# *chapter one*

# *Introduction to the modeling and simulation-based systems engineering handbook*

***Daniele Gianni, Andrea D'Ambrogio, and Andreas Tolk***



*Contents*



Systems engineering—"… an interdisciplinary collaborative approach to derive, evolve, and verify a life-cycle balanced system solution which satisfies customer expectations and meets public acceptability"(IEEE 1998, p. 11)—has been taught and practiced for several decades, and a body of knowledge (http://www.sebokwiki.org/) has been developed to share and centralize experiences and successful practices. However, despite the fact that they are applying systems engineering processes, many projects still overrun their budgets, fail to deliver in time, and sometimes even do not deliver something really useful or do not even meet the requirements. Many team members continued to work in *silos of domain excellence*, and the systems engineering processes did not connect the team members as intended. New emerging ideas were gradually introduced under the efforts of model-based systems engineering (MBSE) (Estefan 2008; Wymore 1993), bringing a significant improvement with respect to the aforementioned shortcomings. With MBSE, a common model becomes a centerpiece of the engineering process. This common model is used to communicate with various stakeholders, users in different phases of the life cycle, and in particular team members of the systems engineering team in charge of defining, developing, operating, maintaining, upgrading, and finally retiring the system. The model is rooted in system requirements and is modified step-by-step through analysis and real-world constraints to become a blueprint of the real system.

Introducing models to capture the efforts of the various systems engineering processes instead of pure documents is mathematically and organizationally a significant step forward. Using the findings of isomorphism, the equivalency of models can be evaluated precisely, replacing educated guesses of subject-matter experts with precise model-transformation rules and formal evaluations. In the software domain, the model-driven architecture (MDA) approach of the Object Management Group was a milestone when introduced in 2003. The core idea of MDA is to base software engineering on a series of model transformation, from platform-independent models via platform-dependent models to code and executables. The interested reader is referred to Kleppe et al. (2003) for more details. In the domain of modeling and simulation (M&S), the idea to use model transformation as the backbone of simulation development was recently featured in detail by Cetinkaya (2013). The book of Mittal and Martin (2013) is another example that the idea starts to not only be accepted, but also has a significant influence for practitioners in the field, as also witnessed by the various editions of the *International Workshop on Model-Driven Simulation Engineering*, started in 2010 by D'Ambrogio and Petriu (2011) within the SCS SpringSim multi-conference. The theoretic foundations for such efforts can be traced back to the mathematical branch of model theory, as shown by Tolk et al. (2013). Models, in particular formal models and their transformations, are claiming a central role in systems engineering. Theoretically and practically, we made some significant progress.



Looking at these success stories of MBSE and our own experience with this topic, a question arose spontaneously: is it possible to add more value when using M&S methods and computational capabilities to numerically evaluate the properties of complex systems? It is well known that the earlier mistakes are uncovered in system designs, the cheaper it is to correct them. Introducing M&S-based systems engineering, as an extension of MBSE practices to also include simulation methods, offers the opportunity for a more comprehensive coverage of the design space by enabling systems engineers to identify a wider set of early mistakes. Currently, M&S already has its place in testing, in particular in the defense domain (Zeigler et al. 2005), but could we move it even closer to the system design? Is it possible to start thinking about M&S-based systems engineering?

## *Toward M&S-based systems engineering*

The aim of this handbook is to start laying the foundations for such an effort, which may lead to a textbook in the future, when many of the currently open research questions will be answered. To structure the chapter contributions—and also hopefully a research agenda for a future discipline of M&S-based systems engineering—we utilized the taxonomy already utilized by MBSE, as starting point for possible extensions too. Of the identified topics, the handbook covers the following thematic areas:

- *Methods to support system modeling*: How can the process of modeling a system be supported, in particular when M&S shall be applied in this process, or the use of M&S methods shall be prepared to optimize these processes? This topic is addressed in Chapters 2–4.
- *Improving system architecture*: System architectures are well known in the systems engineering community. Frameworks such as the Zachman framework (Inmon et al. 1997), The Open Group Architecture Framework (Harrison 2007), the United States Department of Defense Architecture Framework and related efforts in NATO and other nations (U.S. DoD 2014), and related systems engineering efforts are well established, but can they support the idea of using M&S and systems engineering better? This topic is addressed in Chapter 2.
- *Domain-specific languages*: General languages such as the Unified Modeling Language (UML) or the System Modeling Language (SysML) are well established, but are there new developments that support M&S-based systems engineering better? Will ontologies and metamodeling play a larger role here? This topic is addressed in Chapters 6 and 10.
- *Model-driven simulation engineering*: Systems engineering principles already are applied often to drive simulation solutions. How can the latest research insights drive the ideas forward to use models and simulation to drive the engineering of systems? This topic is addressed in Chapters 6 and 7.
- *Collaborative environments*: A common and well-orchestrated set of tools and methods supporting M&S experts as well as systems engineering is the ultimate vision. However, until we reach this vision, we need better support of integrative collaborative environments that bring current methods and tools of experts of all domains together. This topic is addressed in Chapter 7.
- *Simulation algorithms and performance engineering*: To support systems engineering effectively, simulation not only needs to be reliable and informative, but also needs to be performing. High-performance computing, performance engineering, and better simulation algorithms will support engineers with thousands of simulation runs to cover the decision space with numerical evaluations. This topic is addressed in Chapters 8 and 9.
- *Simulation software architecture*: When simulation systems are used for decision support for systems engineering, new requirements for the simulation software architecture will emerge. Part of these will be derived from the need for the reuse and integration of systems engineering tools; others will be derived from performance

challenges. The integration into the operational environment and the fully calibratable accessibility of measures of metrics for performance and efficiency are part of them. This topic is addressed in Chapters 10 and 11.

- *Processes:* Western philosophy of science has been heavily influenced by the idea that substantials are the main carriers of knowledge. Objects and their attributes, and their relations to other objects dominate the world of knowledge representation. Processes play a subordinate role as they are merely seen as the things that create, change, or destroy objects. Several alternative approaches are using objects and processes as equally important concepts and have successfully been applied to systems engineering (Dori 1995). We assume that such balanced approaches will be advantageous for supporting system engineers, their stakeholders, and other team members. This topic is addressed in Chapters 12 and 13.
- *Verification and validation (V&V)*: Systems engineering as well as M&S use the terms V&V, but the interpretation thereof is slightly different. They agree, however, that verification ensures that the system is built correctly, and that validation ensures that the correct system is built. Nonetheless, for systems engineers, the requirements and their fulfillment build a trusted foundation for these V&V processes. If they can show that a system fulfills the requirements, they are good. Simulation, on the other hand, is the result of modeling, which is the task-oriented purposeful abstraction and simplification of a perception of reality. Ensuring that simulations are correct and useful for the system or representing the system appropriately is an important task with many open research points. This topic is addressed in Chapters 14 and 15.
- *Enterprise architecture*: The era of net-centric systems introduced a new set of challenges. Systems are no longer providing their functionality exclusively in the context of the original requirements, but they can be composed into federations or system of systems in which the individual functionalities of systems are combined to satisfy new customer needs. Systems have to be resilient and adaptive, and enterprise architectures provide one of the means to support this. The topic is closely related to systems architectures, but adds a new level of complexity to it.
- *Model repositories*: We assumed that models can be reused before, so it makes sense to provide the reusable models in common repositories. Although this topic can be seen as part of the collaborative environment, we decided to give it its own bullet. The challenges on how to describe a model that can be identified as a candidate, the selection of a good set of candidates based on problem descriptions, composing the models to provide a solution, and many more

challenges belong to this topic. This topic is addressed in Chapters 7 and 16.

- *Advanced concepts*: There are likely many more emerging categories that have to be captured in the body of knowledge that describes what to do to provide better M&S support for systems engineering, and we captured contributions that are not easily mappable to the earlier bullets here. Chapter 17 falls under this topic.

In summary, we tried to identify a way forward to use the potentials of M&S for better systems engineering in the domains of architecture alignment and improvement, collaboration tools and repositories, and theoretical foundations. It may be too early to coin the term M&S-based systems engineering as the necessary concepts are still in the state of infancy. However, by bringing the various experts together to contribute to a common book to become the foundation of future research and collaboration, we hope to have contributed to make this happen in the foreseeable future.



## Chapter contributions

The book consists of 16 chapters (Chapters 2 through 17) that deal with the aforementioned topics.

In Chapter 2 (“Systems engineering, architecture, and simulation”), A. Tolk and T. K. Hughes propose the alignment of processes, methods, and means that are already used by both systems engineers and simulation practitioners. Aligning the processes and using system architecture artifacts as a common repository result in a consistent model that can be executed as a simulation, providing additional numerical insight and eventually resulting in high-quality, trustworthy, and cost-effective systems as required by the customer.

In Chapter 3 (“System modeling: Principled operationalization of social systems using Presage2”), S. Macbeth, D. Busquets, and J. Pitt first provide various reasons for modeling and simulating social systems (e.g., formalize human methods of problem solving as a source and inspiration for developing computational systems to address engineering challenges), and then describe the implementation of the multiagent-based animation and simulation platform, Presage2, which supports the process of principled operationalization, defined as the mapping between formal models of social systems and implemented systems.

In Chapter 4 (“Formal agent-based models of social systems”), I.-C. Moon introduces two approaches for formal agent-based modeling and simulation of social systems. The chapter first shows the meta-network model that is useful in describing the structure of social systems. Next, the discrete event systems (DEVS) formalism is applied to illustrate the behavior of social systems.

In Chapter 5 ("On the evolution toward computer-aided simulation"), L. F. Perrone addresses the problem of failures in simulation studies that arise from the complexity of the simulation workflow, which introduces several points of failure for those with less expertise in the field. He then proposes the introduction of *computer-aided simulation* systems, which include comprehensive support tools that guide users throughout most if not all the stages of the modeling and simulation endeavor, thus leading to highly credible results without overburdening the user.

In Chapter 6 ("Model-driven method to enable simulation-based analysis of complex systems"), P. Bocciarelli and A. D'Ambrogio address the use of distributed simulation techniques to model the inherently distributed architecture of complex systems. The chapter proposes a method that exploits principles, standards, and tools introduced in the model-driven engineering field and supports the automated generation of high-level-architecture-based distributed simulations from system models specified by the use of SysML, the UML-based general purpose modeling language for systems engineering.

In Chapter 7 ("Collaborative modeling and simulation in spacecraft design"), V. Schaus, D. Lüdtke, P. M. Fischer, and A. Gerndt deal with the collaborative and concurrent design of spacecrafts with respect to modeling and simulation. Looking at the systems engineering life cycle, they discuss expectations and advantages of model-driven design approaches, and highlight the potential of formalized design for continuous verification and validation. The chapter closes with a vision on promising ideas to integrate modeling and simulation as a fast and easy-to-use means of supporting the spacecraft design process.

In Chapter 8 ("Performance engineering of distributed simulation programs"), G. Iazeolla and A. Pieroni apply software performance engineering techniques to predict, at design time, the effect of program partitioning and network capabilities on the efficiency of distributed simulation systems, to establish whether the use of distributed simulation can lead to a speedup with respect to the use of conventional sequential simulation, and thus decide the convenience to proceed with the distributed simulation implementation.

In Chapter 9 ("Reshuffling PDES platforms for multi/many-core machines: A perspective with focus on load sharing"), F. Quaglia, A. Pellegrini, and R. Vitali address the use of parallel discrete event simulation (PDES) and discuss how rethinking the organization of PDES platforms to make them perfectly suited for exploiting the computing power offered by modern multi/many-core machines. The proposed approach is based on an innovative load-sharing paradigm suited for PDES systems run on top of multicore machines.

In Chapter 10 ("Layered architectural approach for distributed simulation systems: The SimArch case"), D. Gianni introduces a layered architecture named SimArch, for local and distributed simulation systems, based on the IEEE High Level Architecture standard. The chapter also outlines a mechanical procedure that informally proves that no extra know-how or effort is necessary to develop a distributed simulation system with regard to the equivalent local simulation one.

In Chapter 11 ("Reuse-centric simulation software architectures"), O. Dalle addresses the importance of reusability in simulation software, which is also motivated by business-specific reasons, such as providing a better reproducibility of simulation experiments, or avoiding a complex validation process, other than by more conventional reasons such as improved reliability and decreased development costs. The chapter considers reuse as a problem that may be considered in two opposite directions: reusing and being reused.

In Chapter 12 ("Conceptual models become alive with Vivid OPM: How can animated visualization render abstract ideas concrete?"), D. Dori, S. Bolshchikov, and N. Wengrowicz address the processes needed to define and coordinate activities in M&S-based systems engineering. The chapter describes OPM, an emerging ISO Standard 19450 for modeling complex systems using a compact set of concepts, and Vivid OPM, a software module that enables the modeler to create a moving cartoon, that is, a video clip that is driven by the OPM underlying conceptual model.



In Chapter 13 ("Processes to support the quality of M&S artifacts"), J. Himmelspach and S. Rybacki focus on well-defined processes as one of the preconditions for high-quality products. Based on an overview of work done before, they describe how M&S software can be created utilizing workflow management systems, methods, and technologies. This approach is leading to well-defined processes in the software and to well-defined experiments, by keeping all degrees of freedom for users.

In Chapter 14 ("Formal validation methods in model-based spacecraft systems engineering"), J.-P. Katoen, V. Y. Nguyen, and T. Noll introduce a novel modeling language and toolset for a (semi) automated validation approach in the context of spacecraft systems, in which validation methods are labor intensive, usually being based on manual analysis, review, and inspection. The proposed language enables engineers to express the spacecraft system design, the software design, and their reliability aspects in a single integrated model. The model can then be analyzed by the COMPASS toolset for checking requirements related to functional correctness, safety and dependability, and performance.

In Chapter 15 ("Modeling and simulation framework for systems engineering"), S. Diallo, A. Tolk, R. Gore, and J. Padilla introduce a modeling and simulation system development framework (MS-SDF) that can be effectively applied to successfully model, simulate, verify, and validate a model in support of systems engineering. Within the MS-SDF, the chapter helps the reader determine whether a model satisfies design requirements and adequately represents a real system without violating key assumptions.

In Chapter 16 ("Liquid business process model collections"), W. M. P van der Aalst, M. La Rosa, A. H. M. ter Hofstede, and M. T. Wynn address the problems due to the increasing amounts of data (*big data*) that many organizations need to deal with intelligently to turn data into valuable insights that can be used to improve business processes. The chapter proposes to manage large collections of process models and event data in an integrated manner. The collection should self-adapt to evolve organizational behavior and incorporate relevant execution data (e.g., process performance and resource utilization) extracted from the logs, thereby allowing insightful reports to be produced from factual organizational data.

Finally, in Chapter 17 ("Web-based Simulation using Cell-DEVS Modeling and GIS Visualization"), S. Wang and G. Wainer focus on web-based simulation as a means to integrate geographic information systems (GIS), modeling, simulation, and visualization, to support the simulation-based design process in system engineering. The chapter proposes a method to extract information from GIS, model with Cell-DEVS theory, run web-based simulation, and visualize results back in Google Earth. The web-based simulation is based on RESTful Interoperability Simulation Environment (RISE), which supports different simulation environments for both remote simulation on a single server and distributed simulation on multiple servers.



## Conclusion

In his essays on life itself, the famous scientist and researcher Robert Rosen, who studied, in addition to biology, mathematics and the history of science, states (Rosen 1998):

> I have been, and remain, entirely committed to the idea that modeling is the essence of science and the habitat of all epistemology.

For M&S researchers, this is a powerful statement, as it makes the argument that models are the essential way to capture knowledge, and we can preserve this knowledge in the form of model repositories. Furthermore, we can make this knowledge applicable to all in the form of simulations.

Tolk et al. (2011) make the case that in order to capture intelligence for systems engineering support, three components are needed: (1) ontologies to provide the unambiguous structure for terms describing the concepts of the application domains as well as of the implementation domains;

(2) simulations to capture the knowledge in dynamic and executable form; and (3) software agents that support the process of identification of applicable models, selection of the optimal set of models, composition of these models into a usable solution, execution of these composition of models, and evaluation of the results in the light of the customers task. Again, simulation plays a pivotal role in capturing knowledge.

The contributions to this book show that we are making progress in all identified research domains, in architectures, collaboration, repositories, and theoretical foundations. M&S experts become aware of systems engineering processes that can support them and that they can support. Systems engineers recognize the potential of M&S to produce numerical insight into the behavior of complex systems—including emerging behavior that can be positive or negative in regard to the intended use of the system—in all life cycles of a system. MBSE opened the door for more collaboration, and the use of dynamic simulation, in addition to the static models, is a logical next step. We honestly hope that many young researchers will be motivated to contribute to the increasingly growing and overlapping domains of M&S and systems engineering so that an update of this handbook with many new insights will have to be published soon.



## *References*


Cetinkaya, D. (2013). Model Driven Development of Simulation Models, Defining and Transforming Conceptual Models into Simulation Models by Using Metamodels and Model Transformations. PhD Thesis at the Technische Universiteit Delft, the Netherlands.

D'Ambrogio, A. and Petriu, D., *First International Workshop on Model-driven Approaches for Simulation Engineering (Mod4Sim'11)*, http://www.sel.uniroma2.it/mod4sim11, last accessed February 2014.

Dori, D. (1995). Object-process analysis: Maintaining the balance between system structure and behaviour. *Journal of Logic and Computation*, 5(2), 227–249.

Estefan, J. A. (2008). *Survey of Model-Based Systems Engineering (MBSE) Methodologies*. International Council on Systems Engineering, San Diego, CA.

Harrison, R. (2007). *TOGAF Version 8.1*. Van Haren Publishing, Zaltbommel, the Netherlands.

IEEE. (1998). *IEEE Standard 1220: Application and Management of the Systems Engineering Process.* IEEE Press, New York, NY.

Inmon, W. H., Zachman, J. A., and Geiger, J. G. (1997). *Data Stores, Data Warehousing and the Zachman Framework: Managing Enterprise Knowledge*. McGraw-Hill, Chicago, IL.

Kleppe, A. G., Warmer, J. B., and Bast, W. (2003). *MDA Explained: The Model Driven Architecture: Practice and Promise*. Addison-Wesley Professional, Boston, MA.

Mittal, S. and Martín, J. L. R. (2013). *Netcentric System of Systems Engineering with DEVS Unified Process*. CRC Press, Boca Raton, FL.

Rosen, R. 1998. *Essays on Life Itself.* Columbia University Press, New York.

Tolk, A., Adams, K. M., and Keating, C. B. (2011). Towards intelligence-based systems engineering and system of systems engineering. *Intelligence-Based Systems Engineering*. Edited by A. Tolk and L. C. Jain. Springer Verlag, Berlin/Heidelberg, Germany. ISRL Vol. 10, 1–22.

Tolk, A., Diallo, S. Y., Padilla, J. J., and Herencia-Zapana, H. (2013). Reference modelling in support of M&S—foundations and applications. *Journal of Simulation*, 7(2), 69–82.

U.S. Department of Defense (DoD) Deputy Chief Information Officer. (2014). *Web portal DoD Architecture Framework*, http://dodcio.defense.gov/dodaf20 .aspx, last accessed February 2014.

Wymore, A. W. (1993). *Model-Based Systems Engineering*. Series on Systems Engineering, Vol. 3., CRC Press, Boca Raton, FL.

Zeigler, B. P., Fulton, D., Hammonds, P., and Nutaro, J. (2005). Framework for M&S-based system development and testing in Net-centric environment. *ITEA Journal of Test and Evaluation*, 26(3), 21–34.