Simulation Modelling Practice and Theory 103 (2020) 102095

Contents lists available at ScienceDirect

Simulation Modelling Practice and Theory

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

Software requirements for interoperable and standard-based power system modeling tools

Francisco J. Gómez[a,†], Miguel A. Aguilera[b], Svein H. Olsen[c], Luigi Vanfretti[d,*]

[a] Semantum Oy, Helsinki, Finland
[b] Instituto Costarricense de Electricidad (ICE), San José, Costa Rica
[c] Statnett SF, Oslo, Norway
[d] Rensselaer Polytechnic Institute (RPI), Troy, NY, USA

ARTICLE INFO

Keywords:
Interoperability
CIM
ISO
UML
SysML
Model exchange
Information modeling
Power systems modeling
Power systems dynamics
Modelica
OpenIPSL
Power system simulation
Power system analysis

ABSTRACT

Standardization processes and collaborative tools are becoming essential for interoperable modeling and simulation of power systems. The implementation of new electrical network codes at European level requires the development of a common grid modeling exchange standard. Consequently, this requires the development of dynamic models which expose their individual and integrated physical behavior in simulations used in trade-off analyses. These analyses are of particularly importance due to the increasing number of renewable variable energy sources connected to the grid. The development of such models leads to the development of new requirements for power system studies, while ensuring interoperability, security, and privacy of the models. The use of open standards for modeling and simulation could allow the export of all the necessary model information and physical behavior to guarantee consistency between software vendors and stakeholders. This work presents a formalization of functional and non-functional requirements for new standard-compliant software tools using the System Modeling Language (SysML). This work also presents the implementation of these requirements using available information modeling and equation-based modeling standards, illustrating that standards can be combined for model exchange and co-simulation by the application of the FMI standard.

1. Introduction

1.1. Motivation

Simulation models representing the dynamic behavior of a power system are gaining higher interest at the European level. The European Network of Transmission System Operators for Electricity (ENTSO-E) [1] has released a series of requirements for the implementation of network codes, which require dynamic models for simulations in trade-off analysis for the connection of different energy sources to the grid [2]. Therefore, standardization processes and collaborative tools to enable stakeholder interactions are required to improve the two above (models and regulations, i.e. network codes) and to adapt to new requirements while ensuring interoperability, security, and privacy [3,4].

In a specific example, a technical regulation for grid connections stipulates that "... the transmission system operator must

---

[*] Corresponding author.
E-mail address: vanfrl@rpi.edu (L. Vanfretti).
[†] Deceased September 21st, 2019

https://doi.org/10.1016/j.simpat.2020.102095
Received 5 January 2020; Received in revised form 19 March 2020; Accepted 20 March 2020

maintain and expand the simulation models continuously …" [5]. This technical regulation also states that "The simulation model must be supplied in the form of a block diagram which mainly by means of logical and mathematical functions – primarily transfer functions in the Laplace / z-domain – describes the properties of the PV power plant" [5].

These requirements indirectly state that software vendors should embrace the utilization of an approach that offers better reusability of code and allows for information exchange of steady-state and dynamic behavior. However, traditional power system simulation tools only partly comply with the requirements because they do not allow the export of all the necessary model information to guarantee consistency (i.e. explicit model realizations using a strict mathematical representation). Furthermore, power system models based on transfer functions could be replaced by other domain's mathematical representation. Thus, power system multi-domain models could provide better representation of real systems behavior.

Most of the requirements in relation with power systems applications and analysis methodologies are developed with natural language. The ambiguity of natural language makes it difficult to extract precisely specific information requirements and is up to the engineer to interpret those requirements and their relationships, leaving an open door to ambiguity [6]. Industry practices are more concerned with the direct implementation of technical requirements, mainly because of issues regarding business objectives and productivity incentives [7]. The development of legacy power systems simulation tools is mostly guided by direct prerequisites from tool's users, e.g. Transmission System Operators (TSOs), and by proprietary-tool vendors [8] using a traditional cathedral-like method [9].

Model and tool development is conceived to solve short-term or specific technical problems. Models only allow to represent specific time-scales such as "static" (i.e. power flow), "dynamic" electro-mechanical, and for electro-magnetic dynamics (i.e. fast-transient system responses). In this approach, models are described within the structure of a specific simulation solver (e.g. discretized models using the trapezoidal integration rule), and thus, **unambiguous modeling is left for later stages of the tool development process if at all considered, resulting in a deficit of general requirements catering to the use of modeling and simulation standards**.

1.2. Related work

Current trends in power system tool development is to include both model exchange and analysis tools within a common framework [10]. For example, in the power system domain, the design and specifications implementation is focused in a goal or problem to be solved using pre-defined algorithms implemented in domain-specific tools (e.g. PSS/E or DigSilent). In other words, specification and requirement analysis are entirely coupled with the implementation of power system M&S tools.

In other engineering domains, the focus on Requirement Engineering is gaining more interest. The work in [11] presents a study on how safety critical industries address the development of requirements. This study shows that the "**challenge is to produce more useful requirements documents".** The MODRIO project [12] proposes the design of meta-models and requirements for multi-domain systems, involving energy and transportation domain models. Results from this project have been shown in [13] and [14], where equation-based modeling languages can be used for the design and implementation of requirements. Those works provide examples on how to transform the design specifications within the model's equations of shared cyber-physical models for the mechanical and fluid engineering domains.

The OpenCPS Project [15] focuses on interoperability between cyber-physical system models from multiple domains. One of the goals of the project is the definition of requirements on the reliability, usability and flexibility of system models, for the development of a Master Simulation Tool for (co-) simulation, capable of combining information models with mathematical models, with Modelica and the Functional Mock-Up Interface (FMI) standard [16].

The line research of requirements engineering for multi-domain system models, works such as [17,18], provide detailed explanation and specifications for the modeling of new cyber-physical systems using the SysML standard [19]. Thus, the use of SysML-based tools, such as Papyrus [20], and Modelica-based tools, such as OpenModelica [21], in combination of the FMI standard, make it possible the development of tools for system modeling separately from systems simulation and analysis tools.

As part of the work carried out in the OpenCPS project, this paper proposes a formalization of requirements that can be applied for the development of new power system modeling and simulation tools, compliant with both power system information exchange (e.g. CIM) and Modeling and Simulation (M&S) standards (e.g. FMI and Modelica).

1.3. Contributions

The authors propose the formalization of requirements for the design of new power system modeling and simulation tools. The objective is the application of System and Software Engineering theories and practices, which can be implemented to build new standard-compliant software tools.

For this purpose, the Use Case Methodology based from the IEC 62,559–2 [22], is followed in the sequel. The use of different standards is considered for the **development of tools for modeling, and separately, tools for simulation**. The application of these standards can provide new functionalities to the well-known power system analysis and simulation methods. Moreover, the use of different standards can help to fulfill TSOs specifications concerning the exploitation and exchange of simulation models for dynamic studies.

A general use case for modeling tools and another for simulation tools is discussed. Each of them is described using parts of the use case template suggested by the IEC 62,559. The description of the use cases is completed with SysML Use Cases and SysML Requirements diagrams.

Industry requirements, defined in natural language, can be derived into formal technical requirement using SysML. Following the classification proposed by [17], these requirements are divided into functional (i.e. scenarios) and non-functional (i.e. technical specifications).

The definition of functional requirements and non-functional requirements will use available standard modeling semantics, such as the Common Information Model (CIM) for power grid information modeling, and Modelica for equation-based modeling.

In addition, strategies for multi-domain simulation with the adoption of the FMI standard for Model Exchange and Co-Simulation, are also considered. These strategies, in addition to the works presented in [21] and [22] with research projects such as the OpenCPS make the research on modeling and simulation tools an essential part for the development of new software tools.

1.4. Paper organization

The remainder of this paper is organized as follows. In Section II the background considered for the modeling of requirements is explained. Section III describes the use case and requirements for modeling tools. Section IV describes the use case and requirements for simulation tools. Section V illustrates the feasibility of using the different standards proposed to meet requirements. The paper finalizes with conclusion in Section VI.

2. Background

2.1. What do the power system analysis use domain-specific M&S tools for?

A system model is a representation of a physical system that can be expressed in terms of mathematical equations and then translated into a computer program. In the case of power system models, they consist of a set of devices that are typically described by continuous dynamics and discrete events, and become suitable for steady-state and dynamic simulation studies. The simplifications and hypothesis made during the design of power system models are often driven by the numerical solution methods that are available in a specific tool to solve the given system equations, which in turn prescribe modeling specifications.

The implementation of those model components is in most cases ambiguous, and often tightly coupled to both the software tool and the numerical methods used to obtain the steady-state or dynamic response of the system. Thus, the modeling of any power system is defined by the constraints imposed by the model components, prescribed model information that is available in the simulation tool, and by the vendor/developer [25].

Moreover, it becomes difficult to reuse and maintain power network models, because they are confined to a specific computer program. Furthermore, de facto simplifications and hypothesis do not consider the uncertainties that might affect the behavior of the model. These uncertainties are often considered insignificant and simulation results are expected to be conservative [26].

Different strategies are employed to model a limited type of uncertainties of the component's dynamic behavior in the power system model. The typical system approach consists in applying random variations to parameters [26] or by defining more complex power system models with the use of “user-defined models” in a tool's own Domain Specific Language (DSL). However, these strategies are tool specific, leading to lack of portability and imposing a need to reproduce results that becomes cumbersome [8].

2.2. Information modeling for power systems

The Smart Grid Architecture Model (SGAM) developed by the Grid Coordination Group/Reference Architecture Working Group (SG-CG/RA) in the context of the European Commission's Standardization Mandate M/490 [4] proposes different layers that cover the organizational, informational and technical aspects of the use and development of cyber-physical power systems.

The adoption of the SGAM by TSOs, Distribution System Operators (DSOs) and other players within the electrical grid, although limited to relatively few tools [8], has elicited new use cases and requirements to comply with interoperability requirements. Moreover, it led to continuous development and maintenance of new semantic information, such as the CIM and associated software tools [27].

Borrowing from these advances in the power system domain, the CIM [28] and the Common Grid Model Exchange Standard (CGMES), are proposed as standard interfaces specification and semantics for consistent information exchange between TSOs [29]. These standards base their implementation on the IEC 61,970–301 Information Model and the IEC 61,970-(452,456) Profile definitions. The exchange of dynamic models is based on IEC 61,970–302 Information Model for Dynamics and the IEC 61,970–457 Dynamics Profile Specification [30].

The CIM/CGMES standards provide information models of plant's components from the power system perspective, which might result in a limited definition of physical components’ dynamics. Thus, new requirements for modeling tools could include the adoption of available standards for projects related to power plants, such as the ISO 15,925 [31]. The application of such standards could be beneficial for the exchange and reuse of complex plant models’ information, mitigating the high costs of reformatting model information.

The ISO 15,925 standard provides data models, reference data and templates, with Web Ontology Language (OWL) [32] and Resource Description Framework (RDF) [33] representation as a set of rules for information exchange. Thus, its implementation can be easily integrated within the layers of the SGAM.

2.3. Leveraging software and system domain standards for power system modeling and software systems

2.3.1. General purposes modeling languages for modeling requirements

The adoption of other information standards within the power system domain can be used to extend the CIM/CGMES information models and profiles. Equation-based modeling languages for dynamic modeling and simulation can complement those information standards. Thus, it is necessary to use General Purpose Modeling Languages (GPMLs), such as the Systems Modeling Language (SysML) [19] to represent the requirements and enhance the interoperability of cyber physical power systems.

The SysML provides means to develop complete documentation for **system engineering applications**. It is used to define the system components (e.g. software modules or hardware units) and the relationships between those components. These relationships can be designed to express data communication between modules or to represent the physical connection between units and which data they provide.

SysML represents a general abstraction of any system and it does not provide any specific detail on how the implementation of the system should be performed. It is suitable for the adoption of system engineering principles to understand a software tool as modular system, in which each module could be implemented with different M&S languages. Thus, the SysML can be used as a common language to represent the composition and interaction of software modules. The Appendix provides information that supports the choice of SysML for the scope of this work.

With the adoption of available software technologies for M&S, Requirement Engineering (RE) for new software tools is considered [18]. In this work, the RE gathers available standards and methods to support the development of **horizontal modeling and dynamic simulation** workflows, applicable to power system analysis. The RE will be developed following the IEC 62,559–2 template [22], and SysML diagrams will be used to complement the requirement description within the IEC 62,559 methodology.

2.3.2. Modelica language

Equation-based modeling languages such as the Modelica language [34], have proven to be a suitable complement to the CIM/CGMES (which suffers from lack of mathematical description of the system dynamics [24]). Modelica enables the modeler to describe a system by a set of explicit ODEs and DAEs. It allows non-causality in the definition of equations and incorporates object-oriented means to divide a system into different subsystems.

The use of the Modelica languages gives another perspective to develop power systems models, enabling unambiguous modeling and simulation. Thus, a Modelica model is de-coupled from the mathematical solver of the analysis tool. Consequently, the Modelica language provides means to represent the dynamic behavior using an explicit and rigorous mathematical description, and to perform consistent simulations in different tools.

The authors foresee that the development of new software tools that adopt the Modelica language and the continuous enhancements of Modelica compiler capabilities, will enable a transparent basis for the coordination of “data exchange and settlement rules, network security and reliability rules, interoperability rules, and transparency rules” [3].

Current analysis tools use their own implementation of models and solvers. Thus, new standards for model exchange and co-simulation need to be adopted. Modelica and the FMI (below) are proposed herein as a practical solution. This choice was made because of the fact that Modelica is currently the only existing standard that is supported by multiple tools and the only that offers an “open access” language specification for its support by additional potential vendors.

2.3.3. Functional mock-up interface standard

New software technologies enhance simulation capabilities, including the ability to provide more detailed models. Consequently, a power system model could be developed by integrating equations from other domains in such a way that conventional over-simplifications are reduced (one example [34] is the modeling and simulation of industrial control systems in power plants with a richer description of analog, digital and computer systems components [35]).

The FMI standard [16] has a great potential to facilitate the data and functional interoperability of multi-domain components in a system model. FMI is particularly suitable for Cyber-Physical Systems (CPSs), where model components may represent distinct subsystems from different domains. Using the FMI standard, the models from each domain can be exchanged using Functional Mock-Up Units (FMUs). The feasibility of exchanging unambiguous detailed multi-domain models, e.g. combination of gas turbine model with a power system grid model, is demonstrated using a simulation tool that supports the FMI standard.

The adoption of the FMI standard has great potential for multi-domain model exchange that would allow manufacturers to exchange equation-based models, while at the same time protecting their intellectual property. Concurrently, it provides new requirements to be considered in the development of new power systems software tools, e.g. simulation tools that are capable to execute a model that has been implemented by a different modeling tool.

3. Formalization of requirements for modeling tools

This section describes the use case for modeling tool requirements to support standard modeling languages and allow multi-domain modeling. The application of different modeling technologies, shown in section II.B and II.C, makes it possible to create this use case, which gathers modeling capabilities for development of network models. For the sake of simplicity, little part of the template provided by the IEC 62,559 is considered. The description of different scenarios or functional requirements within this use case will be shown. To complete and give further details of this use case, the SysML Requirement diagrams will be used to gather and describe the non-functional requirements and how they are related with the functional requirements.

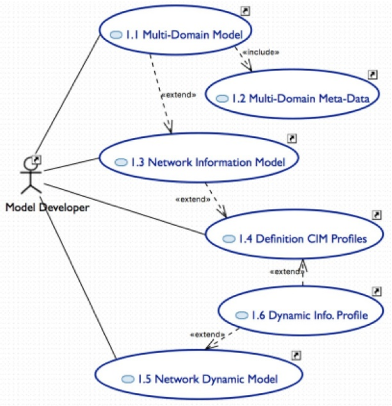


Fig. 1. Use case diagram with the main functional requirements related to the model developer.

3.1. Description and objectives

New modeling tools could be developed as stand-alone applications or plug-ins for general purpose simulation tools. Their functionality could be developed as modules able to interact with each other. To develop more accurate system models, the adoption of multi-domain modeling is attractive and feasible by using the well-known CIM standard in combination with semantics from the ISO 15,926. A module for automatic generation of models could help the model developer to generate different model configurations. Such a module combines the information model with Modelica libraries, e.g. OpenIPSL [36], to describe the dynamic behavior of typical component models and user-defined models. In the case of user-defined models, the modeling tool should be able to generate FMUs, which can be exported. A model checking functionality would be desirable to assess the model's implementation.

3.2. Actors and scenarios for functional requirements

For the development of models, two actors are identified: A Model Developer responsible to define the parameters, mathematical equations and topology of the power systems model; and the Modeling Tool, with capabilities to help the developer to define a complete model.

Figs. 1 and 2 show the main functional requirements to be required by new modeling tools regarding to the related actors. A brief description of each functional requirement is given in Table 1. These requirements come from the use of available standards from information and mathematical modeling that can be combined to create richer model representations including dynamic behavior. The description of Table 1. suggests the use of those standards.

3.3. Non-Functional requirements for the model developer scenarios

Fig. 3. shows the main non-functional requirements related with the Model Developer actor. The REQ_1.2.1, which gathers the semantic information from the ISO 15,926 Data Model specification, satisfies the Multi-Domain Meta-Data (REQ_1.2). The former complies with the requirement to implement the CIM/RDF semantics. Thus, the Model Developer should be able to provide

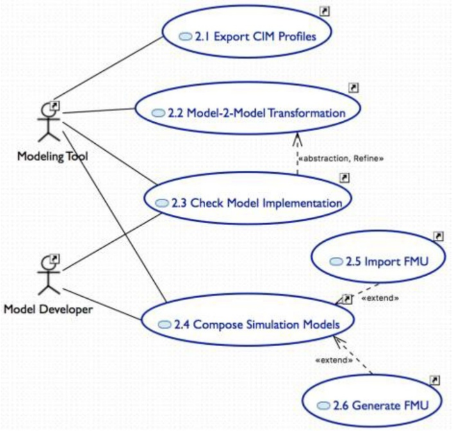


Fig. 2. Use case diagram with the main functionalities related to the modeling tool.

Table 1
Description of power systems specific functional requirements for modeling tools.

Id	Implementation description	Req. relation
1.1	Use of Information Model Standards from other (non-power system) domains.	Includes REQ 1.2 and Extends REQ 1.4
1.2	Creation of a data structure and relationships with the ISO 15,926 Information Model.	
1.3	Information Model expressing power systems semantics from IEC 61,970–301 CIM packages.	
1.4	Classification of power system information from IEC 61,970-(452,453,456,457) CIM Packages, and multi-domain information	Included in REQ_2.1
1.5	Mathematical models for power system components with, e.g., OpenIPSL Modelica library [36].	
1.6	Information Model expressing power systems semantics from IEC 61,970–302 CIM packages	Extends REQ_1.4 and REQ_1.5
2.1	Creation of a CIM/XML file for each CIM Profile.	Includes REQ_1.2 and REQ_1.4
2.2	Automatic generation of models comprised by information model values and mathematical equations.	Includes REQ_1.4 and REQ_1.5
2.3	Each parameter has a numerical value and each component is connected.	
2.4	Simulation model assembly. Connect a FMU block as a part of the current model.	Extended by REQ_2.5 and REQ_2.6
2.5	Translation in whole or in part of the model into an FMU	
2.6	Translation of the model into a description (e.g. parameters) and executable code. The model should not give any compilation errors.	

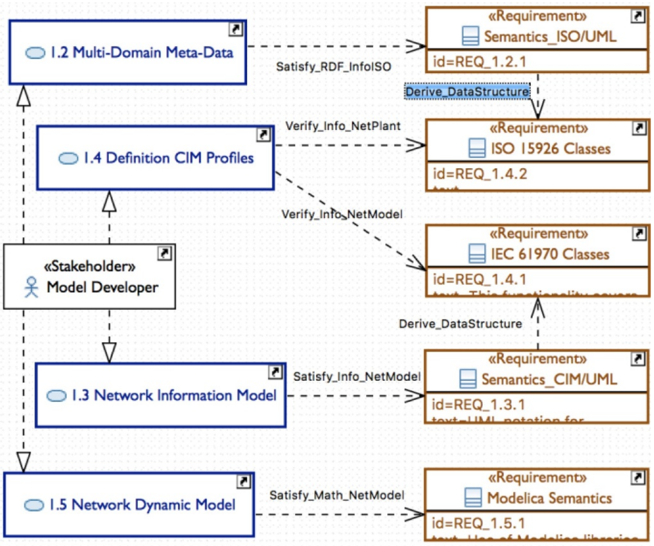


Fig. 3. Requirement specification of the functionalities related to the model developer.

information in RDF from other information standard formats, to comply with the rules provided by the CIM. Moreover, this information must comply with the semantics provided by the CIM in its UML definition (REQ_1.3.1) and used within the REQ_1.3. To ensure detailed profile information of the network models, with multi-domain information, the REQ_1.4 is verified by IEC 61,970 and ISO 15,926 Semantics and Data Structures (REQ_1.4.1 and REQ_1.4.2). Table 2. shows brief description of these requirements.

3.4. Non-Functional requirements for the modeling tool scenarios

The application of semantic information is fulfilled with the implementation of the proper language syntaxes. Fig. 4. shows the requirements related to the Modeling Tool.

These requirements gather the use of the language syntaxes relate to the semantics information depicted in Fig. 4. They are used

Table 2
Non-functional requirements for the model developer actor.

ID	Implementation description	Req. relation
1.2.1	Creation of a data structure and relations from the ISO 15,926 standard	Satisfies REQ_1.2
1.3.1	UML language for modeling power systems components, following the IEC 61,970 standard definitions	Satisfies REQ_1.3
1.4.1	Selection of classes and attributes from the CIM canonical model	REQ_1.4, derived from REQ_1.3.1
1.4.2	Selection of classes and attributes from the ISO-related data structure	REQ_1.4, derived from REQ_1.2.1
1.5.1	Modelica language and Modelica libraries for power systems models, e.g. OpenIPSL	Satisfies REQ_1.5

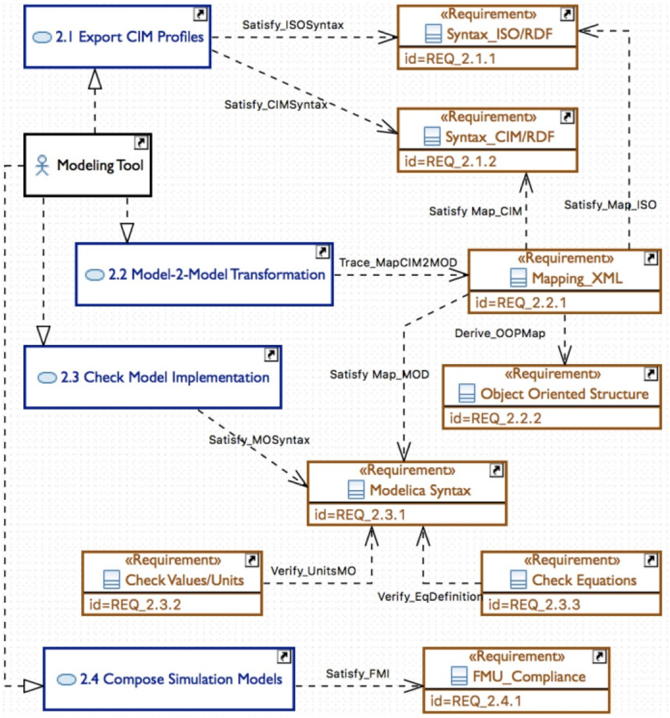


Fig. 4. Requirement specification of the main functionalities related to the modeling tool.

to create the network models for simulation. The functional REQ_2.2 is related with the REQ_2.2.1 that indicates that the model transformation process is implemented through a mapping between information models (REQ_2.1.1 and REQ_2.1.2) and equation-based models implemented in Modelica (REQ_2.3.1).

The bottom of Fig. 4 shows the technical specification related to the application of the FMI standard. The Modeling Tool should be able to provide mechanisms to import FMUs or to generate FMUs from the final model. With the former requirement, a network model can be composed by different Modelica blocks and FMU blocks.

With the later, a Modelica model can be exported into an FMU so to be used as part of other simulation models, which comply with the FMI specifications. Table 3. summarizes these requirements.

4. Formalization of requirements for simulation tools

This section describes the scenario for simulation tools requirements to support standard modeling languages and to allow multi-domain simulation. The application of different simulation technologies, shown in section II.C, makes it possible to create this use case, which gathers simulation capabilities for execution of network models. The description of different functional requirements within this use case will be shown. To complete and give further details of this use case, the SysML Requirement diagrams will be used to gather and describe the non-functional requirements and show how they are related with the functional requirements.

Table 3
Non-functional requirements for the modeling tool.

2.1.1	RDF language and RDF schema for the implementation of the UML definitions from the ISO data structure	Satisfies REQ_2.1
2.1.2	RDF language and RDF schema, following the implementations rules defined by the IEC 61,970–552	Satisfies REQ_2.1
2.2.1	Mapping rules for CIM and ISO naming and OpenIPSL models, implemented with XML language	Traces REQ 2.2, Satisfies REQ_2.1.1 and REQ_2.1.2 and REQ_2.3.1
2.2.2	The transformation tool should be implemented with an object-oriented language, so to support the XML mapping.	Derived from REQ_2.2.1
2.3.1	Use of Modelica language notation, keywords and model structure	Satisfies REQ_2.3
2.3.2	Values must be consistent with power flow information and dynamic specifications; Units defined by power system standards	Verifies REQ_2.3.1
2.3.3	Check component connections, #model variables equal to #model eq.	Verifies REQ_2.3.1
2.4.1	Provide API able to generate and import FMUs to compose the current model.	Satisfies REQ_2.4

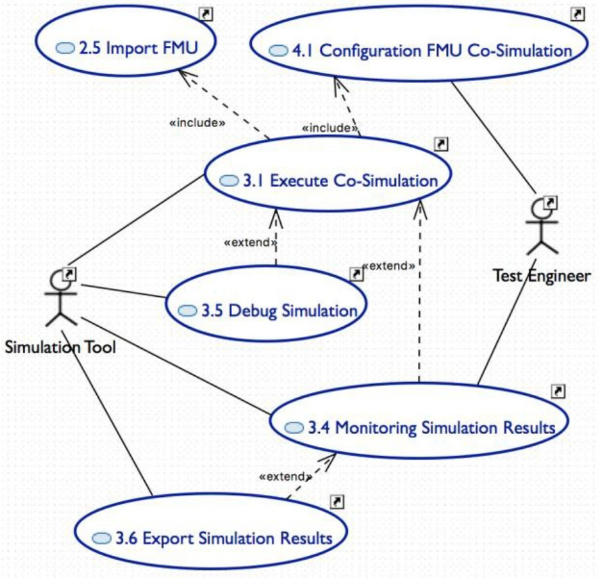


Fig. 5. Use case diagram with the main functional requirements for simulation tools supporting FMUs.

4.1. Description and objectives

This use case applies the same concept of modularity described in Section III.A. Simulation tools should first be conceived as different modules that can interact with each other. And therefore, the development of such tools should consider the adoption of the FMI standard. In this way, the tool mathematical solvers could be used to simulate models that are built from FMUs for Model Exchange (preferred), or the tool could also use the mathematical solver from the simulation model if FMUs are exchanged for Co-Simulation. The simulation tool should also include a mechanism to connect FMUs [37] and define methods for steady-state and time-domain analysis. The results from these simulations could be exported to CIM/RDF formats or to binary data formats, such as the HDF5 that can facilitate the exchange for results. This is described by the diagrams in Figs. 5 and 6 and described in Table 4.

These new tools should allow the monitoring and debugging during simulations. The debugging of equations can give details of status of equations at a specific point in time during simulation [38]. An additional functionality to consider is the development of

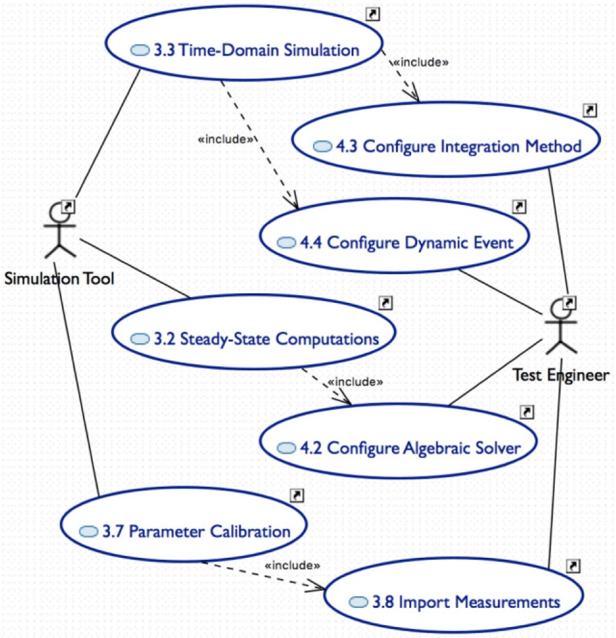


Fig. 6. Use case diagram with the main functional requirements for simulation tools supporting power system models.

Table 4
Description of power systems specific functional requirements for simulation tools.

Id	Implementation description	Req. relation
3.1	Execution of Modelica and FMUs blocks that compose the simulation model.	Includes REQ_3.2 and REQ_3.3
3.2	Execution of algebraic equations of the simulation model.	Includes REQ_4.2
3.3	Execution of the simulation model dynamic-algebraic equations w.r.t. time.	Includes REQ_4.3 and REQ_4.4
3.4	Online traceability and graphical view of model state variables and behavior of the model during the simulation.	Extends REQ_3.1
3.5	Status display of: equation performance at any step of the simulation; time simulation performance; solutions at initialization process & equation solving	Extends REQ_3.1
3.6	Information in the Profiles complying with the IEC 61,970-(301,442,446) CIM packages: State-Variable, Topology and Measurement.	Extends REQ_3.4
3.7	Model dynamic parameters updated with respective values identified from measurements. Updates the dynamic Profile complying with the IEC 61,970-(302,457) CIM packages for dynamic information.	Extends REQ_3.1 and Includes 3.8
3.8	Connection of signal measurements from measurement instruments connected to the real power system.	
4.1	Supply simulation values to the internal solver of the FMU.	
4.2	Supply simulation values to the available non-linear algebraic equations.	
4.3	Supply simulation values to the available mathematical integrator solvers.	
4.4	The model is modified with the implementation of additional fault model or changes in dynamic components' parameters.	

calibration methods to match the model parameters with available measurement data [39].

4.2. Actors and scenarios for functional requirements

For the development of models, two actors are identified: A Model Developer responsible to define the parameters, mathematical equations and topology of the power systems model; and the Modeling Tool, with capabilities to the help the developer to define a complete model (see Fig. 2).

4.3. Non-Functional requirements for simulation tools with FMI support

When defining functionalities, RE allows the use of equation-based modeling languages such as Modelica and the FMI standard for simulation.

To support multi-domain simulation, it is necessary to provide support for FMI so that a model can be composed by either power system elements and components from other domains. The technical specifications related to FMI support for Co-Simulation are synthesized in Fig. 7. A simulation tool should provide analysis features for power system studies, either steady-state simulations (REQ_3.2) or time-domain simulations (REQ_3.3), both related to the Co-Simulation Execution (REQ_3.1). Thus, new simulation tools with FMI support have to be able to execute the basic power system simulations and comply with their requirements as shown in Fig. 8.

As stated in section II.B-3, the FMI standard enables the translation of equation-based models into blocks or FMUs containing the

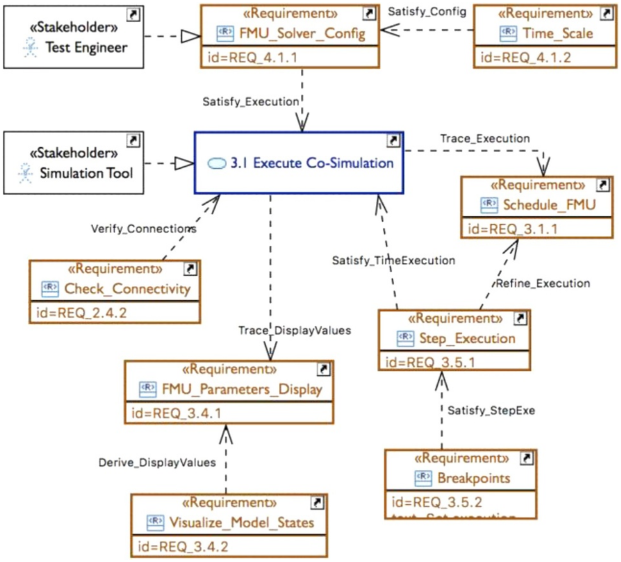


Fig. 7. Relation of functional and non-functional requirements for simulation tools with support for the FMI standard.

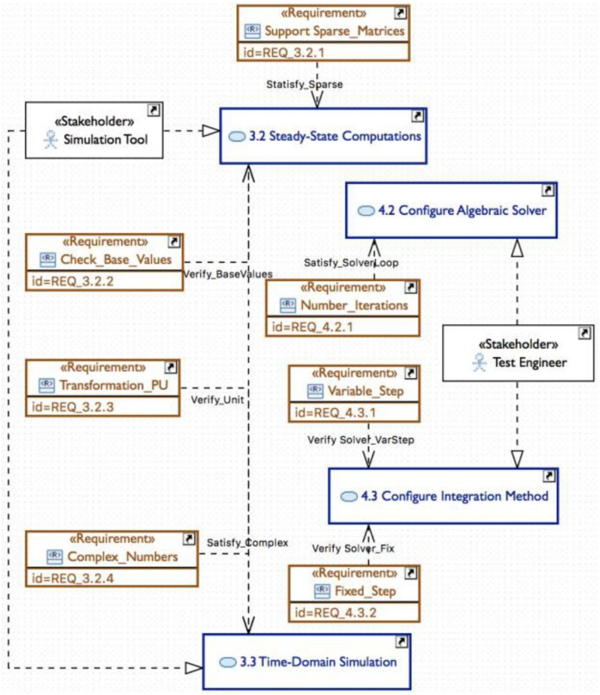


Fig. 8. Relation of functional and non-functional requirements for simulation tool, related to specific power systems functionalities.

model description and the model executable code.

The SysML diagram in Fig. 7 defines as requirements some basic state-of-the-art functionalities as requirements, such as REQ_3.4.1 for model parameter display, which is related with traceability of model behavior, i.e. graphical representation of the model behavior. Moreover, the FMU and equation-based modeling allow the application of functionalities such as the REQ_3.5.2 that affects the debugging of the model execution.

The SysML diagram on Fig. 8 shows basic state-of-the-art functionalities that might seem trivial from the power systems analysis perspective but are important to consider for the development of power system analysis tools based on the already mentioned modeling and simulation standards. For instance, it is important to consider the base values of the model, and the unit system to be utilized, as stated by REQ_3.2.2 and REQ_3.2.3. Tables 5 and 6 offer further description of these requirements.

5. Methodology and technologies towards meeting design requirements

The design of functional and non-functional requirements for the development of new power system M&S tools was described in the previous sections. An objective of those requirements is to enhance information and equation-based models for dynamic co-simulation, applying UML & SysML based specifications for the development of new power systems tools. These requirements are aimed to be the starting point of a workflow methodology for power system modeling tools and simulation tools. This methodology can be implemented with the integration of UML/SysML & FMU-based tools such as the Moka plug-in for Papyrus [40] or Modelica tools, such as OpenModelica [41].

This section aims to illustrate how the most important requirements can be met by different technologies readily available. This work serves as a proof of concept of how each of the future standards-based software architecture components can be capable of meeting the proposed designed requirements, or how existing tools can be enhanced.

Table 5
Description of non-functional requirements, related with FMI support for simulation tools.

Id	Implementation description	Req. relation
2.4.2	Correct flow of FMUs inputs and outputs, one-direction or bi-directional.	Verify REQ_3.1 and REQ_2.5
3.1.1	Application of HPC techniques (concurrent of parallel) for complex models	Trace REQ_3.1
3.4.1	Selection of model variables to trace during simulation	Trace REQ_3.1
3.4.2	Graphical representation of model variables, inputs/output, from $t_0=$ start_time to $t_f=$ stop_time	Derive REQ_3.4.1
3.5.1	Simulation execution from time t to $t+1$	Satisfy REQ_3.1, Refines REQ_3.1.1
3.5.2	Assignment of breakpoints within the model's equations	Satisfy REQ_3.5.1
4.1.1	Setting the FMU solver's parameters.	Satisfy REQ_3.1
4.1.2	Synchronize the FMU solver with the simulation tool's master simulation routine.	Satisfy REQ_4.1.1

Table 6
Description of non-functional requirement, regarding power systems related methods, for simulation tools.

Id	Implementation description	Req. relation
3.2.1	Computation of sparse matrices	Satisfy REQ_3.2
3.2.2	Supply values to parameters affecting the whole system	Verify REQ_3.2 & REQ_3.3
3.2.3	Define the units of the model's variables (per unit or SI)	Verify REQ_3.2 & REQ_3.3
3.2.4	Support for complex values within calculation of DAEs	Verify REQ_3.2 & REQ_3.3
4.2.1	Specify a finite number of iterations for the master simulation routine	Satisfy REQ_4.2
4.3.2	Computation of model equations trajectory with a defined number of steps until $t_f = stop\_time$	Satisfy REQ_4.3
4.3.2	Computation of model equations trajectory at every k steps until $t_f = stop\_time$	Satisfy REQ_4.2

5.1. *Methodology for consolidated information and equation-based modeling*

The first step of this methodology is the application of Model Developer requirements (see Table 1) for network modeling. The Model Developer could use a Modeling Tool to build the network dynamic simulation model, with an equation-based language (REQ_1.5). The utilization of the CIM packages IEC 61,970-(301,302) provides the necessary parameter information of the network model and complies with the REQ_1.3 and REQ_1.4. The utilization of an equation-based power system library, such as the OpenIPSL, allows to comply with REQ_1.5. The model information is required to complete the components' behavior (REQ_1.6) and their mathematical representation (REQ_1.5) of the dynamic model.

The second step consists on using the modeling standards stated before to build a complete simulation model. The Modeling Tool should be capable of automatically creating a network model, with the use of mapping between the information model and the equivalent equation-based model. This could be achieved by a transformation mechanism that can connect both models (REQ_2.2). An example of such a requirement can be found in [24]. This work presents a model-transformation tool that can connect the source information model with the target equation-based model, obtaining a final dynamic simulation model with the proper parameter values and components equations. A Modeling Tool must be able to manipulate the information model and create different profiles (see REQ_1.4 and REQ_2.1), which are needed by the transformation tool.

As an example of the previous steps, Fig. 9. shows a brief description of how a subset of the CIM is used to build a target simulation model in Modelica. The model-2-model (M2M) transformation tool reads the model CIM data and builds an equivalent benchmark model in Modelica, ready to be used for dynamic simulation. In this example, the utilization of those standards is gathered with the requirements REQ_1.4.1 & REQ_2.1.2 (for IEC 61,970 semantics and syntax), REQ_1.4.2 & REQ_2.1.1 (for ISO 15,926 semantics and syntax), and REQ_1.5.1 & REQ_2.3.1 (for Modelica semantics and syntax).

5.1.1. *Support for multi-domain modeling*

Because of the Object-Oriented nature of the modeling standards previously discussed, it is possible to scale those models with further information from other engineering domains and develop more detailed power system models. The use of transfer-function representation to express the behavior of some components' models could be replaced by more detailed equations and model representation if different domains are combined. This is the main reason why REQ_1.1 and REQ_1.2 have been designed. The ISO 15,926–8 [42] could be used to extend the CIM information model of power plant devices with more detailed information [43].

An example of multi-domain modeling in Modelica (with references to REQ_1.5) can be seen in Fig. 10, with simulation results in Fig. 11. The benchmark model [44] consists on the model of a gas turbine from the thermo-mechanical domain [45]. A power grid model and a power-system-domain generator model [36]. The governor model for the turbine is also a model from the power system domain (it uses a transfer function modeling abstraction). The load component of this benchmark model is a variable load model used to illustrate the REQ_4.4. The gas turbine model has similarities with the model description from the ISO 15,926. This serves to show

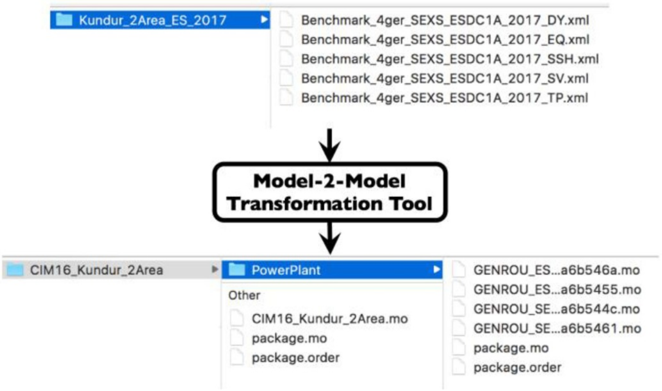


Fig. 9. Concept of Model-2-Model transformation, from a source model in CIM to a target dynamic simulation model in Modelica.

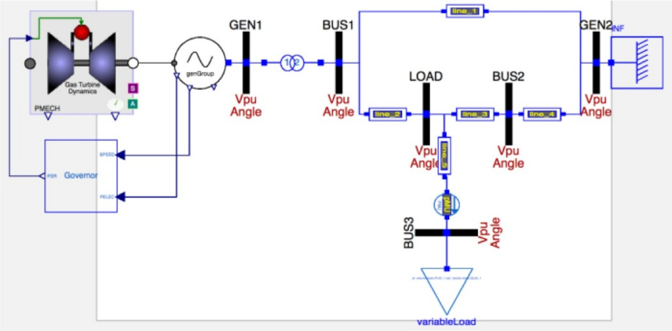


Fig. 10. A Gas Turbine mechanical domain connected a Single-Machine Infinite Bus network model of the power system domain.

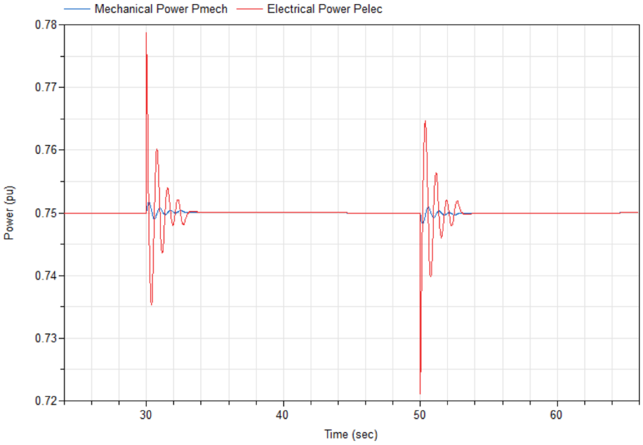


Fig. 11. The plot shows the electrical and mechanical powers in p.u.. The variable-step solver DASSL has been used, with a tolerance of 1e-06, during 100 s. A load variation of 0.2 p.u. has been applied at 30 s and reset at 50 s.

how this standard is a good choice to complement the CIM information model for gas turbines.

Observe that, in the proposed approach, the power system network topology is specified in the TP profile of the CIM model. When changes are made to the topology of the network these are reflected in the corresponding TP file (e.g. in Fig. 9, “Benchmar_4 ger_SEX_ESDC1A_2017_TP.xml”). While only a simple power network with fixed topology has been illustrated herein, previous work shows that it is possible to apply the model transformation process to larger power networks [53]. Finally, in case different topology variants are not included via TP profiles, the node-breaker information of the CIM model can be used to generate them [54].

5.2. Model simulation methodology

The previous section showed steps of the methodology for creating dynamic models. The next step in the overall M&S methodology is the execution of the dynamic simulation models, as pure Modelica models or in combination with other components distributed as FMU blocks. This step of the methodology is described in the following three subsections.

5.2.1. Modelica-based power systems simulation requirements

A simulation software that can comply with the requirements for simulation tools (Figs. 6 and 8) should be able to process either CIM models or Modelica models. Works such as [46,54] propose topology processing methods that rely on the EQ and SSH profile. This results on the processing of the TP and SV profiles that illustrate the steady state condition of the network and comply with REQ_3.2.

Those results are used to initialize and provide different initial conditions for the Modelica network models, so they can be executed for dynamic simulations. Existing Modelica compilers, such as OpenModelica or Dymola, show that time-domain analysis can be executed with pure Modelica models (see REQ_3.3) composed only by power system network models [36] or multi-domain network models composed by components from different libraries, as shown in [42].

A simulation software based on Modelica must be able to support different Modelica compilers, which provide different set of integration methods. Thus, based on the software prototype presented in [23] the REQ_4.3, REQ_4.3.1, to support fix-step methods, and the REQ_4.3.2, to support variable-step methods are designed.

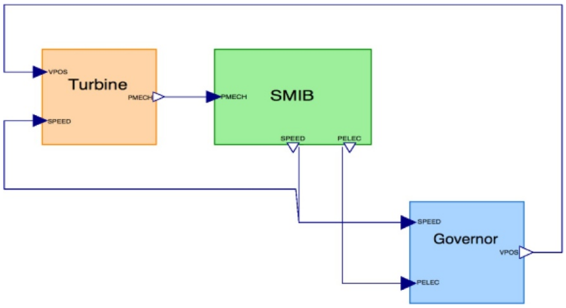


Fig. 12. Block representation of the multi-domain Modelica model from Fig. 10. It has been executed by the Dymola compiler and by Simulink.

5.2.2. FMI-based simulation requirements

The application of the FMI standard allows the separation of modeling tools and simulation tools. By applying the REQ_2.4 and the REQ_2.4.1 a Modeling tool could use the FMI API to generate FMU blocks that comprise component models, which can be used to complement other power system models compatible with the FMI. Thus, it is possible to implement simulation tools for model execution supporting co-simulation (stated by the REQ_3.1).

The FMI standard provides the capability to execute the simulation of a model with the solver implementation provided by the simulation tool or use the own FMU's solver for co-simulation capabilities. It can also be combined with the Modelica language and benefit from a Modelica-compliant tool's compiler capability to execute the simulation model. An example of this is shown in Fig. 12, with the resulting simulations shown in Fig. 13. The SMIB network has been separated into multiple FMUs: one FMU for the turbine component, an FMU for the governor and an FMU for the network. With this setup, different types of turbines and governors could be simulated within the same network configuration by replacing the FMUS corresponding to each component.

A model in FMI makes accessible the parameters of the models and while it hides its implementation at the same time. This reason makes FMU models suitable for model exchange. If measurements are available, the FMU network model could be simulated and calibrated to fit the available measurements (complying with REQ_3.7 and REQ_3.8). Thus, a simulation tool could integrate calibration capabilities, such as in [39], comply with the requirements designed in and Figs. 5 and 7. It is important to notice from Fig. 13. that the use of FMI puts strong demands on the means for initialization of specific tools, therefore, the use of FMI needs careful verification.

6. Discussions

6.1. Requirements engineering

Requirements Engineering is being applied in the design and implementation of physical systems [13]. Current methods from RE allow to define the Modeling & Simulation functionalities in isolation from the development of the model. In the power systems domain, those functionalities have been mostly studied as part of the same model, i.e. the development of models have been attached to the development of the M&S tools. The requirements presented in this work help to argue in favor the separation of both modeling and simulation stages in isolation from each other. This is possible with the application of Model-Based Software Engineering

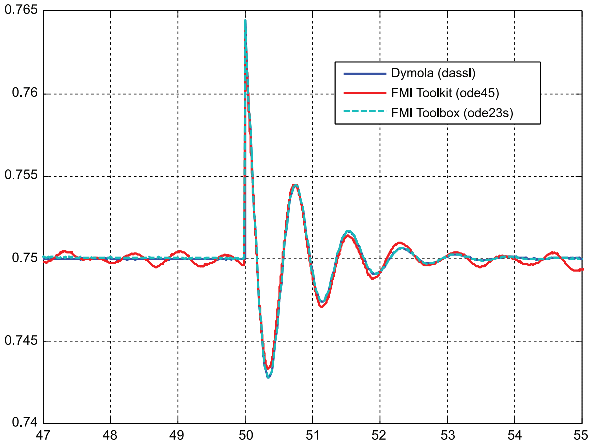


Fig. 13. The plot shows the electrical power in p.u. The variable-step solver, DASSL from Dymola, Ode45 from FMIToolkit and Ode23s from the FMIToolbox have been used, with a tolerance of 1e-06, during 100 s. Figure shows the differences of these three solvers, on a presence of a load variation of 0.2 p.u. has been applied at 30 s and reset at 50 s.

principles and the use of the FMI standard used and discussed in this work. For example, UML tools such as Papyrus + Moka [40] can implement a FMI API to import FMU blocks into a UML model, or vice versa. This could lead to develop CIM-based tools that can export their information into FMU blocks. Another example is the use of FMI compilers or Modelica compilers with support for FMUs to execute and represent the behavior of those models. This makes possible to the user to choose and exploit the models in different computer environments.

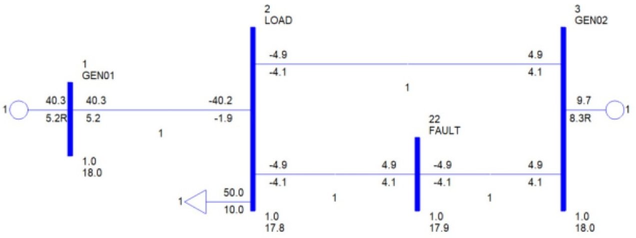
With the adoption of RE, better interoperability of tools and models could also be achieved. First, the use of UML or SysML does not depend on how the system is implemented. They define the information and behavior to be implemented to support the available standards, without strictly specify which programming language should be used for the tools implementation. Second, further application of the RE with SysML and Modelica could help to define different restrictions within the same model, provide more specific model details and improve their interoperability. Models developed in Modelica are translated into executable code by an equation-based compiler. Thus, all the equations and restrictions of the model could be able to be interpreted in different simulation environments.

6.1.1. Simulation accuracy and efficiency

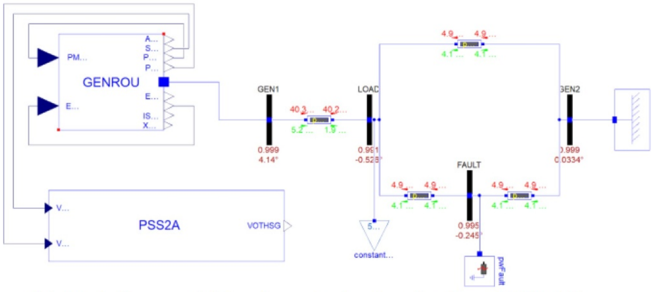
6.1.1.1. Simulation accuracy. Traditional power system modeling simulation tools are commonly considered by their users as the ground truth, hence, the approach considered in the authors' previous work is to assure the traditional tool users that the results from the model transformation, i.e. from CIM to Modelica, will result in identical simulation results when comparing the traditional tool against the results from "*a*" Modelica tool. This is called "*software-to-software*" verification. In this work we do not make any assumption on which Modelica tool can be used, but for the sake of argument the scenario below illustrates the use of two Modelica tools and the PSS/E software (i.e. the traditional tool).

The Modelica library used in this work is the OpenIPSL [1]. This library provides the component models that are mapped to the CIM/CGMES model in order to provide the mathematical equations representing it. A number of these models have been verified against PSS/E in order to obtain the same model accuracy. This procedure is illustrated as follows. Using PSS/E and the OpenIPSL library a sample power system is assembled as shown in Fig. 14. The results in Fig. 15 are obtained when simulating a fault in the system with the simulation settings of Table 7. Observe that the solvers for each tool are different and need to be carefully configured for this assessment. The library developers analyze the error between the simulation results and determine the error in order to decide whether or not the model has the same modeling accuracy. In the example in Fig. 15, it can be concluded that the simulation results are identical between the three different tools. This guarantees that the modeling accuracy with respect to traditional tools will be achieved, providing flexibility on which Modelica tool to use. However, this does require that the Modelica library is adequately and routinely maintained. More complex examples involving larger scale models have been addressed in the author's previous work, see [15].

6.1.1.2. Simulation efficiency. Maintaining adequate levels of simulation efficiency in the proposed approach is left entirely to the specific Modelica tool provider / vendor. This means that while the model is defined in a standardized modeling language, the simulation approach is not standardized, as evident in Table 7 where different solvers are used to simulate the model in different



(a) Single Line Diagram from PSS/E



(b) Modelica model implemented using the OpenIPSL Library

Fig. 14. A single machine infinite bus model including a round rotor generator model and a PSS2A power system stabilizer implemented in (a) PSS/E and (b) Modelica using the OpenIPSL library.

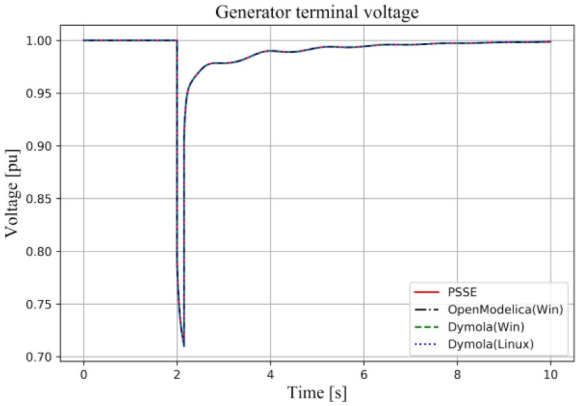


Fig 15. Simulation results in three different simulation environments: PSS/E (reference), OpenModelica and Dymola.

Table 7
Simulation configuration settings in different tools.

Software	PSS/E	Dymola	OpenModelica
Output interval length	0.001	0.001	0.001
Integration algorithm	Modified Euler	Rkfix2	rungekutta
Tolerance	0.0001	0.0001	0.0001

tools. This is attractive from a user's perspective because while the models will be defined in the different standards proposed, they are not “locked-in” to a specific simulation tool or method. In addition, this would provide quantifiable means for the tool providers to compete based on the quality and performance of these tools.

One potential concern of traditional tool users would be simulation speed/performance when compared either to Modelica or FMI-based simulators. This is because while traditional tools have the benefit of more than 40 years of development (which includes the use of domain specific heuristics to gain simulation performance), generic simulation tools based on Modelica may be exposed to some limitations. However, this gap is quickly blurring thanks to efforts from the Modelica community by applying advanced numerical simulation strategies. Examples of strategies that have been successfully realized in Modelica tools, such as Dymola and the 3DEXPERIENCE platform, involve: multirate simulation [47], mixed-mode simulation [47,48], model decoupling and parallel execution [49], and sparse solvers [50]. These efforts have proven valuable for the use of Modelica tools for the solution of large-scale power system models, as it can be evidenced in [51] where a new Differential and Algebraic equation solver has been implemented which is capable of simulating with comparable performance for the simulation of complex power system models and even performing faster than the de-facto simulation tools (i.e. PSS/E). Moreover, thanks to speed improvements, Modelica-based tools are now being used for industrial-scale simulations at the French power system operator, RTE [52].

7. Conclusions and further work

7.1. Conclusions

The Use Case methodology has been applied to derive functional and non-functional requirements for the development of new power system modeling and simulation tools. Requirements for modeling tools and simulation tools have been separated in two main use cases for better distinction of new functionalities. The Use Case Methodology has been extended with the use of SysML Requirements diagrams as a support for the description of the requirements of each use case.

From the modeling perspective, the need to exchange steady-state and dynamic information of cyber-physical power systems models is increasing. The presence of Renewable Energy Sources (RES) is driving the need to assess power system dynamics with more rigurosity, and to analyze new power generation, transmission and distribution technologies which leads to more complex systems models across system simulation tools. It is necessary to provide complete information and equation-based modeling to better represent the behavior of such complex systems. In this case, the adoption of the CIM/CGMES can be complemented with more detailed information from other standards, such as the ISO 15,296 for power plant operation. Equation-based modeling languages have been shown as a good complement, providing the mathematical behavior that an information modeling language cannot provide on its own.

From the dynamic simulation perspective, the Modelica language and the FMI standard for model exchange and co-simulation, can also motivate existing commercial tool vendors to support the definition of Modelica (equation-based) user defined models, that could replace the much simpler and ambiguous block diagram representations. Moreover, the Modelica language can be used to define all available standard models and create additional validation models to verify the behavior of different tools.

7.2. Future work

The challenge expressed by this work is to encourage the development of power system tools to integrate advances from the field of M&S in system-of-systems engineering to fully comply with standard modeling languages, allowing multi-domain modeling and co-simulation of cyber-physical power systems.

Further work is focused on a formalization for information-based modeling and development of new profiles. The adoption of the FMI standard implies that mathematical solvers need to adapt to handle co-simulation, parallelization, etc., which may require the use of different solver tolerances and simulation time-steps. Finally, with the emergence of the SSP standard [37], the requirements described in this paper can be expanded to enhance the interoperability of the future cyber-physical power system tools. The SysML model of the requirements proposed in this paper are available on-line at https://github.com/ALSETLab/sysml.powersystems.framework, so that it can be modified and expanded by other researchers in the future.

Acknowledgements

This work was supported in part by the ITEA3 OpenCPS project number 14018, in part by the New York State Energy Research and Development Agency (NYSERDA) through the Electric Power Transmission and Distribution (EPTD) High Performing Grid Program under agreement number 137940, in part by the Engineering Research Center Program of the National Science Foundation and the Department of Energy under Award EEC-1041877 and in part by the CURENT Industry Partnership Program.

In Memoriam

The publication of this article is dedicated to the memory of the first author who passed on September 21st, 2019 while the original manuscript of this paper was being completed.

Supplementary materials

Supplementary material associated with this article can be found on-line at https://github.com/ALSETLab/sysml.powersystems.framework.

Appendix 1 – Rationale for the Choice of SysML

As it has been stated in Section II.C.1, it is necessary to adopt GPML languages to represent the requirements and improve the interoperability of cyber-physical power system modeling and simulation. While UML was selected due to its synergy with existing grid standards, SysML was deemed a good choice due to the following reasons:

- The language has an orientation to Systems Engineering (not to Software Engineering).
- It provides an explicit and simple way to define non-functional requirements (i.e. Requirements Diagram).
- It enables the integration of the semantic models required for the suitable configuration of the modeled systems (parameter component) with those required to represent their behavior and physical structure (mathematical component).
- Its block-based modeling orientation establishes an easy-to-interpret language for the domain expert engineers responsible to design and undertake simulation-based studies with the models (i.e. the Model Developer).

Fig. A.1 and Table A.1 are intended to illustrate and extend the selection criteria of SysML in this work through a comparison with other GPMLs.

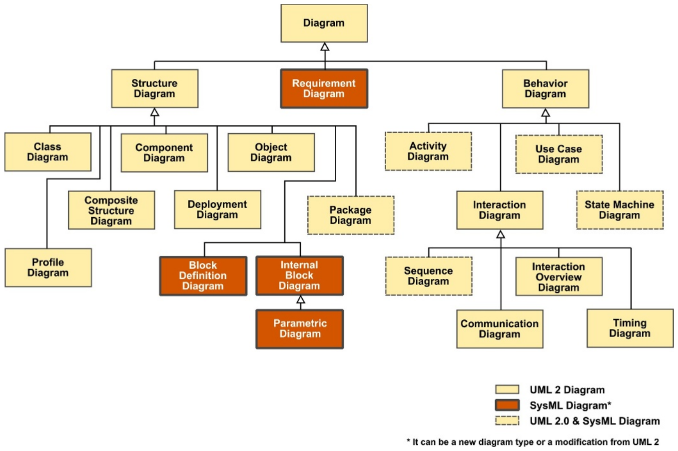


Fig. A.1. Graphical comparison between the taxonomies of the UML 2 and SysML diagrams.

Table A.1
Modeling features of three General-Purpose Modeling Languages (GPML).

	UML	SysML	EXPRESS
Structure diagrams	Package diagram Component diagram Class diagram Deployment diagram Composite structure diagram Object diagram Profile diagram	Package diagram **Block definition diagram** **Internal block diagram** **Parametric diagram**	EXPRESS-G diagram or Text-based description
Behavior diagrams	Use case diagram Sequence diagram Activity diagram Timing diagram State machine diagram Interaction overview diagram Communication diagram	Use case diagram Sequence diagram Activity diagram State machine diagram	N/A
Requirements	N/A	**Requirement diagram**	N/A
Current version	2.5.1 (dec 2017)	1.6 (Dec **2019**)	1.1 (May 2015)
Target application	**Software** modeling	**Systems** engineering (ISO/IEC 15,288)	**Systems** engineering **Information modeling** language
Official specifications website	https://www.omg.org/spec/UML/	https://www.omg.org/spec/SysML/	https://www.omg.org/spec/EXPRESS/ ISO 10,303–11.2:2004
Metamodel used by	CIM (Common Information Model) IEC 61,970 / IEC 61,968 / IEC 62,325 IEC 62,559		ISO 15,926

References

[1] European network of transmission system operators of electricity [On-line], https://www.entsoe.eu/Pages/default.aspx.

[2] ENTSO-E, “Making non-mandatory requirements at European level mandatory at national level ENTSO-E guidance document for national”, [On-line], https://electricity.network-codes.eu/network_codes/cnc/cnc-igds/, 2016.

[3] Ch. Ivanov, T. Saxton, J. Waight, M. Monti, G. Robinson, Prescription for interoperability: power system challenges and requirements for interoperable solutions, IEEE Power Energy Mag. 14 (1) (Jan.-Feb. 2016) 30–39, https://doi.org/10.1109/MPE.2015.2485798.

[4] CEN-CENELEC-ETSI Smart Grid Coordination Group. “Smart grid reference architecture”. November 2012. [On-line] Available: http://ec.europa.eu/energy/gas_electricity/smartgrids/doc/xpert_group1_reference_architecture.pdf.

[5] EnergyNet.dk, “Technical regulations 3.2.2. for pv power plants above 11KW”, July 2016, Revision 4 [On-line], Available: https://en.energinet.dk/Electricity/Rules-and-Regulations/Regulations-for-grid-connection.

[6] C. Arora, M. Sabetzadeh, L.C. Briand, F. Zimmer, Extracting domain models from natural-language requirements: approach and industrial evaluation, MODELS (2016) 250–260, https://doi.org/10.1145/2976767.2976769 ACM.

[7] H. Kaindl, S. Brinkkemper, J.A. B. Jr., B. Farbey, S.J. Greenspan, C.L. Heitmeyer, J.C.S. do Prado Leite, N.R. Mead, J. Mylopoulos, J.I.A. Siddiqi, Requirements engineering and technology transfer: obstacles, incentives and improvement agenda, Requir. Eng. 7 (3) (2002) 113–123.

[8] ENTSO-E CIM inter-operability tests, 2015. [On-line] Available: https://www.entsoe.eu/major-projects/common-information-model-cim/interoperability-tests/Pages/default.aspx.

[9] E. Raymond, The cathedral and the bazaar, Knowl., Technol. Policy 12 (3) (1999) 23–49 https://doi.org/10.1007/s12130-999-1026-0.

[10] A.A. van der Meer, Cyber-physical energy systems modeling, test specification, and co-simulation based testing, 2017 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES), Pittsburgh, PA, 2017, pp. 1–9, , https://doi.org/10.1109/MSCPES.2017.8064528.

[11] L.E.G. Martins, T. Gorschek, Requirements Engineering for Safety-Critical Systems: An Interview Study with Industry Practitioners, IEEE Transactions on Software Engineering, vol. 46, 2020, 1st April, pp. 346–361, , https://doi.org/10.1109/TSE.2018.2854716.

[12] MODRIO: model driven physical systems operation, [On-line], Available: https://www.modelica.org/external-projects/modrio.

[13] L. Buffoni, P. Fritzson, Expressing requirements in Modelica, Proceedings of the 55th International Conference on Simulation and Modeling, Aalborg, Denmark, 2014, October 21-22, https://doi.org/10.11128/sne.25.tn.10314.

[14] M. Otter, N. Thuy, D. Bouskela, L. Boufoni, H. Elmqvist, P. Fritzson, A. Garro, A. Jardin, H. Olsson, M. Payeleville, W. Schamai, E. Thomas, A. Tundis, Formal requirements modeling for simulation-based verification, Proceedings of the 11th International Modelica Conference, Versailles, France, 2015, https://doi.org/10.3384/ecp15118625.

[15] Open cyber-physical system model-driven certified development, [On-line], Available: https://opencps.eu.

[16] Functional mock-up unit interface, [On-line], Available: http://fmi-standard.org.

[17] M. dos Santos Soares, J. Vrancken, Model-Driven user requirements specification using SysML, J. Softw. 3 (6) (2008) 57–68, https://doi.org/10.4304/jsw.3.6.57-68.

[18] D.M. Berry, D. Damian, A. Finkelstein, D. Gause, R. Hall, A. Wassyng, To do or not to do: if the requirements engineering payoff is so good, why aren't more companies doing it? RE ’05: Proceedings of the 13th IEEE International Conference on Requirements Engineering (RE’05), Washington, DC, USA, IEEE Computer Society, 2005, p. 447.

[19] Systems modeling language specification OMG (2015) [On-line] http://www.omg.org/spec/SysML/1.4/.

[20] Papyrus modeling environment, [On-line], Available: https://www.eclipse.org/papyrus/.

[21] OpenModelica modeling and simulation environment, [On-line], Available: https://openmodelica.org.

[22] M. Gottschalk, M. Uslar, C. Delfs, The Use Case and Smart Grid Architecture Approach, Springer, 2017, https://doi.org/10.1007/978-3-319-49229-2.

[23] F.J. Gómez, L. Vanfretti, S.H. Olsen, A modelica-based execution and simulation engine for automated power system model validation, IEEE PES Innovative Smart Grid Technologies, Europe, Istanbul, 2014, pp. 1–6, , https://doi.org/10.1109/ISGTEurope.2014.7028828.

[24] F.J. Gómez, L. Vanfrett and S.H. Olsen, “CIM-Compliant power system dynamic model-to-model transformation and Modelica simulation”, in IEEE Trans. Ind. Inf., https://doi.org/10.1109/TII.2017.2785439 .

[25] F. Milano, Power Systems Modeling and Scripting, Springer, 2010, https://doi.org/10.1007/978-3-642-13669-6.

[26] F. Milano, R. Zárate-Miñano, On the stochastic nature of deterministic power system models for dynamic analysis, IEEE PES General Meeting, Boston, MA, 17-21

July 2016.

[27] A.W. McMorran, G.W. Ault, C. Morgan, I.M. Elders, J.R. McDonald, A common information model (CIM) toolkit framework implemented in JAVA, IEEE Trans. Power Syst. 21 (1) (Feb. 2006) 194–201, https://doi.org/10.1109/TPWRS.2005.857846.

[28] M. Uslar, M. Specht, S. Rohjans, J. Trefke, J.M. Gonzalez, The Common Information Model CIM: IEC 61970, 61968 and 62325, Springer, Heidelberg, 2012, https://doi.org/10.1007/978-3-642-25215-0.

[29] ENTSO-E common grid model exchange standard (CGMES), [On-line], 2014, https://www.entsoe.eu/digital/common-information-model/cim-for-grid-models-exchange/.

[30] L.O. Osterlund, K. Hunter, K. Demaree, M. Goodrich, A. McMorran, B. Iverson, T. Kostic, Under the hood: an overview of the common information model data exchanges, IEEE Power Energy Mag. 14 (1) (Jan.-Feb. 2016) 68–82, https://doi.org/10.1109/MPE.2015.2485859.

[31] David Leal, ISO 15926 "Life cycle data for process plant": an overview, Oil & Gas Sci. Technol. - Rev. IFP 60 (4) (2005) 629–637.

[32] Web ontology language (OWL), [On-line], Available: https://www.w3.org/OWL/.

[33] Resources description framework (RDF), [On-line], Available: https://www.w3.org/RDF/.

[34] M. Bonvini, A. Leva, A modelica library for industrial control systems, Proceedings of the International Modelica Conference, Munich, Germany, 2012, pp. 477–484, , https://doi.org/10.3384/ecp12076477.

[35] M. Bonvini, F. Donida, A. Leva, Modelica as a design tool for hardware-in-the-loop simulation, Proceedings of the 7th International Modelica Conferenca, September 20-22, 2009, pp. 378–385, , https://doi.org/10.3384/ecp09430119.

[36] M. Baudette, M. Castro, T. Rabuzin, J. Lavenius, T. Bogodorova, L. Vanfretti, OpenIPSL: open-Instance power system library — update 1.5 to iTesla power systems library (iPSL): a Modelica library for phasor time-domain simulations, SoftwareX 7 (2018) 34–36 ISSN 2352-7110 https://doi.org/10.1016/j.softx.2018.01.002.

[37] J. Köhler, H.-.M. Heinkel, P. Mai, J. Krasser, Deppe M., M. Nagasawa, Modelica association project: system structure and parametrization – Early Insight, Proceedings of the 1st Japanese Modelica Conference, Tokyo, Japan, May 23-24, 2016, https://doi.org/10.3384/ecp1612435.

[38] M. Sjölund, F. Casella, A. Pop, A. Asghar, Integrated debugging of equation-based models, Proceedings of the 10th International Modelica Conference, 2014, pp. 195–204, , https://doi.org/10.3384/ecp14096195.

[39] L. Vanfretti, M. Baudette, A. Amazouz, T. Bogodorova, T. Rabuzin, J. Lavenius, F.J. Gómez, RaPId: a modular and extensible toolbox for parameter estimation of Modelica and FMI compliant models, SoftwareX 5 (2016) 144–149 https://doi.org/10.1016/j.softx.2016.07.004.

[40] Papyrus Moka plug-in [On-line], Available: https://wiki.eclipse.org/Papyrus/UserGuide/ModelExecution.

[41] P. Fritzson, P. Bunus, Modelica, a general object-oriented language for continuous and discrete-event system modeling, Proceedings of the 35th Annual Simulation Symposium, 2002, pp. 14–18.

[42] ISO/TS 15926-8:2011, "Industrial automation systems and integration — integration of life-cycle data for process plants including oil and gas production facilities — part 8: implementation methods for the integration of distributed systems: web ontology language (OWL) implementation", 1st Edition, October 2011. Online: https://www.iso.org/standard/52456.html.

[43] F.J. Gómez, M. Aguilera Chaves, L. Vanfretti, S.H. Olsen, Multi-Domain semantic information and physical behavior modeling of power systems and gas turbines expanding the common information model, IEEE Access 6 (2018) 72663–72674, https://doi.org/10.1109/ACCESS.2018.2882311.

[44] M. Aguilera, L. Vanfretti, F.J. Gomez, Experiences in power systems multi-domain modeling and simulation with Modelica & FMI, Proceedings of 2018 Workshop on Modeling and Simulation of Cybyer-Physical Energy Systems, April 10, 2018.

[45] F. Casella, A. Leva, Modelling of distributed thermo-hydraulic processes using Modelica, Proceedings of the MathMod '03 Conference, Wien, Austria, February 2003 https://doi.org/10.1080/13873950500071082.

[46] S. Shukla, M.G. Yao, Efficient method for extracting network information from the CIM asset model, 2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT), Minneapolis, MN, 2016, pp. 1–5, , https://doi.org/10.1109/ISGT.2016.7781190.

[47] B. Thiele, M. Otter, S.E. Matsson, Modular multi-rate and multi-method real-time simulation, Proceedings of the 10th International Modelica Conference; March 10-12; 2014, Lund; Sweden, 2014-03-10, pp. 381–393 VolumeISSN 1650-3740.

[48] A. Schiela, H. Olsson, Mixed-mode integration for real-time simulation, Proceedings of Modelica 2000 Workshop, Sweden, Lund University, Lund, Oct. 23-24, 2000, pp. 69–75 2000.

[49] H. Elmqvist, S.E. Matsson, H. Olsson, Parallel model execution on many cores, Proceedings of the 10th International Modelica Conference; March 10-12; 2014, Lund; Sweden, 2014-03-10, pp. 363–370 ISSN 1650-3740.

[50] W. Braun, F. Casella, B. Bachmann, Solving large-scale Modelica models: new approaches and experimental results using openModelica, Proceedings of the 12th International Modelica Conference, Prague, Czech Republic, 132 May 15-17, 2017, pp. 557–563 Volume2017-07-04ISSN 1650-3740.

[51] E. Henningsson, H. Olsson, L. Vanfretti, DAE solvers for large-scale hybrid models, Proceedings of the 13th International Modelica Conference, Regensburg, Germany, 2019, p. 12 2019-02-01ISSN 1650-3740 http://dx.doi.org/10.3384/ecp19157491.

[52] K. Abdelhak, B. Bachmann, F. Rosiere, A. Guironnet, Experience on the use of the DAE mode in industrial power system simulations, OpenModelica Workshop, Linkoping University, February 3, 2020Online: https://openmodelica.org/images/M_images/OpenModelicaWorkshop_2020/PresentationOMDAE.pdf.

[53] L. Vanfretti, S.H. Olsen, V.S. Narasimham Arava, G. Laera, A. Bidadfar, T. Rabuzin, S.H. Jakobsen, J. Lavenius, M. Baudette, and F.J. Gómez-López, An open data repository and a data processing software toolset of an equivalent Nordic grid model matched to historical electricity market data, data in brief, Available online 13 February 2017, ISSN 2352-3409.

[54] T. Kovac, Translating CIM model to bus-branch model, 2019 27th Telecommunications Forum (TELFOR), 2019 27th Telecommunications Forum (TELFOR), Belgrade, Serbia, 2019, pp. 1–4.