

A Shift from Model-Based to Simulation-Based Paradigm: Timeliness and Usefulness for Many Disciplines

Tuncer Ören^{1*}, Saurabh Mittal² and Umut Durak³

¹School of Electrical Eng. and Computer Science, University of Ottawa, Ottawa, Ontario, Canada

²Modeling, Simulation, Experimentation and Analysis, MITRE Corporation, McLean, VA, USA

³Institute of Flight Systems, German Aerospace Center (DLR), Braunschweig, Germany

Abstract

Model-based paradigm has been adopted by a number of disciplines since its introduction in the late 70s. After a brief history on how model-based approach started in simulation, the merit and the spread of the model-based approach are described. The concept of model is subsumed in simulation, but many times employing just the model-based approach does not involve simulation. The superiority and advantages of simulation-based approach over model-based approach are manifold. This article describes the differences and advocates the use of simulation-based paradigm in advancing any computational discipline.

Publication History:

Received: December 16, 2017

Accepted: January 23, 2018

Published: January 25, 2018

Keywords:

Model-based approach, Model-based disciplines, Model-based simulation, Model-based systems engineering, Model-based software engineering, Simulation-based disciplines, Simulation-based engineering, Simulation-based software systems engineering

Introduction

In the early days of simulation, namely in the 1970s, computers were not very powerful. Before the 20th century, most calculations were done by humans and the term computer which dates back 1640s was used to denote humans who calculate [1]. Later the term is used to denote machines that perform knowledge processing. Similarly, in the 1970s, software engineering was at its infancy, as even the term software engineering was only coined in a NATO conference held in 1968 [2].

Accordingly, in early days of simulation, the aim of computerization was to generate model behavior under specified experimental conditions and simulation was considered by some as a technique of last resort. Even simulation languages had flaws in their definitions. For example, one of the early and widely used simulation language was CSSL (Continuous System Simulation Language) [3]. Syntactic errors of the original definition of CSSL were documented in a systematic way in a report [4]. The following quote from the Stanford Encyclopedia of Philosophy reflects the established understanding about the importance of models in science:

Models are of central importance in many scientific contexts. The centrality of models such as the billiard ball model of a gas, the Bohr model of the atom, the MIT bag model of the nucleon, the Gaussian-chain model of a polymer, the Lorenz model of the atmosphere, the Lotka-Volterra model of predator-prey interaction, the double helix model of DNA, agent-based and evolutionary models in the social sciences, and general equilibrium models of markets in their respective domains are cases in point. Scientists spend a great deal of time building, testing, comparing and revising models, and much journal space is dedicated to introducing, applying and interpreting these valuable tools. In short, models are one of the principal instruments of modern science [5].

Models are essentially representations or purposeful abstractions of reality. Traditionally we build models of systems to understand their structures and behaviors through simulation. Simulation provides us

the possibility to perform goal-directed experimentation or gaining experience about the systems of interest. With the advent of model-based era we started to talk about models that we develop to build or to understand the systems. Model-based software or systems engineering is an example of that. Simulation-based paradigm promotes simulating the models both during the initial design phase and to analyze natural phenomena, and to test best control strategies for existing systems. The increasing significance of simulation-based approaches as a natural successor of model-based approaches is currently observable in many disciplines. The recent edited volume of Mittal, Durak and [6] provides a comprehensive review of simulation-based approaches applied to various disciplines ranging from engineering to education.

The article is structured to share the testimony to the paradigm shift from model-based to simulation-based as follows. Section 2 provides an overview on the historical aspects of Modeling and Simulation (M&S) and various disciplines that employed M&S. Section 3 provides an overview of various model-based approaches. Section 4 elaborates on the shift from the model-based to simulation-based paradigm. Section 5 describes the importance of simulation-based paradigm in next generation multi-disciplinary systems engineering. Finally, Section 6 concludes the article.

Early Studies of Model-Based Simulation

In an article by Ören and Zeigler [7], a simulation program was dissected for the first time, and lines of simulation program code were identified as representing model and different elements of

Corresponding Author: Prof. Tuncer Ören, School of Electrical Eng. and Computer Science, University of Ottawa, Ottawa, Ontario, Canada; E-mail: oren.tuncer@sympatico.ca

Citation: Ören T, Mittal S, Durak U (2018) A Shift from Model-Based to Simulation-Based Paradigm: Timeliness and Usefulness for Many Disciplines. Int J Comput Softw Eng 3: 126. doi: <https://doi.org/10.15344/2456-4451/2018/126>

Copyright: © 2018 Ören et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the experimental conditions. In an article in 1982, computer-aided modeling systems were promoted where the role of the computer was promoted from mere generation of model behavior to assisting users to specify simulation model [8]. The term model-based was –for the first time– part of the title of a book which resulted from a NATO-sponsored [9]. The first chapter of the book laid down the basis for model-based approach [10]. Two Post publications were supportive of two aspects of simulation:

The founder of The Society for Modeling & Simulation International (SCS), John McLeod [11] wrote:

Dr. Ören was the first speaker. After an appropriate greeting he presented a lecture on Computerization of Model-Based Activities: A Paradigm Shift and New Vistas. Although he covered the fundamental elements of a simulation study; models and behaviour; elements of a model-based simulation software system; synergies of simulation, software engineering, artificial intelligence, and general system theories; knowledge-based modeling and simulation systems; and highlights of desirable research in simulation, Tuncer's main thrust, it seemed to me, was to place simulation in a central position for several scientific disciplines by a shift of paradigm.

Samson in his review wrote [12]:

For both simulation researchers and practitioners, this book may be worth buying just for its two chapters (by Ören and Elzas). They provide a unifying view of simulation, artificial intelligence, system theories, software engineering and modelling. ... *For those in our profession who still consider simulation to be merely a set of techniques for use when all else fails* (emphasis added) [13], this book contains an interesting and quite powerful counterinterview.

Model-based systems engineering

The first book on Model-Based Systems Engineering was written by A.W. Wymore [14]. Use of the term model-based in Wymore's pioneering systems engineering book was not accidental. Ören's dissertation was done, at the University of Arizona, at World's first Systems Engineering Department (later Systems and Industrial Engineering Department), under its founder Wymore and was a development of a simulation specification language–instead of a simulation programming language (GEST – General System Theory implementor) [15,16]. It was based on Wymore's book A Mathematical Theory of Systems Engineering: The Elements [17]. GEST was developed for the specification of continuous-change simulation models described by ordinary differential equations. However, acceptance of a systems-theory-based simulation specification language was not automatic: Ören's first article that was sent late 1970 to SCS was rejected. On the 50th anniversary of the foundation of SCS, Ören wrote the following:

While in Tucson, I sent an article on GEST to Simulation, the journal of the SCS. The reviewers rejected the article. I started to correspond with the editor. Finally, John McLeod intervened and informed me that the article would not be published in Simulation. I was not expecting at that time to have a reviewer open to a categorically different paradigm. Since then however, John –founder of the SCS– has become one of my best friends [18].

Theory of Modeling and Simulation (M&S) by Bernie Zeigler was among the first to present a theoretical framework linking the

discrete event paradigm, the model and the underlying computational framework, a simulator, to execute the model dynamically. This was well supported by the General Systems Theory. Later, Bernie and Tuncer wrote an article titled: System Theoretic Foundations of Modeling and Simulation: A Historic Perspective and the Legacy of A. Wayne Wymore [20] that summarized the historical undertakings.

Other Model-Based Approaches

The evolution of software engineering to model-based software engineering [21] coincides with the model-based approach already successfully applied to simulation and systems engineering. The Unified Modeling Language (UML) that was also developed in the 1990s [22] opened new horizons in engineering of software intensive systems.

The motivation of model-based approaches in software engineering is to improve productivity with the generation of software artefacts, including source code via transformation and stepwise refinement of models [23].

There are many types model-based approaches in software engineering. Brambilla and his colleagues introduce a comprehensive categorization of the model-based approaches used by software engineering community [24]:

Model-Driven Development (MDD) proposes a paradigm that utilises models as the primary artefacts and redefines the implementation as (semi) automatic generation from the models. The process relies on the use of models, and the systematic production and transformation of models.

Model-Driven Architecture (MDA) is the particular vision of MDD proposed by the Object Management Group (OMG).

Model-Driven Engineering (MDE) expands MDD to cover all engineering process areas with a focus on developing metamodels to facilitate automated transformations.

Model-Based Engineering (MBE) utilises model-driven practices pragmatically, not necessarily in an integrated fashion, in various steps of the engineering process. While models are important in MBE, they do not necessarily drive the development process. In this sense, all model-driven processes are regarded as model-based.

Model Integrated Computing (MIC) refines MDD approaches and provides an open integration framework to support formal analysis tools, verification techniques and model transformations in the development process [25].

Model-driven Systems Engineering (MDSE) defined by Mittal and Martin [26] for System of Systems utilizes the benefits of both MBSE and MDE

Other model-based disciplines

Model-based simulation, model-based systems engineering, and model-based software engineering were successfully followed by other fields such as model-based science [27], model-based engineering [28], and model-based testing [29].

As a result, MB/MD approaches place models at the core of an entire system/software development process and try to increase the system specification quality. While MBE and MBSE

originated in late 70s and have been applied to entire systems engineering life cycle, MDA, MDD and MIC are mostly applied to software intensive systems. For a detailed comparative analysis, see Mittal and Martin [26].

Shift from Model-Based to Simulation-Based Paradigm

Simulation has been maturing for a long time [30, 31] and became an essential infrastructure for many disciplines by allowing model-based experimentation and model-based experience gaining using dynamic models, namely by models where behavior and/or structure change over time. Several books already elaborated on the benefits of simulation-based approaches. For example: Simulation-Based Engineering Science (SBES): Revolutionizing Engineering Science through Simulation [28]. Some quotations from NSF report [28] follow:

Seldom have so many independent studies been in such agreement: simulation is a key element for achieving progress in engineering and science.

SBES constitutes a new paradigm that will be indispensable in meeting the scientific and engineering challenges of the twenty-first century.

A program in SBES could also lead to new approaches to medical practice, collectively called Simulation-Based Medicine.

SBES has the potential to deliver, within a short design period, designs that are optimized for cost performance and total impact on the environment.

The first comprehensive book on simulation-based disciplines is published in 2017 [6]. The first chapter by Ören, Mittal, and Durak [32] of the book, clarifies the benefits of simulation-based approach for many disciplines as well as synergies of agent-directed simulation, with computerization, and systems thinking. Many of the disciplines have already accepted model-based approaches and methodologies to support the design of new systems, and analyze the structure and behavior of current systems. However, in some cases, model-based approaches have not yet find their way to simulation-based approaches due to either lack of need of simulation or project time-constraints, for the simple reason that turning a model-based approach to a simulation-based approach is a non-trivial effort—albeit very profitable for the success of the project. The advances in model transformation technologies in the last decade are a step in that direction where abstract models can be turned into concrete models that can be transformed into executable code, and eventually participate in a simulation experiment in the final stage.

Modeling and Simulation are two distinct activities, both in practice and engineering. Modeling is user-facing, that is, the user, a modeler, is “facing” the reality and observing (or contemplating) a real-world phenomenon. The act of modeling is to represent it either in his mind, on paper or in a computational environment (using graphical editors or modeling software), to gain a common understanding with other observers and modelers for a shared understanding of the reality. Modeling is the process of developing and using abstractions to simplify the real-world for its usefulness. The discipline of developing models through various methodologies and computational infrastructure is incorporated in Model Engineering. This defines how modeling editors, workbenches, languages and representations are developed to aid a domain or abstractions in general. Domain specific modeling (DSM) is a derivation of Model Engineering applied to a

domain, or a discipline for that matter. Simulation, on the other hand, subsumes modeling activity and is focused towards experimentation on the model and providing an experience of model-interaction to the modeler. The discipline of developing a simulator, a device that executes the model on a time-base, is incorporated in Simulation Engineering, with the resulting artifact called as Simulator and the professional dealing with simulation engineering, a Simulationist. The simulator can be an algorithm that is executed on a paper, or a software program, that is executed in a computational environment either on a single machine, or in a parallel distributed environment or in a High Performance Computing (HPC) environment or in a federated environment with other simulation systems across large geographical distances, as in Live, Virtual and Constructive (LVC) environments. Within the Experimentation aspect, simulation is used for:

1. behavior prediction and performance analysis;
2. analysis of alternatives;
3. sensitivity analysis;
4. engineering design;
5. virtual prototyping;
6. planning;
7. acquisition; and
8. proof of concept.

Within the Experience aspect, simulation (sometimes as virtual reality [VR] and augmented reality [AR]) is used for:

1. training to enhance one of the three types of skills, i.e., motor skills (virtual simulation), decision making skills (constructive simulation), and operational skills (live simulation);
2. entertainment (simulation games); or
3. shared knowledge and emotions, as in art [33] and literature. For the last category, visual renderings are theatre, movie, and TV.

In most of the design problems, the real system does not exist. In some cases, even if it exists, it is not accessible for experimentation such as in space exploration problems. The dynamics and the response of the system also play an important role in determining what task could be supported. Both too slow dynamics, such as economic studies, and too fast dynamics, such as particle physics, make the observations almost impossible. The safety considerations may also, sometimes, dictate a simulation-based experimentation. Conducting test for aircrafts out of their flight envelopes is an example of that. In other cases, experimentation may not be acceptable by populace, such as experimentation with education systems. The cost of experimentation is also another driver for simulation use. Conclusively, it is more convenient to use simulation when experimentation conditions cannot be fulfilled physically, for example, the range of the system variables cannot be controlled or are unachievable. These characteristics also persuade the use of simulation.

Why Simulation-Based Paradigm is Crucial for Computerization and Software Systems Engineering?

We live in the computer age which is also called digital age (referring to the fact that the pertinent computers are digital), machine age (which implicitly refers to computers), and information age. Information

existed even long before computers. At the Alexandria library in Egypt, for example, The earliest surviving figure, from the 3rd century BCE, is reported as more than 200,000 books. [34]. However, access to and processing of information are possible due to the widespread availability of computers.

The importance of computers in our era is so well established that, underlining it is even a truism. And it is equally well established that software engineering is crucial for the success of any non-trivial computer applications. However, still one cannot not claim that the advancement of software engineering matched the advancement of computers. In the NSF Report [28] the following was posited:

Finding: *Much of our current software in computational engineering science is inadequate for dealing with the multifaceted applications and challenges of SBES. New software tools, paradigms, and protocols will need to be developed so that software is more transferable between fields and not wastefully duplicated. In the multidisciplinary teams we establish for SBES research, we must incorporate experienced software developers who will work closely with engineering scientists to develop tomorrow's SBES software.*

In an earlier article Ören pointed out another aspect of software requirement:

So far as users are concerned, the aim of computerization is not necessarily to develop software but to solve problems with the assistance of computers. Therefore, the software industry would serve the users better by providing Computer-Aided Problem Solving (CAPS) environments. Teaching software packages as well as advanced Computer-Aided Software Engineering (CASE) tools and environments can be useful as part of short-term vocational training. However, what is important is the realization of the necessary shifts of paradigms in software engineering in order to establish computer-aided problem solving as a reliable, easy, efficient, productive, and profitable process [35].

Paradigms like Model Based System Engineering (MBSE), Model Based Development (MBD), and Model Based Engineering (MBE) are often used to describe the usage of modeling practices in systems and software engineering. Many times, modeling is performed without the simulation. In that role, modeling acts as a validation mechanism to bring a common understanding to all the stakeholders that include the end-users (who use the system), the engineers (who make the system) and the business holders (who invest in the system). These validated models serve as architectural blueprint. Such models are usually static in nature and the verification or correctness of these models rests with the modeling workbench used to create these models. The metamodels implemented in the modeling editors are the foundation on which the modeling representation rests. To evaluate the dynamic behavior of these models, simulation is warranted. While we have mechanisms to deal with software and systems engineering for such M&S aspects, the problems take wicked shape when it comes to multi-disciplinary engineering, or in the extreme cases of complex adaptive systems (CAS) engineering [36] or Cyber CAS engineering employing M&S [37]. An example of CyCAS is a Smart City that incorporates a complex adaptive System of System involving intelligent transportation system, autonomous cars, adaptive power and utility systems, and other community services and systems interacting with humans and other systems. How can we even begin to fathom a model of a smart city, and the subsequent study and design of it using simulation-based methodologies?

A Cyber CAS is a CAS manifesting in a netcentric environment and has eight characteristics [38]:

1. Human-in-the-system
2. Multi-agent system
3. Efficient Control and communication in a netcentric environment
4. Resource-constrained and scalable
5. Displays emergent attention based on second-order cybernetics
6. Defined phase transitions
7. Structured knowledge
8. Resilient or Anti-fragile.

In order to create a model of CAS or CyCAS, modeling paradigms from domains like Cognitive Science, Complexity Science, Network Science, Cybernetics, Operations Research and Distributed Artificial Intelligence, to name a few, are required to be brought together for a multi-paradigm multi-resolution model [37]. Underlying theories in these domains, for example, Network theory, Communications theory, Estimation theory, Systems theory, along with their engineering practices need to be brought together as a multi-disciplinary team for a cohesive M&S endeavor. Undoubtedly, validation (of the model) and verification (of the simulator) takes on a whole new meaning.

Without a fail safe validation process, the model can fall out-of-sync quickly rendering the model useless. Much of the model validation depends on the data validation. Data validation depends on the data exchanged between the actual system and its environment, in comparison to the data exchanged between the model and the environment. Acquisition of structured data from natural systems and man-made systems that can capitalize on various data-driven analyses is a growing challenge. Data-driven methodologies need to scale up in a multi-disciplinary environment as well. If the modeling workbenches can support agile modeling processes, the simulation infrastructure needed to validate the dynamical behavior should be made agile as well.

Currently, the validation and verification approaches to undertake a CAS or CyCAS M&S endeavor are limited in scope. The community is encouraged to develop such approaches to exploit the simulation-based engineering of CyCAS for the benefit of society towards a richer technological future. Simulation-based experimentation and experience are very much needed to drive the investments for Smart City undertakings.

Similar to maturing of simulation engineering to simulation systems engineering, considering software systems engineering can be very beneficial [39, 40].

Conclusions

Simulation establishes a model in a computational environment and allows us to experiment with the model to solidify our understanding in a dynamic environment. "Computation" in computational M&S is an aspect of modeling and simulation and does not (cannot) imply that modeling and simulation is a subfield of computation or software engineering; similar to the case of computational astronomy which uses computation but is distinct from computation. Simulation extends the power of computation by allowing experimentation and experience. Furthermore, as a superiority to databases, simulation models can act as generators of new data under the associated scenarios.

Simulation, being possible with any dynamic model in any field, is a universal and powerful endeavor. Simulation-based thinking is prevalent in many disciplines such as complex systems, engineering, medicine, natural sciences, physical sciences, social sciences, education and training. A recent book by Mittal, Durak and Ören [6] provides a compendium of the state-of-the-art in these disciplines that are now supported with simulation technologies. Dedicated professionals, in leadership, in management and in operation, practice Modeling and Simulation as a profession and as integral infrastructure in their day-to-day operations, as highlighted in another recent effort by Tolk and Ören [41]. Simulation is a big business with new technologies such as serious gaming, augmented reality and virtual worlds that are now household entities. Without simulation at their core, such experiences would never have become possible. Simulationists, over the course of their careers, work together with many subject-matter-experts and provide simulation-based problem-solving capabilities to several domains.

Today, we cannot (should not) embark any complex system study (design, analysis, or control) without considering simulation-based techniques! Simulation-based approaches in any discipline - like the proverbial sharpening the axe-is a rational way to enhance engineering performance in an effective way. By emphasizing the role of simulation in education in many disciplines, future professionals in these disciplines may be better equipped for their professions. Involving -even non-computational- simulation in K - 12 educations, future generations may be better prepared to have enhanced thinking abilities.

The technology-rich future of tomorrow warrants the use of simulation-based approaches in the very definition of it.

Competing Interests

The authors declare that no competing interests exist.

References

- (OED-Computer): Online Etymology Dictionary
- NATO (1968) Software Engineering: Report of a conference sponsored by the NATO Science Committee. Scientific Affairs Division, NATO
- SCi CSSL (1967) The Sci Continuous System Simulation Language. *Simulation* 9: 281-303
- Ören TI (1975) Syntactic Errors of the Original Formal Definition of CSSL 1967 Technical Report TR75-01. IEEE Computer Society Repository, Computer Science Department University of Ottawa Ottawa, Ontario, Canada.
- Frigg R, Hartmann S (2017). Models in Science. The Stanford Encyclopedia of Philosophy, Edward N. Zalta (ed.)
- Mittal S, Durak U, Ören T (2017) Guide to Simulation-Based Disciplines: Advancing our Computational Future. Springer.
- Ören TI, Zeigler BP (1979) Concepts for Advanced Simulation Methodologies. *Simulation* 32: 69-82.
- Ören TI (1982) Computer-Aided Modelling Systems. Academic Press, London, England 189-203.
- Ören TI, Zeigler BP, Elzas MS (1984) Simulation and Model-Based Methodologies: An Integrative View. NATO ASI Series, Springer-Verlag, Berlin, Heidelberg, New York
- Ören TI (1984a) Model-Based Activities: A Paradigm Shift. *Simulation and Model-Based Methodologies: An Integrative View* 3-40.
- McLeod J, McLeod S (1982) IF I KNEW THEN ..., Simulation in the Service of Society IX-X.
- Samson D (1985) Review of Simulation and Model-Based Methodologies: An Integrative View. *European Journal of Operations Research* 21: 411-412.
- Wagner HM (1975) Principles of Operations Research: With Applications to Managerial Decisions. Prentice-Hall.
- Wymore AW (1993) Model-Based Systems Engineering. CRC Press, Boca Raton.
- Ören TI (1971) GEST: General System Theory Implementor. A Combined Digital Simulation Language. University of Arizona, Tucson, AZ.
- Ören TI (1984b) GEST - A Modelling and Simulation Language Based on System Theoretic Concepts. *Simulation and Model-Based Methodologies: An Integrative View* 281-335.
- Wymore AW (1967) A Mathematical Theory of Systems Engineering: The Elements. Krieger, Huntington, NY.
- Ören TI (2002) SCS and Simulation: Fifty Years of Progress. 50th Anniversary Issue, *Modeling and Simulation* 1: 32-33.
- Bernie Zeigler (1976)
- Ören TI, Zeigler BP (2012) System Theoretic Foundations of Modeling and Simulation: A Historic Perspective and the Legacy of A Wayne Wymore. Special Issue of *Simulation* 88: 1033- 1046.
- Bruno, G (1995) Model-Based Software Engineering. Chapman & Hall.
- Jacobson I, Booch G, Rumbaugh J (1999) The Unified Software Development Process. The Complete Guide to. Addison-Wesley.
- Atkinson C, Kuhne T (2003) Model-driven development: a metamodeling foundation. *IEEE Software*, 20: 36-41.
- Brambilla M, Cabot J, Wimmer M (2012) Model-Driven Software Engineering in Practice. Morgan & Claypool Publishers.
- ISIS (1997) Model Integrated Computing (MIC).
- Mittal S, Martin JLR (2013) Model Driven Systems Engineering for Netcentric System of Systems with DEVS Unified Process. Winter Simulation Conference.
- Oden JT, Belytschko T, Fish J, Hughes TJR, Johnson C, et al. (2006) A Report of the National Science Foundation Blue Ribbon Panel on Simulation-Based Engineering Science: Revolutionizing Engineering Science through Simulation.
- Kramer A, Legeard B (2016) Model-Based Testing Essentials: Guide to the ISTQB Certified Model-Based Tester: Foundation Level. Wiley
- Ören TI (2005) Maturing Phase of the Modeling and Simulation Discipline. International Academic Publishers - World Publishing Corporation 72-85.
- Ören TI (2007) The Importance of a Comprehensive and Integrative View of Modeling and Simulation. Proceedings of the Summer Simulation Conference.
- Mittal S, Durak U, Ören T (2017) Guide to Simulation-Based Disciplines: Advancing our Computational Future. Springer.
- Ören TI, Mittal S, Durak U (2017) The Evolution of Simulation and its Contributions to Many Disciplines. *Guide to Simulation-Based Disciplines* 3-24.
- Dewey J (1934) Art as Experience. Penguin Group, New York, NY.
- Encyclopedia of Britannica. Library of Alexandria.
- Ören TI (1993) Needs of the Software Industry in the Next Decade. Proceedings of the National Workshop on Software Engineering Education 109-111.
- Mittal S, Martin JLR (2017) Simulation-based Complex Adaptive Systems. *Guide to Simulation-based Disciplines* 127-150
- Mittal S, Zeigler BP (2017) Theory and Practice of M&S in Cyber Environments. The Profession of Modeling and Simulation: Discipline, Ethics, Education, Vocation, Societies and Economics, John, Wiley and Sons
- Mittal S (2014) Model Engineering for Cyber Complex Adaptive Systems. European Modeling and Simulation Symposium, Bordeaux, France
- Sage AP (1990) Software Systems Engineering. Wiley-Interscience
- Esparza, Grumberg O (2016) Dependable Software Systems Engineering. *losPr Inc.*
- Tolk A, Ören TI (2017) The Profession of Modeling and Simulation: Discipline, Ethics, Education, Vocation, Societies and Economics. John Wiley & Sons.