GAMES: a General-purpose Architectural model for Multi-Energy System engineering applications

GAMES: a General-purpose Architectural model for Multi-Energy System engineering applications

Luca Barbierato*, Daniele Salvatore Schiera*, Edoardo Patti*, Enrico Macii*, Enrico Pons*, Ettore Francesco Bompard*, Andrea Lanzini*, Romano Borchiellini* and Lorenzo Bottaccioli*
*Energy Center Lab, Politecnico di Torino, Turin, Italy. Email: name.surname@polito.it

***Abstract*—The growing interest in Multi-Energy Systems (MES) leads the scientific community to implement innovative technologies to analyse and simulate these complex systems. Two main research trends are identified in such analysis: *i)* improve the usability and capability of preexisting reference architectures in the energy field to cope with high-level use case descriptions, and *ii)* study the interoperability of such reference architectures in order to increase systematic and functional analysis of MES use cases. *GAMES* is a a general-purpose architectural model for MES engineering application. The aim is twofold: *i) GAMES* implements an extension of Smart Grid Architecture Model (SGAM) to cope with MES use case descriptions, and *ii)* it offers a methodology to deal with a systemic description of the use case through a combination of UML and SysML integrated in the proposed architectural model. Furthermore, *GAMES* will allow the implementation of Domain Specific Language (DSL) and hardware configuration for the specific components described by UML/SysML diagrams. Compared to other solutions, *GAMES* allows to assess both research trends in a single hierarchical ICT infrastructure.**

***Index Terms*—Reference Architecture, MBSE, MDA, SGAM, Multi-Energy System (MES), System Engineering**

I. Introduction

Multi-Energy System (MES) analysis will take into account heterogeneous energy vectors (e.g. electricity, heat exchanging fluids, natural gas, and possibly hydrogen) interacting together to globally optimise different operational perspectives of each energetic supply chain [1]. MES evolves the common vertical analysis of each energy vector towards a more holistic approach. MES infrastructures are difficult to be analysed comprehensively without an effective hardware deployment to study the dynamics involved. To avoid huge investments in hardware components, system engineering provides a viable solution to design MES. It provides several tools to sketch use cases based upon Model-Based System Engineering (MBSE) [2]. MBSE offers an high-level description of use cases involving physical, Information and Communication Technologies (ICT), functional, economical and social perspective through reference architectures. Reference architectures allow to share tacit knowledge among different stakeholders accounting the above-mentioned perspectives in parallel. These different viewpoints jointed together allow a consistent final description of use case objectives. However, architecture models present weaknesses in developing and evaluating functional and systemic behaviours of components involved in such use cases. Software simulation helps addressing dynamic and functional analysis of complex systems. Nonetheless, MES could be depicted as a System-of-Systems (SoS) in which systems are interconnected to fulfill composite operations and create synergies. Thus, the aforementioned solutions present difficulties in scaling up the complexity of the system. Co-simulation improves stand-alone simulation capabilities to deal with large scale complex systems. Such techniques allow the interconnection of distributed stand-alone simulators, and the integration of real-world devices enabling Hardware-In-the-Loop (HIL) simulations. A common effort between MBSE and the functional software simulation approach is needed to follow a MES designer into the difficult process involving use case definition, systemic analysis, and operational assessment. Literature analysis lacks of solutions to couple such diverse analysis.

In this paper, we present *GAMES*, a *General-purpose Architectural model for MES engineering applications* following *"from the black-box to the white-box"* approach. The presented architectural model integrates and extends the Smart Grid Architectural Model (SGAM) to deal with the definition of MES use cases. Thus, the proposed solution enables a modular methodology where different aspects of a MES can be analysed altogether with high-level details of the use case description, exploiting UML to characterise each component involved as a *black-box*. Furthermore, *GAMES* integrates SysML to enable a systemic description of the MES component and their interconnections, following a *grey-box* approach. This will allow to translate the systemic description of components involved in MES use cases into Domain Specific Languages (DSL) code of each software simulator involved in the co-simulation scenario and hardware configuration files for the interconnection of real-world devices, allowing a *white-box* analysis of software and hardware involved in the MES use case.

The structure of this paper is as follows. Section II lists three main challenges and objectives we identified to define the proper characteristics of a valuable framework to plan, design, develop and test a MES use case. Section III provides a literature analysis of already implemented techniques that address the above-described challenges in different technology fields related to energy systems. Section IV presents the *GAMES* architectural model with the description of the hierarchical methodology proposed to solve the identified challenges. Finally, Section V reports concluding remarks.

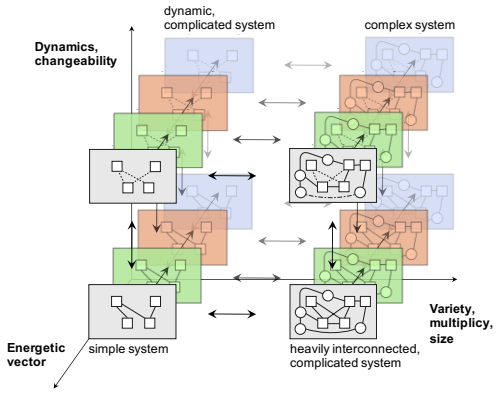

Fig. 1: Vertical and Horizontal Knowledge Integration of a generic Multi-Energy System (MES)

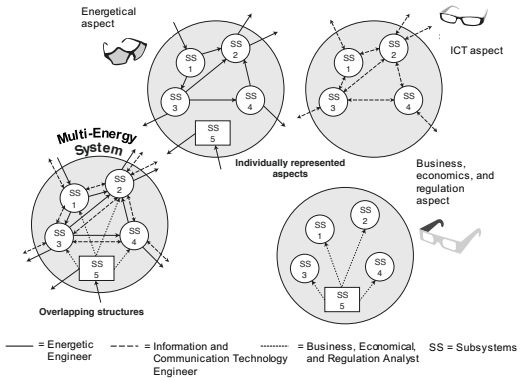

Fig. 2: Aspects of a generic Multi-Energy System design

II. Challenges and Objectives

The complex nature of Multi-Energy Systems (MES) leads to the essential needs of new framework to plan, develop and test these large scale System-of-Systems (SoS). A MES designer demands proper engineering and validation approaches, methods, concepts, and corresponding tools. This section tries to list different challenges still open in this field.

Challenge 1 *Knowledge Integration*

A Multi-Energy System (MES) is categorised as a *complex system* from a system engineering perspective [2]. Complex systems are composed by a great number of different elements employing dynamic connections. In Figure 1, two principal viewpoints can be assessed: *i)* system dynamics and changeability [3] in terms of parameters, effects and mechanisms, and *ii)* variety, multiplicy and size to represent scalability of the system. Specifically, the elements that compose a MES should be grouped by energy vector membership, represented by the third axis. Following this interpretation, MES can be stated as heavily interconnected SoS with composite system dynamic interactions belonging to a single energy vector. Due to the complexity of the analysis, we can define two different perspectives in designing a MES: i) *Vertical Knowledge Integration*, and ii) *Horizontal Knowledge Integration*. Vertical Knowledge Integration represents the different scientific contexts specific to a particular energy vector. Each of these vectors requires a specific expertise to be comprehensively analysed. Instead, *Horizontal Knowledge Integration* is aiming at a broader MES analysis by avoiding or reducing complexity for a specific energy vector, towards the concept of Simple System. From the perspective of the energy system, these viewpoints generate two mutually exclusive objectives: *i)* reducing the MES details design to address a more scalable system, or *ii)* increase the MES details, reducing the dimension of the overall system.

Integrating monitoring, management and control aspects in a MES analysis dramatically increase the order of complexity. Data signalling through ICT engineering is a requirement to enable such functionalities. A wider vision is required to define the exchange of information between all elements composing a MES. Dealing with ICT rises different needs: *i)* a common information model description evenly distributed among MES entities, *ii)* dedicated application protocols to monitor, manage and control such MES entities, *iii)* network management to send information, and iv) cyber-security and data privacy management to increase resilience of MES against cyber attacks and threats.

Lastly, great importance is played by business, financial, and regulation aspects to join the energy market and offer new commercial services (e.g. ancillary services, Demand Response (DR) and Demand Side Management (DSM)). A MES designer must choose among different alternatives to face decision-making assessment and economic feasibility of such services: *i)* outlining a business model to commercialise a service, *ii)* selecting different MES-related technologies to provide such a service, and *iii)* enabling the ICT component to fulfil the service in an optimal and secure fashion. Figure 2 shows the representation of concurrent MES aspects as a complex Cyber-Physical Energy System (CPES) with overlapping structures. Each of the possible stakeholders of a MES design introduces its own aspect of the SoS infrastructure. Thus, the first challenge corresponds the need of unified tools to support the design and analysis of MES architectures. These tools must allow effort parallelisation of MES architecture design, permitting each stakeholder to design its own aspect exploiting the others high-level SoS structures.

Challenge 2 *Design Framework*

The major concern regarding MES design lies in developing and testing of innovative technologies to ensure operational functionality, stability and safety. Systematic analysis on a formal basis is impossible when dealing with the high-level complexity of MES design. Also, hardware testing in laboratory or field trials are expensive and inflexible tools to assess a MES scenario. On the other hand, software simulation could be a viable solution which is scalable and cost-effective. Operational analysis of a MES requires experts collaboration in each different aspect presented in Figure 2. To achieve these objectives, each analysed field of technology introduces

knowledge and terminology from different MES domains. To deal with such heterogeneity, a *Design Framework* is needed to allow the interconnection of executable interdisciplinary models without involving their detailed complexity. The manual configuration of such multi-faceted scenarios in nowadays design framework can be error-prone. For instance, holonic component with composite behaviour and many interfaces with different domains (e.g. Communication Network) requires to handle complex simulation engine. Moreover, scaling the analysis from small environments up to large scale scenarios (e.g. house, district, city, region, state) could increase the complexity of different components interconnections. Thus, the design framework must be focused on the high level description of the MES scenarios, avoiding details of model coupling and interfacing. It should rely on a simple and viable modelling language shared among MES scientific community. This will ensure reduction in coding effort to achieve a valid analysis result. Models, software, simulators and hardware must be accessible among experts in the fields through a disposable shared library, to avoid their design from the scratch and enhance their reliability. Moreover, a discovery service must be integrated in the design framework to foster re-use of items already compliant with the co-simulation infrastructure. Finally, the interconnection of models into scenarios must be automated to allow a set of interconnection rules defined for each library item. Avoiding manual interconnections reduces the possible erroneous link between models, facilitating faultless and reliable scenario generation. These requirements will reduce the modelling effort needed to cope with such complex scenario.

Challenge 3 *Automated Composability*

The last challenge reflects the complex effort needed by ICT engineering to deal with the physical interconnection of MES software simulators and hardware (e.g. micro combined heat and power, thermal storages, and heat pump). Normally, software simulators employ a *Domain Specific Language* (DSL) to parametrise simulated entities in a grey-box modelling approach. DSL allows usage of *Application Programming Interface* (API) to describe different aspects of a system (i.e. inputs, attributes, control variables and outputs). Often, these software simulators follow a vertical design in different technology fields (e.g. electrical and thermal engineering, distribution and transmission grid management and energy market analysis). Hence, MES scenario developers must dedicate a steep learning curve to master these solutions. Furthermore, simulations are commonly achieved in a standalone environment without taking into account information from distributed or third-parties components. Nevertheless, this process should be implemented by considering the different interactions between the simulation environment and external sources. For instance, a simulated model could retrieve input data from an interconnected hardware to generate the desired system stimulus. Hardware setups are of great value in hardware assessment for a MES scenario but originate intrinsic criticalities in the test-bed configuration. Depending from the hardware, strict real-time constraints are introduced increasing the interconnection complexity. Moreover, the ICT integration of distributed software simulators and hardware entities requires expertise in distributed network interconnection (e.g. TCP/UDP Socket Management). An *Automated Composability* process could address these issues. Relying on the above-mentioned design framework, a dedicated compiler could generate each specific DSL code related to a particular component of the MES scenario, either hardware or software and could offer a self-regulating distributed network interconnection of the relationship among the different participants.

III. Related Works

In the current literature different solutions have been implemented to face the above-mentioned challenges when considering just an individual energy vector. Following the graphical description in Figure 1, electricity must be considered as one of the MES System-of-System (SoS), allowing a detailed analysis of Smart Grid concepts. A solution to *Knowledge Integration* challenge has been already indentified for Smart Grid, so called *Smart Grid Architectural Model* (SGAM) [4]. SGAM is a tool to design and validate Smart Grid use cases in an architectural viewpoint. SGAM is a 3D structure with three main axis for the dimension of: i) *Domains* , ii) *Zones*, and iii) *Interoperability Layer*.

Domains axis represents the overall conversion chain of electricity following the *NIST Conceptual Model* [5] based upon IEC 62890 Value Stream Chain [6]. Domains are described as follow: *i)* ***Bulk Generation***, *ii)* ***Transmission***, *iii)* ***Distribution***, *iv)* ***Distributed Energy Resource (DER)***, and *v)* ***Customer Premise***. *Zones* represents the hierarchical levels of ICT control system implemented along the Domains to control the conversion chain of electricity based upon IEC 62264 [7] and IEC 61512 [8] Hierarchical Level for Automation. Zones are described as follow: *i)* ***Market***, *ii)* ***Enterprise***, *iii)* ***Operation***, *iv)* ***Station***, *v)* ***Field***, and *vi)* ***Process***. Zones reflect a functional separation of data aggregation. For instance, real-time measurements systems are typically in the Field and Station zones. Instead, functions that cover a large geographical area are usually located in Operation or Enterprise zone.

Finally, *Interoperability Layers* are relevant to the goal of integrating and interoperating different systems in the electrical conversion chain [9]. Interoperability Layers are found on Model-Based System Engineering (MBSE) and follow ISO 42010 International Standard for Architecture Description of System and Software [4]. Figure 3 defines particular *architecture views* based upon *architecture viewpoints*. A particular *viewpoint* governs a *view* and frames one or more *concerns*, each one having different *stakeholders*.

Interoperability Layers are architecture viewpoints to frame business, functional, informational, communication and physical concerns. Thus, they are divided in five main layers: *i)* ***Component Layer***: represents the physical distribution of all participating equipment in the Smart Grid context; *ii)* ***Communication Layer***: describes protocols and mechanism with the purpose of data signalling; *iii)* ***Information Layer***:

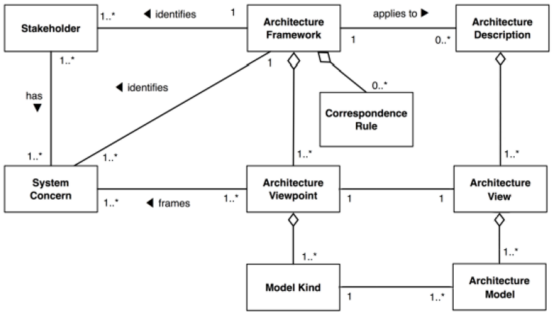

Fig. 3: Model-Based System Engineering (MBSE) conceptual model of an architecture description according to ISO 42010

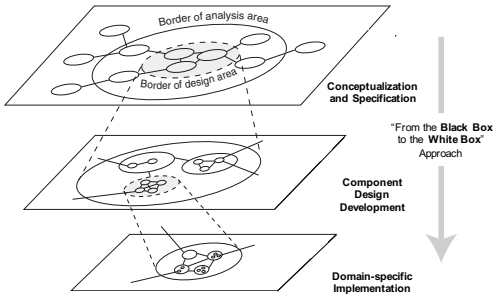

Fig. 4: *GAMES* hierarchical structure of the System-of-Systems.

describes information exchanged between functions, services and components in terms of data models and information models; *iv)* ***Functional Layer***: includes functions and services with their relationships and links independently from actors and physical implementations; *v)* ***Business Layer***: maps the regulatory policies, the market structure and the economic business models of the stakeholders involved and supports business executives in decision making related to new business models and cases.

Thus, SGAM identifies the right solution for an high-level architectural analysis of an electrict Smart Grid and offers an effective response when dealing with reference architecture in smart context [10]. SGAM adoptions have been proposed in literature to couple Smart Grid use case with others areas. For instance, Hurtado et al. [11], [12] couple a Building Energy Management Systems with the Smart Grid (SG-BEMS). In [13], [14], authors present E-Mobility system architectures to deal with human interaction with electric vehicles in Smart Grid use cases.

Several improvements have been attempted in order to extend these reference architectures to deal with a systemic description of dynamics involved in complex use cases. In such context, co-simulation has been proved as key technology that could provide a solution to this challenge. It allows the interconnection of different simulators enabling to exchange information between different software and hardware simulators. Moreover, co-simulation manages the simulation environments to initialise simulators and controls their step evolution.

To the best of our knowledge, a reference architecture for MES is not present in literature. None of the above-mentioned SGAM extensions deal with MES use case design. Moreover, the presented literature solution on co-simulation does not follow reference architecture models backing an high-level use case description. Thus, they do not assist use case designers in analysing dynamic complex system scenarios to deploy a reliable operational analysis of a MES. Thus, an integrated approach is needed to: *i)* propose a reference architecture model for MES use case description to enhance knowledge integration among MES designer community, *ii)* allow the integration of a systemic description of a MES use case into the above-mentioned reference architecture to deal with a grey-box description of the components involved, and *iii)* incorporate an automated process that allows to translate the systemic grey-box description into specific simulator DSLs and interconnect them following a co-simulation approach.

This paper proposes *GAMES*, a general-purpose architectural model for MES engineering applications. *GAMES* follows the innovative approach *"from the black box to the white box"*. It exploits and extends SGAM to deal with MES use cases following a black-box approach. In its core, it implements a UML/SysML coupling with the proposed architectural model to deal with a grey-box systemic description of the MES use case describing *i)* single components structure (i.e. input, attribute, parameter, and output), *ii)* each component specific simulation target (i.e. DSL or HIL specification), and *iii)* interconnection between components towards an operational description of the MES use case. Furthermore, *GAMES* will automatise the generation of each software component specific DSL code and HIL specification allowing a reliable and secure setup of a co-simulation framework to run MES use cases.

IV. *GAMES* Architectural Model

The *General-purpose Architectural model for MES engineering application* (*GAMES*) is an architectural modelling methodology which allows knowledge integration of different energy, ICT, financial, business and regulatory frameworks to perform Multi-Energy System modelling, use case definition and simulation extending SGAM to cope with MES. Moreover, it allows a systemic description of a MES use cases exploiting UML/SysML to describe cyber and physical components in depth and their interconnections. Finally, it will automatise the generation of each specific DSL code or HIL configuration related to a particular MES component. Furthermore, it will interconnect them using co-simulation techniques. Following the scheme in Figure 4, *GAMES* is structured in three main layers: *i) Conceptualisation and Specification* , *ii) Component Design Development*, and *iii) Domain-specific Implementation*. Each layer addresses one of the challenges defined in Section II.

A. Conceptualisation and Specification

Software Engineering offers tools to address architectural descriptions, use cases, scenarios and case studies. The aim

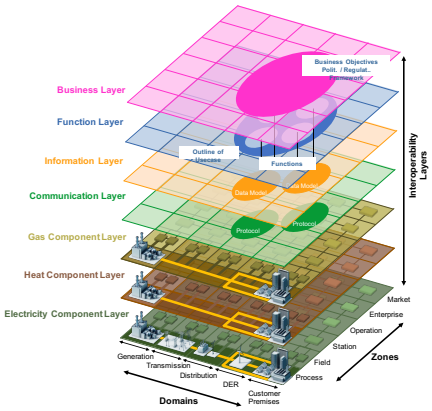

Fig. 5: General-purpose Architectural model for MES engineering application (GAMES).

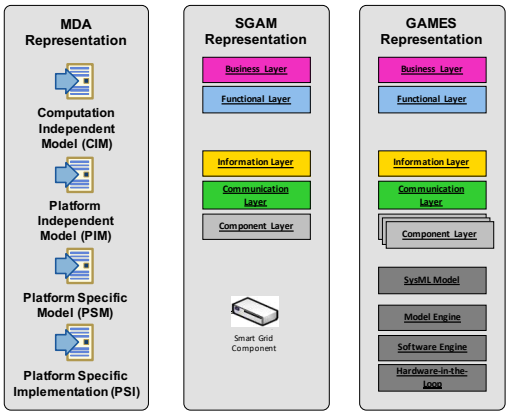

Fig. 6: SGAM and GAMES parallelism with Model Driven Architecture.

of such tools is transferring tacit knowledge when trying to document experience gained in a vertical field of technology (e.g. electrical and thermal engineering, distribution and transmission grid management and energy market analysis). Formal and explicit knowledge (as opposed to tacit knowledge) must be avoided to offer a clear high-level viewpoint of a use case.

Following the hierarchical structure of *GAMES* shown in Figure 5, *GAMES Conceptualisation and Specification* is based upon an extended version of the Smart Grid Architectural Model (SGAM) to cope with Multi-Energy System (MES) use cases description. One contribution of *GAMES* is to expand the SGAM Component Layer to three (or more) dimensions, each one for every energy vector. *GAMES* Conceptualisation and Specification extends SGAM considering that others energy vectors (e.g. heat exchanging fluids and gas) share a common structure with the electrical one. Following the MBSE structure in Figure 3, each energy vector identifies a different *architecture view* of physical layer *concern*. Moreover, the physical management of each energy vector supply chain involves different *stakeholders* from a structural and regulatory framework perspective addressing such *concern*. Consequentially, each energy vector requires a different *architectural viewpoint*, governing the *architectural view* of the physical interconnection between components.

It may be argued that the same division should apply for other layers (e.g. Communication and Information Layer). However, MES designers must focus their effort in developing communication, informational, functional and business solutions shared among all energy vectors involved in the use case. So, others layers are inherited by SGAM and extended to cope with all energy vectors interactions.

Unified Modelling Language (UML) is considered as *de facto* standard language in the field of Software Engineering, and it has been implemented into *GAMES*. The usage of UML in *GAMES* allows to share use cases universally among MES scientific community fostering reuse and extension of already developed use case. A MES designer could choose among different Integrated Development Environments (IDEs) supporting UML to scratch their MES use cases. Use Case formalization and MES conceptualization are processed into Business and Functional layers and they are suitably described by Use Case, Activity and Sequence UML diagrams. While MES specifications are handled into Information, Communication and Component layers and they are commonly described by Class, Object, Timing, Interaction and Communication UML diagrams.

B. Component Design Development

A conceptual mapping between SGAM Interoperability Layers and Model Driven Architecture (MDA) reveals a lack of SGAM approach to technological representation. MDA is a design approach for software system separating functionality and technology. The abstraction defined by MDA are:

- ***Computational Information Model (CIM)***: which is a systematic level describing functionality perspective;
- ***Platform Indipendent Model (PIM)***: which is an architectural level that decomposes the system in subsystem approaching it in a black-box fashion;
- ***Platform Specific Model (PSM)***: which involves the technological platform description of each component and the whole system, necessary for the actual implementation;
- ***Platform Specific Implementation (PSI)***: which is the implementation of the physical component, either hardware or software (i.e. source code).

A parallelism between MDA and SGAM representation is shown in Figure 6. Business and Functional Layers express the CIM level describing the functional perspective by the means of economical and service description. Furthermore, Information, Communication and Component Layers are represented by the PIM level concerning the SoS from ICT and physical equipment perspective. Finally, PSM and PSI are related to the Smart Grid constituent components considered as black-boxes. As shown in Figure 6, *GAMES* Conceptualisation and Specification behaves as SGAM. However, *GAMES "from the black box to the white box"* approach allows to expand Conceptualisation and Specification (i.e. black box approach) and zoom in analysis details providing *GAMES Component Design Development*.

Component Design Development enables a design framework to describe a MES use case from a systemic viewpoint covering the needs of the MDA PIM with in-depth analytical and logical description of the components. It exploits Systems Modeling Language (SysML), a general-purpose architecture modeling language for Systems Engineering applications based on UML. SysML is an enabling technology for Model-Based Systems Engineering (MBSE). It supports the specification, analysis, design, verification and validation of a broad range of systems. These systems may include hardware, software, information, and processes. SysML usage fosters the integration of functional aspects of MES components described in *GAMES* and their interfaces over all interoperability layers, taking advantage of the same IDEs to deploy the SysML diagrams. Each constituent component of *GAMES* is operationally, logically and analytically described by the Component Designe Development. Moreover, it describes the component interconnections and data exchange between entities bypassing the specific components coupling and interfacing. Conversely to SGAM, each component of *GAMES* is unpacked as a grey box model through SysML application. The component is described by so-called Four Pillars of SysML, referring to the four essential diagrams of SysML: Requirement, Activity, Block and Parametric diagrams.

C. Domain-specific Implementation

Formal knowledge is rather important when different vertical field of a MES are focused. In this context, formal knowledge is represented by the effort needed to design a MES component. Commonly, a MES component shows complex behaviour, endogenous and exogenous dynamics, and many interconnections with different energy vectors. When the scope of the analysis cover multiple vertical fields, the challenge becomes quite demanding.

GAMES Domain-specific Implementation will relieve MES designers from the formal knowledge required to exploit domain models related to the above-mentioned component technologies. The aim is to combine the UML/SysML diagrams with executable semantics to obtain an high level abstraction, supporting the translation of PIM into PSM (model to model transformation) and compilation of PSM into PSI (DSL code generation). It will specify the selected underlying technology (i.e. DSL and hardware configuration) in which each component will be deployed, tested and validated. The automation of the process identifies the simplest and smallest description of the block, avoiding complex component behaviour. This operation prevents the manual configuration of each component which can be error-prone. Moreover, a MES designer could access each generated DSL code and integrate complex functionalities of the components in the case of a particular simulation objective. Furthermore, Domain-specific Implementation simplifies the appropriate interconnections of such heterogenous software and hardware components allowing the interconnection to a co-simulation framework.

V. Conclusion

In this work, we presented *GAMES*, a general-purpose architectural model tools for MES engineering application. The architectural model deals with different challenges identified in the field of MES planning, development and testing. *GAMES* is divided into a hierarchical infrastructure that follows *"from the black-box to the white-box"* approach. Firstly, it integrates an MBSE architectural model for MES use case description extending SGAM. Then, it allows the systemic description of each model permitting a grey-box description of each component involved in the MES use case through SysML. Finally, *GAMES* will allow to translate the UML/SysML descriptors into PSM and compiled into DSL code to simulate software components and connect specific hardware into the simulation loop.

References

[1] P. Mancarella, "MES (multi-energy systems): An overview of concepts and evaluation models," *Energy*, vol. 65, p. 1–17, 02 2014.

[2] R. Haberfellner, O. de Weck, E. Fricke, and S. Vössner, *Systems Engineering: Fundamentals and Applications*, 2019.

[3] A. M. Ross, D. H. Rhodes, and D. E. Hastings, "Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value," *Systems Engineering*, vol. 11, no. 3, pp. 246–262, 2008. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sys.20098

[4] J. Bruinenberg, L. Colton, E. Darmois, J. Dorn, J. Doyle, O. Elloumi, H. Englert, R. Forbes, J. Heiles, P. Hermans *et al.*, "CEN CENELEC ETSI Smart Grid Coordination Group on Smart Grid Reference Architecture," *CEN CENELEC ETSI Technical Report*, pp. 98–107, 2012.

[5] D. Wollman, C. Greer, D. Prochaska, P. Boynton, F. Mazer, C. Nguyen, G. Fitzpatrick, T. Nelson, G. Koepke, A. Hefner, V. Pillitteri, T. Brewer, N. Golmie, D. Su, A. Eustis, D. Holmberg, and S. Bushby, "NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0," 2014.

[6] IEC Central Secretary, "Life-cycle management for systems and products used in industrial-process measurement, control and automation," International Electrotechnical Commission, Standard IEC TR 26890, 2013.

[7] IEC Central Secretary, "Enterprise-control system integration," International Electrotechnical Commission, Standard IEC TR 62264, 2013.

[8] IEC Central Secretary, "Batch control," International Electrotechnical Commission, Standard IEC TR 61512, 1997.

[9] S. Widergren, D. Hardin, R. Ambrosio, R. Drummond, E. Gunther, G. Gilchrist, and D. Cohen, "GridWise Interoperability Context-Setting Framework," 2008.

[10] M. Uslar, S. Rohjans, C. Neureiter, F. Andrén, J. Velasquez, C. Steinbrink, V. Efthymiou, G. Migliavacca, S. Horsmanheimo, H. Brunner, and T. Strasser, "Applying the smart grid architecture model for designing and validating system-of-systems in the power and energy domain: A european perspective," *Energies*, vol. 12, p. 258, 2019.

[11] E. Mocanu, K. O. Aduda, P. H. Nguyen, G. Boxem, W. Zeiler, M. Gibescu, and W. L. Kling, "Optimizing the energy exchange between the smart grid and building systems," in *In Proc. of: 49th UPEC*, 2014, pp. 1–6.

[12] L. Hurtado, P. Nguyen, and W. Kling, "Smart grid and smart building inter-operation using agent-based particle swarm optimization," *Sustainable Energy, Grids and Networks*, vol. 2, pp. 32 – 40, 2015.

[13] G. Schuh, J. Fluhr, M. Birkmeier, and M. Sund, "Information system architecture for the interaction of electric vehicles with the power grid," in *In Proc. of: 10th ICNSC*, 2013, pp. 821–825.

[14] B. Kirpes, P. Danner, R. Basmadjian, H. De Meer, and C. Becker, "E-mobility systems architecture: a model-based framework for managing complexity and interoperability," *Energy Informatics*, vol. 2, no. 1, p. 15, 2019.