



**HAL**  
open science

## Supporting the Engineering of Multi-Fidelity Simulation Units With Simulation Goals

João Cambeiro, Julien Deantoni, Vasco Amaral

► **To cite this version:**

João Cambeiro, Julien Deantoni, Vasco Amaral. Supporting the Engineering of Multi-Fidelity Simulation Units With Simulation Goals. International Workshop on Multi-Paradigm Modeling for Cyber-Physical Systems (MPM4CPS'21), Oct 2021, Fukuoka, Japan. <hal-03374990>

**HAL Id: hal-03374990**

**<https://inria.hal.science/hal-03374990v1>**

Submitted on 12 Oct 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Supporting the Engineering of Multi-Fidelity Simulation Units With Simulation Goals

João Cambeiro

Université Côte d'Azur

NOVA School of Sciences and Technology

joao.cambeiro@etu.univ-cotedazur.fr

Julien Deantoni

Université Côte d'Azur

Sophia antipolis, France

julien.deantoni@univ-cotedazur.fr

Vasco Amaral

NOVA School of Sciences and Technology

Lisbon, Portugal

vma@fct.unl.pt

**Abstract**—To conceive a CPS is a complex and multidisciplinary endeavour involving different stakeholders, potentially using a plethora of different languages to describe their views of the system at different levels of abstraction. Model-Driven Engineering comes, precisely, as a methodological approach to tackle the complexity of systems development with models as first-class citizens in the development process. The measure of realism of these models with respect to the real (sub)system is called fidelity. Usually, different models with different fidelity are then developed during the development process. Additionally, it is very common that the development process of CPS includes an incremental (and collaborative) use of simulations to study the behaviour emerging from the heterogeneous models of the system. Currently, the different models, with different fidelity, are managed in an ad hoc manner. Consequently, when a (Co-)simulation is used to study a specific property of the system, the choice of the different models and their setup is made manually in a non-tractable way. In this paper we propose a structured new vision to CPS development, where the notion of *simulation goal* and *multi-fidelity simulation unit* are first-class citizens. The goal is to make a clear link between the system requirements, the system properties, the simulation goal and the multi-fidelity simulation unit. The outcome of this framework is a way to automatically determine the model at an adequate fidelity level suitable for answering a specific simulation goal.

**Index Terms**—Model fidelity, Model, Simulation, Model-Based System Engineering

## I. INTRODUCTION

A Cyber-Physical System (CPS) [1] is an integration of computation with physical processes whose behavior is defined by both physical and cyber parts. To understand the emerging behavior of the system, one cannot comprehend the computational and physical parts separately. It is then required that several stakeholders, some of which are specialists in various domains involved in the system, participate and collaborate in the system development process.

The Model-Based System Engineering (MBSE) [2] approach is an appropriate option to use during the system lifecycle to tackle the challenge of designing and maintaining a complex system that deals with multiple domains. MBSE is a methodological approach (supported by methods and tools) to tackle the complexity of systems development systematically. This entails promoting the use of models throughout the system development process, during which several versions of the system models are produced, and usually, it is possible to observe an increase in the fidelity of the models over time.

The fidelity of a model of a system is determined by how accurately it can reproduce the properties of the system under study [1]. It is also considered as a measure of realism [3] of a model or simulation. The fidelity level of a model can be a qualitative description such as "high", "medium", or "low" assigned by an expert on the domain of the model or a quantitative description if a higher precision of the semantics of the value is required.

At each step of this iterative development process, it is essential not to lose track of the desired system properties that initially set the system requirements.

Simulation and Co-Simulation [4], [5] are methods that enable the study of the emerging behavior of the system by executing the developed models in isolation or in a collaborative way. It is part of the Verification & Validation (V&V) [6] process. Intermediate simulations at each iteration step of the development procedure are used to validate a subset of the system requirements. When there is a will to validate only a subset of the requirements, it means that there is a specific *simulation goal*, which targets specific requirements. Additionally, [7], [8] show a trade-off between fidelity and the time and computational resources necessary to perform the simulation. As the fidelity of models increases, the time and computational resources required to perform the simulation also increases.

Currently, the relationships between the system requirements and the simulation goals are usually left implicit or managed in an ad-hoc manner. This forbids reasoning on what models, at which fidelity level, are appropriate to fulfil the simulation goal. Some low fidelity models may not be appropriate because they do not define the properties required by the requirements of interest or because their definition is not accurate enough. Some high fidelity models may not be appropriate since their computation time is too long.

Modelling traceability exists [9], [10] and could be used as a partial solution, however it usually does not consider simulation setup/results and how the simulation goals are defined.

Multi-fidelity models are already known and used [7], [8] to achieve more efficient (Co-)simulations with respect to the simulation time and the required computational resources. However, these approaches do not consider the notion of simulation goal as a rationale of their choices, letting them

implicit and non-tractable.

In the following sections, we present our early work towards formalizing a framework that defines simulation goals and multi-fidelity simulation units as first-class citizens and, considering a simulation goal, determines automatically a set of system models suitable to be used to set up the (co)simulation. In section II, we start by refining the notion of multi-fidelity using a running example. In section III, we present an overview of our framework. In section IV, we describe work that relates to our approach. Finally, in section V we conclude our paper discussing possible research roadmaps.

## II. ILLUSTRATION OF MULTI-FIDELITY WITH A RUNNING EXAMPLE

In this section, we present several examples of how multi-fidelity is present in the Smartlab case study [11]. The Smartlab is a Building Automation System (BAS) CPS [12] deployed in a computer science lab. Since humans participate in the CPS's feedback loop, this system is considered a particular type of CPS, the Human-in-the-loop CPS (HiTLCPs) [13]. In Figure 1 we present a Product Breakdown Structure (PBS) [14] of the Smartlab. The system components are primarily divided, judging if they are members of the system's computational or plant parts. In the case study, humans are not mere participants in the system plant. They are considered components of the system, and they also assume the roles of sensors by assessing the temperature and light conditions available in the room and the role of actuators by performing tasks requested by the system controller. The IoT platform plays a central role in managing the 92 sensors that measure the internal /external light levels, temperature, room occupancy and energy consumption. It also manages 49 actuators comprised of lights, the air conditioning unit (AC), and the power sockets.

We elected, as the property of interest for the examples in this section, the system energy consumption. The Smartlab is an actually deployed system, and from its operation, it is possible to collect real data. The observed system energy consumption behavior is the reference that will be used to determine the fidelity of the produced models during the system development. It is the accuracy of the models to reproduce the existing system energy consumption that will define their fidelity level.

Figure 2 presents the measured energy consumption and its variation during one day at the Smartlab. The energy consumption is divided into two categories: the AC unit's energy consumption which is responsible for most of the energy used during the case study, and the rest of the equipment operating in the laboratory like the light management system and the energy consumption measured by the power sockets. Considering the breakdown structure presented in Figure 1, the sum of the energy consumption of the AC unit, IoT platform hardware and the measured energy consumption at the power sockets determine the total energy consumption. One cannot overlook that each one of these terms can be influenced by system properties belonging to the following domains:

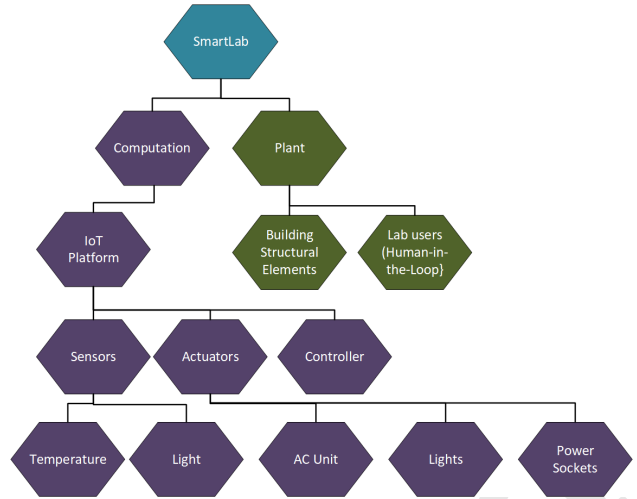


Fig. 1. Smartlab PBS

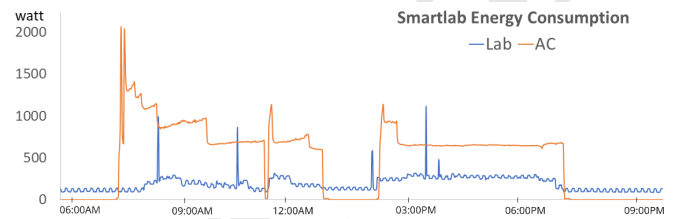


Fig. 2. Energy consumption during one day at the Smartlab. Observed data from the deployed system

- IoT Infrastructure - The control rules representing the desired environmental conditions that influence the actuation on the CPS's plant.
- The HiTL: The behaviour of humans and their working conditions requirements.
- Weather conditions – Outdoors temperature and the natural light available influence the control, which maintains the appropriate working environmental conditions.

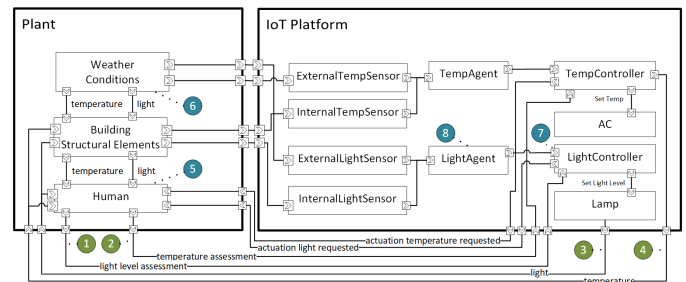


Fig. 3. A high-level model of the Smartlab

The fidelity of a model does not increase simply by injecting more details into it [5]. A higher fidelity is achieved by adding details that better characterize the set of properties of interest.

Figure 3 presents an example of a high-level timed model that focuses on relevant aspects of the system, the management of thermal and light conditions available in the laboratory to

determine our property of interest. This model considers the HiTL participation in the system. Humans play the sensor role by providing qualitative assessments of the temperature and light conditions available in the laboratory (connectors in Figure 3 marked as 1 and 2). They also play the actuator role by performing tasks requested by the system to change the temperature and light conditions like opening/closing a window or opening/closing a blind, respectively (connectors marked as 3 and 4).

In the early stages of the development process, the focus might be on getting the basic functionality working. Due to the complexity required to model the HiTL, the models related to the HiTL might not be available yet, so this component is not considered by the temperature and light controllers. The lumped model can be obtained from the presented model by eliminating the connectors between the human block and the temperature and light controllers blocks. Although the model is suitable to check if the required basic functionality is satisfied, the fidelity of our property of interest, the system energy consumption is lower given that the contribution of the HiTL is not faithfully reproduced. It is then possible that the various models developed during the system lifecycle present different fidelity levels for the system properties.

Choosing different levels of participation of the HiTL in the system models, leads to developing several versions of the Human block (marked as 5). The various versions can be differentiated by their functional interface and the set of observable properties at the interface level. If we consider that in one version of the Human model, the role of sensor is not supported, then the visible light assessment property is not part of the Human block set of observable properties, and at the functional interface, the connectors marked as 2 and 4 do not exist on this version of the model. When integrating models at the different fidelity levels, if the interconnected models are not compatible at the interface level (ex. a required input by a model is not produced by the connected model), modifications on one or both the models are required to make the interfaces of both models compatible. An example of this could be the use of a version of the light controller (marked as 7) that has as one of its inputs, the feedback provided by the HiTL that will be integrated with a version of the Human model's where the set of observable properties does not include the assessment of the visible light property.

To assess the fidelity of a model to reproduce an observable property, it is also necessary to determine how close its representation at the interface level is from its representation on the actual system. The main operational goals of a light level agent (marked as 8) are the collection of the measured data from the available sensors, the pre-processing of the collected data and the delivery of the data to the light controller. In the deployed system, the light agent collects data from 5 indoor light sensors, and at the interface level, exposes each of the collected light level values. Again, an early version of the light agent developed to check some basic functionality requires at the interface level an abstracted representation of the visible light level in the room depicting the average visible

light level in the room. The changed representation of the property at the interface level can lead to incompatibilities when communicating with the light controller, as the output representation should match the expected input representation.

### III. PROPOSED FRAMEWORK OVERVIEW

MBSE is a system engineering methodology that uses models as primary artefacts throughout the system lifecycle. In the requirements elicitation phase, the system stakeholders define the system requirements from the desired system properties. System properties are defined as differentiating factors that can be measured or computed, while a system requirement is a valuation of a system property that can be satisfied or not.

The system development usually includes the decomposition of the system into sub-systems, by following approaches like the Product Breakdown Structure approach as depicted in Figure 4 where the system is decomposed in sub-systems.

Simulation is a popular method to V&V the behaviour of a system with respect to the requirements. A simulation unit (SU) is a virtual version of a system's (sub-)system to be used in a simulation context. Its objective is to mimic behaviourally its system counterpart. Since we are dealing with multi-fidelity, multiple SUs (with different fidelity levels) related to a single sub-system exist. A stakeholder can use simulations to validate one requirement of interest for him. For this, he has to identify the system properties, which are constrained by this requirement and observe if the evolution of the value of these properties during the simulation conforms to the requirement. In our approach, we propose to define a simulation goal explicitly. A simulation goal determines the requirements that must be verified during the simulation. Additionally, a simulation goal defines some extra-functional properties regarding the simulation. The simulation extra-functional properties are quality attributes of the simulation, like accuracy or the time necessary to perform the simulation.

Since different versions of the SU with different fidelity levels may exist, the stakeholder must decide which SU he should use according to the simulation goal. As the fidelity of models increases, the time and computational resources required to perform the simulation also increases [7], [8].

We are currently working on a framework that guides the framework user in selecting the sets of models, possibly at different fidelity levels that can be used in a simulation to answer a simulation goal. We propose a new type of SU, the multi-fidelity SU, to approach multi-fidelity systematically. Its structure is presented in Figure 4. This unit encapsulates all SUs that are related to the same sub-system. The same observable property can have different fidelity values in different SUs, depicted in Figure 4 as the property's fidelity measure.

At the interface level, the multi-fidelity unit exposes a set of observable properties. An observable property of a meta-fidelity SU is a system property for which at least one of the meta-fidelity SU encapsulated SUs can be used to observe its behaviour during a simulation. There are no observable properties at the interface level of a multi-fidelity SU that are not part of the system properties set required to define

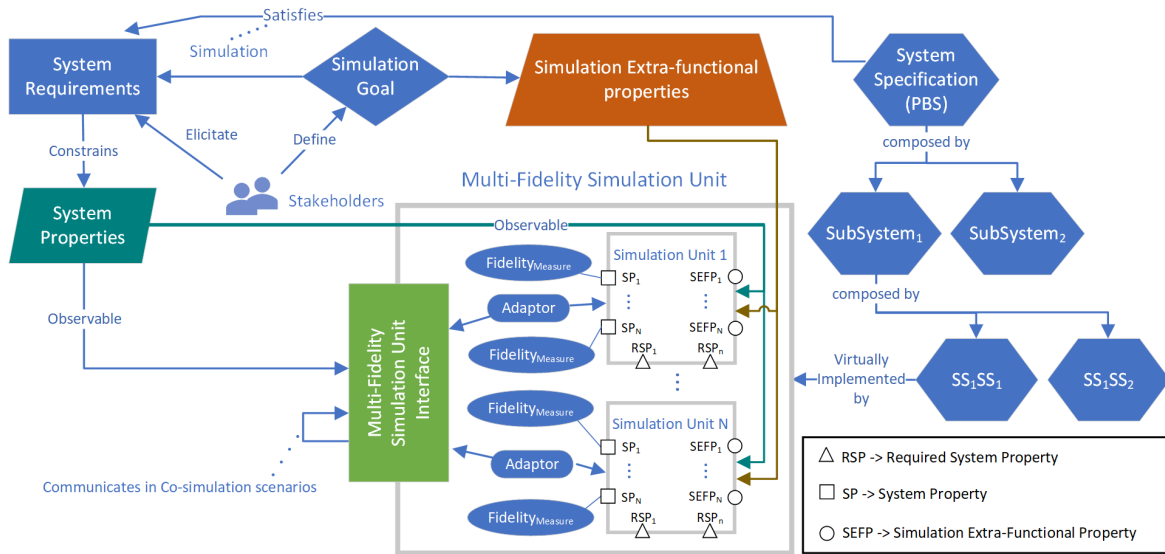


Fig. 4. Proposed Multi-fidelity Framework Overview

the system requirements. In Figure 4 depicts that the system requirements constrain the set of system properties candidates and that the set of observable properties of a multi-fidelity SU is a subset of the system properties. Also represented at the interface level are the SUs simulation extra-functional properties. The valuation of these properties in a SU may result from previous utilizations of the SU in (co-)simulations. These properties can be a factor in a stakeholder decision of which SU to use for answering a simulation goal. To facilitate the identification of which SUs of the multi-simulation unit can be used to answer a simulation goal, expressed as a combination of system properties and simulation properties, a feature diagram [15] representing the supported combinations of observable properties and simulation properties is available. This allows exposing at the interface level all the functionality supported by the encapsulated SUs. Considering the Smartlab example presented in section 2, a HiTL multi-fidelity SU would include all the various versions of the HiTL SUs developed. It would expose all the properties that result from the participation of humans in the various roles supported at the interface level.

In [5] the authors define an experimental frame as a specification of the conditions that the system can be experiment with. It is also proposed a definition for the scope frame of a model as the most relaxed conditions that a model is suitable to be used. A model is deemed suitable to be used on a particular experimental frame if the experimental frame can be derived from the scope frame of the model. We intend to adapt these concepts to a multi-fidelity context, where the notion of experimental frame is linked to the notion of simulation goal and the scope of a model is linked to the set of observable properties that is possible to study from the emerging behaviour of a set of SUs in a (co)simulation. Then in the framework context, a set of SUs is suitable to answer a simulation goal if the frame defined by the system properties of interest and the simulation

extra-functional properties expressed in the simulation goal can be derived from the scope frame (bounded by set of observable properties of the SUs).

If we consider a requirement, R1, expressed as "The participation of humans over the period of one month in tasks to evaluate the temperature and light conditions in the laboratory assigned by the system results in energy consumption savings higher than 10%". A simulation goal could be the verification of R1 in less than 5 minutes. The frame defined by the simulation goal is bounded by the time simulation property that requires a timed SU to check if the property holds, the system properties of interest necessary to answer the simulation goal, the energy consumption, the human as a temperature sensor, the human as a light sensor, and light and temperature controllers that require the assessment of humans, of the light and temperature conditions. The scope frame of a set of SUs obtained from the timed Smartlab model presented in section II includes all the properties that bound the simulation goal frame, and it also incorporates the system properties of humans as temperature and light conditions actuators. It is then possible by applying a simple projection operation on the scope frame of the set of SUs to derive the frame defined by the simulation goal. The set of SUs is then appropriated to verify R1 in a co-simulation. Finally, to set up a co-simulation, it is necessary to address the challenge presented in section II where a system property may have different representations at the interface level of different SUs. When connecting two SUs, each belonging to a different multi-fidelity unit, it is the task of the modeller responsible for a particular SU that to make the interface of the SU compatible with the enclosing multi-fidelity SU interface. We propose the use of adaptors as, e.g. used in SystemC [16]. In Figure 5, we present a possible use of an adaptor that makes it possible to overcome the challenge presented in section II, where the representation of a SU observable property does not match the

representation of a property that belongs to the input set of a target SU. In the example, the adaptor converts a multi-valued representation of the light levels measured in the room, sent by the light agent SU to an averaged value compliant with the representation format that the light controller SU requires.

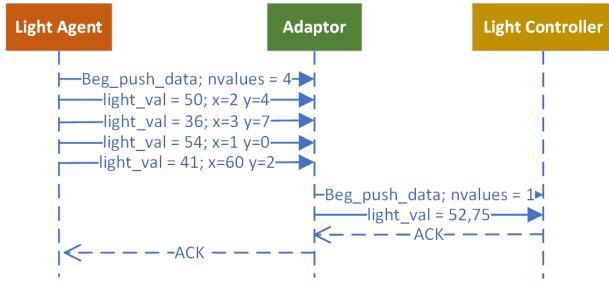


Fig. 5. An adapter is used to compatibilize the SUs interfaces

At the current stage of our work, we should note that providing an answer on whether a combination of models validates a requirement is not a framework's goal. Neither is the ordering of the selected sets of models based on a particular criterion.

#### IV. RELATED WORK

Several research works using multi-fidelity modelling have been reported in the literature [7], [8], [17]–[19]. Typically these works focus on how to select a model with the desired accuracy that is less computational and time demanding to simulate [20]. In [7], [8], [17], multiple models with different fidelity levels are used during a simulation for simulation speed up. A region of interest is defined where the use of a high-fidelity model is required to keep the simulation error at a low level. A low-fidelity model is used in the other regions since its lower accuracy does not prevent that the simulation is performed with an acceptable accuracy. In [21] an adaptive abstraction approach is used to achieve the simulation speed up. During a simulation, switching between abstractions of a model occurs based on a validity frame that tests if the more simple version of the model can reproduce a system property with an acceptable accuracy. When the more abstracted model cannot be used in a predictable way, the simulator switches to the less abstracted (more detailed) model. In our opinion, this work opens up the possibility of research to determine if a link between multi-abstraction and multi-fidelity exists. The existing work provides a valuable source of information on how multi-fidelity is present in system engineering in multiple domains. We seek to extend current works by proposing a framework where the simulation properties are not determined a priori.

#### V. CONCLUSION

We proposed a methodological framework that makes explicit the link between the system requirements, the system properties, the simulation goal and the multi-fidelity SU. Using these links it is possible to reason on which level of fidelity can be used during a (Co-)simulation for a specific simulation goal. While the framework is still a work in progress, we

foresee relevant research questions to be solved as a result of it. Namely, if there is a relation between fidelity and abstraction and if known abstraction techniques can be applied in a multi-fidelity context.

#### REFERENCES

- [1] E. A. Lee and S. A. Seshia, *Introduction to embedded systems: A cyber-physical systems approach*. Mit Press, 2017.
- [2] A. W. Wymore, *Model-based systems engineering*. CRC press, 2018, vol. 3.
- [3] D. C. Gross *et al.*, "Report from the fidelity implementation study group," in *Fall Simulation Interoperability Workshop Papers*, 1999.
- [4] C. Gomes, C. Thule, D. Broman, P. G. Larsen, and H. Vangheluwe, "Co-simulation: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 3, pp. 1–33, 2018.
- [5] B. P. Zeigler, A. Muzy, and E. Kofman, *Theory of modeling and simulation: discrete event & iterative system computational foundations*. Academic press, 2018.
- [6] M. Roza, J. Voogd, and D. Sebalj, "The generic methodology for verification and validation to support acceptance of models, simulations and data," *The Journal of Defense Modeling and Simulation*, vol. 10, no. 4, pp. 347–365, 2013.
- [7] S. H. Choi, S. J. Lee, and T. G. Kim, "Multi-fidelity modeling & simulation methodology for simulation speed up," in *Proceedings of the 2nd ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, 2014, pp. 139–150.
- [8] A. H. Bayoumy and M. Kokkolaras, "A reference error formulation for multi-fidelity design optimization," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 58134. American Society of Mechanical Engineers, 2017, p. V02BT03A038.
- [9] I. May, "Systems and software engineering—architecture description," Technical report, ISO/IEC/IEEE 42010, 2011.(Cited on page 12.), Tech. Rep., 2011.
- [10] E. Villar, J. Merino, H. Posadas, R. Henia, and L. Rioux, "Mega-modeling of complex, distributed, heterogeneous cps systems," *Microprocessors and Microsystems*, vol. 78, p. 103244, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0141933120304051>
- [11] J. Cambeiro, C. Gomes, V. Amaral, A. Rodrigues, and J. Cunha, "A building automation case study setup and challenges," in *2018 IEEE/ACM 4th International Workshop on Software Engineering for Smart Cyber-Physical Systems (SEsCPS)*. IEEE, 2018, pp. 41–44.
- [12] S. Wang, G. Zhang, B. Shen, and X. Xie, "An integrated scheme for cyber-physical building energy management system," *Procedia Engineering*, vol. 15, pp. 3616–3620, 2011.
- [13] S. Munir, J. A. Stankovic, C.-J. M. Liang, and S. Lin, "Cyber physical system challenges for human-in-the-loop control," in *8th International Workshop on Feedback Computing (Feedback Computing 13)*, 2013.
- [14] M. Servello and M. W. Evans, "Work breakdown structure," *Encyclopedia of Software Engineering*, 2002.
- [15] T. Käkölä and J. C. Duenas, *Software product lines*. Springer, 2006.
- [16] T. Grötter, S. Liao, G. Martin, and S. Swan, *System Design with SystemCTM*. Springer Science & Business Media, 2007.
- [17] A. H. Bayoumy and M. Kokkolaras, "A relative adequacy framework for multimodel management in multidisciplinary design optimization," *Structural and Multidisciplinary Optimization*, vol. 62, no. 4, pp. 1701–1720, 2020.
- [18] Y. Kuya, K. Takeda, X. Zhang, and A. I. Forrester, "Multifidelity surrogate modeling of experimental and computational aerodynamic data sets," *AIAA journal*, vol. 49, no. 2, pp. 289–298, 2011.
- [19] K. Geiselhart, L. Ozoroski, J. Fenbert, E. Shields, and W. Li, "Integration of multifidelity multidisciplinary computer codes for design and analysis of supersonic aircraft," in *49th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2011, p. 465.
- [20] M. Giselle Fernández-Godino, C. Park, N. H. Kim, and R. T. Haftka, "Issues in deciding whether to use multifidelity surrogates," *AIAA Journal*, vol. 57, no. 5, pp. 2039–2054, 2019.
- [21] R. Franceschini, S. V. Mierlo, and H. Vangheluwe, "Towards adaptive abstraction in agent based simulation," in *2019 Winter Simulation Conference (WSC)*, 2019, pp. 2725–2736.