernetztes Energiemanagement – Ein Ansatz auf Basis eines verteilten Realzeit-Multiagentensystems. Vieweg + Teubner.
Lehnhoff, S., Rohjans, S., Holzer, R., de Meer, H.D., 2013. Towards a mapping of self-organization properties and non-functional requirements in the operation of future smart grids. In: 7th International Workshop on Self-organizing Systems (IWSOS). Palma.
Linnenberg, T., Wior, I., Schreiber, S., Fay, A., 2011. A market-based multi-agent-system for decentralized power and grid control. In: 16th IEEE International Conference on Emerging Technologies and Factory Automation (IFTA2011). Toulouse, France.
Lynch, N.A., 1996. Distributed Algorithms. In: Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann.
Nieβe, A., Lehnhoff, S., Tröschel, M., Uslar, M., Wissing, C., Appelrath, H.-J., Sonnenschein, M., 2012. Market-based self-organized provision of active power and ancillary services: an agent-based approach for smart distribution Grids. In: Complexity in Engineering (COMPENG), 2012. Aachen.
Oh, R.P.T., Sanchez, S.M., Lucas, T.W., Wan, H., Nissen, M.E., Nov. 2009. Efficient experimental design tools for exploring large simulation models. Comput. Math. Organ. Theory 15 (3), 237–257. URL. http://www.springerlink.com/index/10.1007/s10588-009-9059-1 (accessed 22.08.13.).
Penya, Y., Jennings, N.R., 2008. Optimal combinatorial electricity markets. Int. J. of Web Intel. Agent Syst. 6 (2), 1–13.
Pöchacker, M., Sobe, A., Elmenreich, W., 2013. Simulating the Smart Grid.
Popper, K.R., 1959. The Logic of Scientific Discovery. Hutchinson. Translation of *Logik der Forschung*. published in Vienna, Austria by Verlag Julius Springer, 1934.
Ramchurn, S.D., Vytelingum, P., Rogers, A., Jennings, N., 2011. Agent-based control for decentralised demand side management in the smart grid. In: Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems – Innovative Applications Track (AAMAS 2011).
Sanders, P., 2009. Algorithm engineering – an attempt at a definition. In: Efficient Algorithms. Springer, pp. 321–340.
Scherfke, S., Schütte, S., 2013. Mosaik—Architecture Whitepaper. URL. http://mosaik.offis.de/downloads/mosaik_architecture_2012.pdf (accessed 22.08.13.).
Schütte, S., Sonnenschein, M., 2012. Mosaik – scalable smart grid scenario specification. In: Proceedings of the 2012 Winter Simulation Conference. Berlin, Germany.
Schütte, S., Scherfke, S., Tröschel, M., 2011. Mosaik: a framework for modular simulation of active components in smart grids. In: 1st International Workshop on Smart Grid Modeling and Simulation (SGMS). Brussels.
Schütte, S., Rohlfs, H., Nieße, A., Rohjans, S., 2013. OPC UA compliant coupling of multi-agent systems and smart grid simulations. In: 9th Annual Conference of the IEEE Industrial Electronics Society (IECON 2013). Vienna.
Siebertz, K., Van Bebber, D., Hochkirchen, T., 2010. Statistische Versuchsplanung: Design of Experiments (DoE). Springer. URL. http://www.springerlink.com/content/978-3-642-05492-1 (accessed 22.08.13.).
SmartGrids European Technology Platform, 2010. SmartGrids: Strategic Deployment Document for Europe's Electricity Networks of the Future. URL. http://www.SmartGrids.eu/documents/SmartGrids/_SDD/_FINAL/_APRIL2010.pdf (accessed 22.08.13.).
Sonnenschein, M., Appelrath, H.-J., Lehnhoff, S., Mayer, C., Uslar, M., Nieße, A., Tröschel, M., 2012. Distributed and self-organized coordination in smart grids. In: VDE-Kongress 2012—Intelligente Energieversorgung der Zukunft. Stuttgart.

Spath, D., Linder, C., Seidenstricker, S., 2011. Technologiemanagement: Grundlage, Konzepte, Methoden. Fraunhofer Verlag.
Stamminger, R., Broil, G., Pakula, C., Jungbecker, C., Braun, C., Rüdenauer, I., Wendker, C., 2008. Synergy Potential of Smart Appliances. Tech. rep.. Wilhelms-Universität Bonn.
Telecom Italia S.p.A, 2013. JADE (Java Agent Development Framework). URL. http://jade.tilab.com/ (accessed 22.08.13.).
Trefke, J., González, J., Dänekas, C., 2013. IEC/PAS 62559-based use case management for smart grids. In: Standardization in Smart Grids. Springer.
Tröschel, M., Scherfke, S., Schütte, S., Nieße, A., Sonnenschein, M., 2011. Using electric vehicle charging strategies to maximize PV-integration in the low voltage grid. In: International Renewable Energy Storage Conference (IRES 2011). Berlin.
Vornberger, J., Tröschel, M., Nieße, A., Appelrath, H.-J., 2011. Optimized charging management for electric vehicles. In: 1st International Renewable Energy Conference and Exhibition IRENEC 2011. Eurosolar.
Wooldridge, M., 2009. An Introduction to MultiAgent Systems. John Wiley & Sons.