Model-Based Systems Engineering: An Emerging Approach for Modern Systems

Ana Luísa Ramos, *Member, IEEE*, José Vasconcelos Ferreira, and Jaume Barceló

***Abstract*—To engineer the modern large, complex, interdisciplinary systems-of-systems (SoS), the collaborative world teams must "speak" the same language and must work on the same "matter." The "matter" is the system model and the communication mechanisms must be supported by standard, flexible, and friendly modeling languages. The evolving model-based systems engineering (MBSE) approach is leading the way and is expected to become a standard practice in the field of systems engineering (SE) in the next decade. As an emerging paradigm for the systems of the 21st century, it seems useful to overview its current state of the art concerning the developing standards, the embryonic formalisms, the available modeling languages, the methodologies, and the major applications.**

***Index Terms*—Model-based systems engineering (MBSE), modeling, standards.**

I. SYSTEMS ENGINEERING AT A GLANCE

THE contemporary world is crowded of large, interdisciplinary, complex systems made of personnel, hardware, software, information, processes, and facilities. An integrated holistic approach is crucial to develop these systems and take proper account of their multifaceted nature and numerous interrelationships. As the system's complexity and extent grow, the number of parties involved (i.e., stakeholders and shareholders) usually also raises, thereby bringing a considerable amount of points of view, skills, responsibilities, and interests to the interaction.

The field of systems engineering (SE) aims to tackle the complex and interdisciplinary whole of those sociotechnical systems, thereby providing the means to enable their successful realization. Its exploitation in our modern world is assuming an increasing relevance noticeable by emergent standards, scientific journals and papers, international conferences, and academic programmes in the field. This significance is probably due to the escalating complex and "hasty" nature of our present-day systems and the interest in achieving their overall "maximum" performance through cooperative, integrative, adaptable, and interoperable environments.

The challenge is getting higher as the classical systems are evolving to complex systems-of-systems (SoS) [1], [2], including both technological and social contexts [3], [4], thereby involving a considerable component of customized services with complex human-centered aspects [5] and incorporating an extensive set of challenging requirements, like flexibility, sustainability, real-time capability, adaptability, expandability, reliability, usability, and delivery of value to society [6].

A. Systems for Systems Engineering

The SE field can be either classified as an application of the systems science, and consequently, its perspective is the one of the systems thinking "One could imagine a science of relationships underlying SE" [7] and, as a branch of engineering, with relatively new tradition and characterized by the professional creative application of scientific principles to the design and development of systems. According to Wymore [8], engineering is "the creative exploitation of energy, materials and information in organized systems of men, machine and environment, systems which are useful in terms of contemporary human values."

The definitions of SE, which began to be formalized in the 1970s with the first U.S. military standard, are numerous and diverse; however, they all share the underlying concepts of the systems approach, like holism, synthesis, interrelationships, as well as the engineering-project-based ideas of system life cycle and requirements. The classical definitions, from the 1970s, are still used, but their focus was mainly on the translation of requirements to design. The following ones, from the 1990s and 2000s, are more expanded embracing a more holistic perspective, the emergent properties, and the sociotechnical aspect. The definition from the International Council on SE (INCOSE) can be understood as a consensus of the mentioned different perspectives: "An interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customers needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem" [9].

Surprisingly, in a recent and evolving field, there are already references to "the old SE" (or the traditional, the classical, the ordered) and "the new SE" [6], [7]. This evolution has been reflecting predominantly the nature of the systems to engineer, which, in turn, reflect the tremendous and continuous advances in the technological and societal fields.

The classical systems (i.e., the system-as-machine paradigm) were small to large-scale, multidisciplinary, relatively stable

Manuscript received June 30, 2010; revised September 13, 2010 and November 11, 2010; accepted January 8, 2011. Date of publication March 3, 2011; date of current version December 16, 2011. This work was supported in part by the Portuguese Foundation for Science and Technology under Grant Sistema de Formação de Recursos Humanos/Bolsas de Doutoramento/43892/2008. This paper was recommended by Associate Editor S. H. Rubin.

A. L. Ramos and J. V. Ferreira are with the Unidade de Investigação em Governança, Competitividade e Políticas Públicas, Department of Economics, Management, and Industrial Engineering, University of Aveiro, 3810-193 Aveiro, Portugal (e-mail: aramos@ua.pt; josev@ua.pt).

J. Barceló is with the Department of Statistics and Operations Research, Technical University of Catalonia, 08034 Barcelona, Spain (e-mail: jaume.barcelo@upc.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCC.2011.2106495

and predictable, without people as a component, and were typically from the aerospace and defense industries. The new ones (i.e., the system-as-organism paradigm), which must cope with the global challenges of sustainable development, are large scale, complex, adaptive, interoperable, scalable, technology-intensive, human integrative, and comprise; for example, the so-called "super systems," like transportation and sustainable energy [10]. The perspectives of the different shareholders and stakeholders, which may be conflicting and competing, must be synthesized and resolved to serve the highest order system of interest needs [6].

This emerging metafield of study, in a synergistically coevolution with SE and aiming to add a broader context to the field, is called engineering systems, which is "a field of study taking an integrative holistic view of large-scale, complex, technologically enabled systems with significant enterprise-level interactions and sociotechnical interfaces" [6]. There are some other references that label this new field as complex SE [7], engineering of complexity [11], or SoS engineering [1], [2]. The trend is to evolve to a unified SE of the future. According to Rouse [12], SE should be an integrative discipline, exploring, understanding, and designing how everything fits together.

B. Systems-Engineering Benchmarks

A technical standard is an established norm that allows the unified utilization of criteria, terminology, methods, processes, measures, frameworks, tools, etc. The standards are unifying references necessary to institutionalize the practice of a given discipline, helping to translate the technical perspective to a more business one, helping to clarify its relevance to society, and to meet future challenges [13]. Furthermore, and in emerging collaborative world environments, they facilitate the interoperability between people and organizations. The standardization is somehow a measure of the maturity, widely expansion, and growing acceptance of a given field and, in this sense, SE is still a new area with a lack of accepted definitions and metrics [14].

The core set of SE standards is relatively new, with less than a decade, and is currently in intense development by the Standards Technical Committee of the INCOSE, the Subcommittee Seven of the International Organization for Standardization (ISO), the International Electromechanical Commission (IEC), the Institute of Electrical and Electronics Engineers (IEEE), and the Object Management Group (OMG). The first standards in the SE field have risen from the American military and aerospace industries, in the 1970s and 1980s, and were dedicated to the engineering process or, in other words, to the "WHAT" activities are to be performed. Since then, there has been an effort to take these standards to be domain independent in order to be applicable across different sectors and to be international.

According to Friedenthal [15] the taxonomy of the SE benchmarks includes five major areas: the process, the architecture frameworks (AFs), the methodologies, the modeling tools, and the data/model interchange mechanisms.

The process standards still constitute the predominant core of norms, being the ISO/IEC 15288: "Systems and software engineering—System life-cycle processes," from 2002 (revised in 2008), the most-relevant updated international benchmark. There has been a growing effort to integrate the systems and software-engineering processes, along with hardware and human engineering processes due to the increasing criticality of software within systems and to the increasing emphasis on user-intensive systems and value generation.

Besides the process standards, the fundamental core that provides a foundation for a SE approach, there are other standards in the field. The AFs is one of those groups, which includes the standard frameworks that have been developed to support systems' (and software) architecting. According to Cloutier and Verma [16], a framework is a logical structure or an organizational skeleton used to classify concepts, terminology, data, artifacts, etc. There are several established AF typically oriented for a given target domain. The enterprise architecting, the systems architecting, and the software architecting are the classical contexts [17], [18]. In the first group, we found the well-known "Zachman Enterprise Framework," as well as the The Open Group Architecture Framework (TOGAF) and the Federal Enterprise Architecture Framework (FEAF) [19]. The systems' architecting has been described through the U.S. Department of Defense Architecture Framework (DoDAF) and the MoDAF frameworks. Finally, the software architecting has been represented by the 4+1 view model of architecture [20] and by the more recent model-driven architecture (MDA), from the OMG.

The methodologies is another group with potential upcoming benchmarks in the field. The harmony SE, the object-oriented SE method (OOSEM), the rational unified process for SE (RUP SE), and the object-process methodology (OPM) are informal methodological principles that will mature and may become established norms in the next decade.

The modeling tools will be further described in the next section.

The data/model interchange mechanisms support data and model exchange among tools. The unified-modeling-language (UML) based modeling languages have a common foundation known as OMG metaobject facility (MOF) (which is also an ISO standard ISO/IEC 19502: 2005), an extensible integration framework to define, manipulate, and integrate metadata and data in a platform-independent manner. The XML metadata-interchange (XMI) specification, which is also from OMG (as well as an ISO standard ISO/IEC 19503: 2005), enables the interchange of metadata between UML-based modeling tools, like UML or SysML, and MOF-based metadata repositories in distributed heterogeneous environments, through the XML (eXtensible Markup Language). Probably, the most-relevant and inclusive standard in this area will be the norm STEP/ISO 10303: AP233 (Industrial automation systems and integration: Product data representation and exchange—Part 233: SE data representation). Still under development, this standard is a modular vendor neutral format for interchange of SE data and to support interoperability among tools.

These (formal/informal) standards constitute the core set of norms that have been driven the development of SE. This standardization is crucial to advance the field and to establish benchmark practices across different domains.

II. MODELING FUNDAMENTALS

Modeling is a universal technique to understand and simplify the reality through abstraction. From brain representations to computer simulations, the models are pervasive in the modern world, being the foundation of systems' development and systems' operation.

A model (the term "model" derives from the Latin word *modulus*, which means measure, rule, pattern, example to be followed [21]) is a representation of a selected part of the world, the domain of interest, that captures the important aspects, from a certain point of view, simplifying or omitting the irrelevant features [22]. Ludewig [21] described three criteria that a model must meet in order to be elected as a model: mapping criterion (there is an original object or phenomenon that is mapped to the model), reduction criterion (not all the properties of the original are mapped on to the model, and this one must mirror at least some properties of the original; this is the real strength of models), and pragmatic criterion (the model is useful, i.e., can replace the original for some purpose).

According to Rumbaugh *et al*. [22], the models are important to do the following:

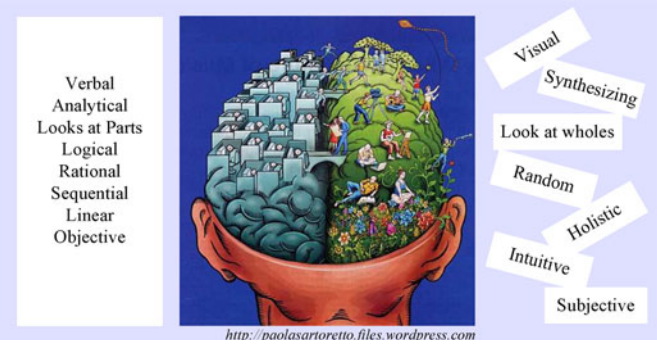
1) Capture and state requirements and domain knowledge so that all stakeholders may understand them.
2) Think about the design of a system.
3) Produce usable work products.
4) Organize, find, examine, filter, manipulate, and edit information about large systems.
5) Explore several solutions operationally, economically, and environmentally.
6) Master complex systems.

Sussman [23] added the importance of using models to gain insight into complex systems, to do experimentation, to operate systems in real time, and to negotiate, with conflicting parties, how the system will be deployed. Buede [24] reinforced the need of modeling in order to gain insight into how the world functions.

According to the Stanford Encyclopedia of Philosophy [25], the models are vehicles to explore, to understand, and to learn about the world, where this cognitive function is the basis of the so-called "model-based reasoning." Learning occurs with denotation (i.e., defining a representation relation between the model and the target), demonstration (i.e., investigating the characteristics of the model in order to demonstrate theoretical conjectures), and interpretation (i.e., converting the findings into claims about the target system) [26]. These activities require a deep analysis of the system to be modeled enhancing its understanding.

The modeler tends to shape his view of the system according to his favorite(s) modeling approach(es). From qualitative network models to quantitative kinetics-based approaches, the "art" of choosing the best approach, and representing the model adequately, in order to answer to the target questions, constitute the major characteristics of a good modeler. Frequently, those decisions are closely related to time and budget constraints and the availability of data.

The success of the model is measured by their users in different ways and according to their perspective/expectations of/on the model purpose. Criteria, such as reliability, completeness, accuracy, power to convince, ease of use, compatibility, run time, and extendibility are of frequent utilization. According to Karcanias [27], modeling is "the common basis to human activities and thus its development is also a measure of our ability to understand nature, society, and related issues."



http://paolasartoretto.files.wordpress.com

Fig. 1. Left side and right side of the brain. L-mode is the verbal, analytical, logical, rational, etc., thinking. R-mode is the visual, integrative, holistic, etc., thinking.

A. Brain Thinking

It is important to understand how the brain system functions and handles information in order to try to improve the learning processes and the mechanisms of communication.

Our brain is divided in two hemispheres, which are connected with fibbers, which interpret the world differently [28]. The left-brain thinking or the L-mode is the analytical, quantitative, verbal, rational, linear, step-by-step thinking. The right-brain thinking or the R-mode is the integrative, qualitative, holistic, creative, and visual thinking (see Fig. 1). Soliman [29] stated that the left side is predominantly analytic and sequential, while the right side seems specialized for holistic mentation being more simultaneous in its mode of operation. The latest neuroanatomical and neurophysiologic studies show that the right brain is in charge of image recognition. The pictures are images of the real world, and therefore, picture recognition is a task for the R-mode that is capable to deal with complex visual elements [29]. The pictorial representation and the amount of information that it can handle, as well as the facility to be stored in our memory, is frequently illustrated by the aphorism "A picture is worth a thousand words."

As previously referred, the SE field is concerned with the whole, the complexity, the multidisciplinarity, the holistic thinking, the synthesis and, consequently, it seems natural to identify these concerns with the R-mode, which is normally neglected in engineering curriculum [28]. According to these authors, the architecting of systems can greatly benefit from the use of the creative holistic thinking provided by the R-mode, and more easily reproduced with visual representations. As Senge [30] stated, "If we want to see system wide, we need a language of interrelationships [R-mode]."

The "ideal" performance can be achieved with the interplay between the left and the right brain, or the whole brain thinking, which allows mixing science and art, creativity and practice, words and pictures.

B. Graphical Modeling Languages

The modeling tools, which are previously referred in Section I-B, can be classified as another group of SE standards. This group includes the common representations used to describe a system. The modeling techniques used in the field of SE, to develop systems (modeling concepts, properties, attributes, structure, behavior, entities, interactions, relations, environment, etc.), have always been predominantly qualitative and based on a graphical or pictorial representation. These techniques require a corresponding describing language (i.e., graphical modeling language or visual modeling language), used to represent reality, that involves semantics (i.e., set of symbols or signs that form the basis of representations) and syntax (i.e., the proper ways of combining the symbols and signs to form thoughts and concepts).

The functional flow block diagrams (FFBDs), which are developed in the 1950s, have been, for many years, the classical representation of SE with a wide spread use within the community. This tool illustrates a step-by-step sequence of a system's functional flow through a functional-decomposition approach. During the 1970s, the structured analysis and design technique (SADT) emerged as the graphical language to communicate ideas [31], and to understand and describe systems as a hierarchy of functions. In 1993, the National Institute of Standards and Technology (NIST) launched the Integration Definition for Function Modeling (IDEF0), which is a graphical notation belonging to the IDEF suite of modeling approaches and derived from the SADT. This notation was developed to represent activities or processes that are carried out in an orderly manner [32], illustrating the functional perspective of a system, the data flow, and the system control.

The enhanced FFBD (EFFBD) and the IDEF0 have been the main modeling languages used in SE in the last decades. Other tools include, for example, the N^2 charts, state-transition diagrams, and petri nets.

These traditional functional decomposition procedures/representations are being "replaced" by object-oriented approaches. The modern object-oriented practices, with its roots in software engineering, are now pervasive in the systems engineering field. Oliver *et al.* [33] traced the origin of the object-oriented paradigm back to the 1970s, with the development of abstract data types and the introduction of classes to programming languages, like Simula67, in order to provide procedure, data, and control abstractions. In the 1990s, these principles have been extended to the analysis and design of software, through the Booch method, the object modeling technique (OMT), and then through the *de facto* UML [34]. The characteristics of the software systems are different from those of the systems of SE (that may also include software components), and consequently, the UML lacks support from aspects, like the whole/part decomposition, or the interconnections provide by physical things (and not by compilers), or the trade studies.

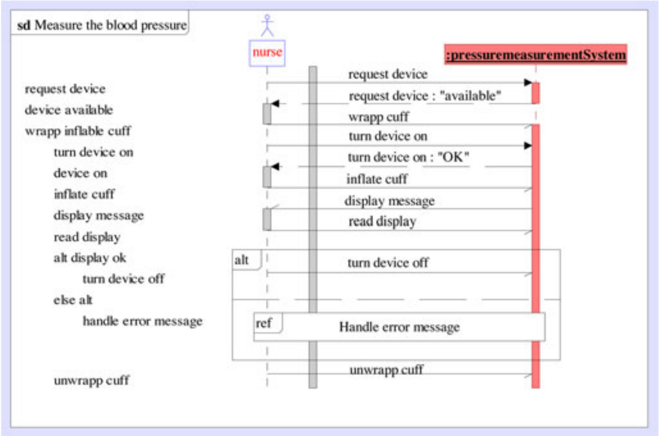


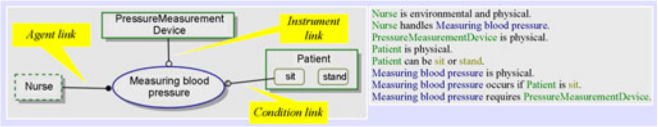
Fig. 2. Simplified *sequence diagram (sd)* for the use case "measure the blood pressure," including an weak sequencing of occurrences, two lifelines, activations, a series of synchronous, asynchronous and reply messages, an alt operand of a combined fragment for the alternative courses of action derived from the displayed message at the measurement device, and an interaction use (frame reference) that specifies an interaction described on other sd.

In order to incorporate these and other features, the OMG and the INCOSE have joined efforts and developed an extension of UML for SE: the systems modeling language (SysML) [15], which was released in 2007. This graphical SysML, which supports the specification, analysis, design, and verification of complex systems, is considered as the next *de facto* modeling language for SE. According to Oliver *et al.* [33] "SysML continues to lack a few of the needed concepts, but has extended others in useful ways beyond historic SE practice."

The SysML-modeling tools usually store the user model as structured data in a model repository, and the model enters and retrieves that information by using the graphical representation that is, the diagrams. The SysML diagrams, which reflect various aspects of a system, are nine and are organized in four major blocks that are known as the four pillars of SysML and represent four key modeling facets: the requirements of the system, the structure, the behavior, and the parametric relationships (Fig. 2 depicts an example of an SysML diagram developed in the Artisan Studio tool). These different views match particular viewpoints (the stakeholders' perspectives) and enable the holistic approach required by SE.

As UML, the modeling language for SE is not attached to any methodology. The SysML also supports model and data interchange via the XMI and via the evolving neutral ISO AP233 standard (this application protocol aims to support the exchange of data during the whole system development lifecycle and across different domain engineering disciplines allowing the creation of one consistent view of the system). The XMI provides interoperability capabilities such as, to export selected parts of an SysML model (in the model repository) to another UML tool in order to support software development, and to import and export parametric diagrams relating data to engineering analysis tools [15].

The unified profile for DoDAF/MoDAF (UPDM) is also an extension of UML to describe SoS and enterprise architectures compliant with DoDAF and MoDAF requirements.



Nurse is environmental and physical.
Nurse handles Measuring blood pressure.
PressureMeasurementDevice is physical.
Patient is physical.
Patient can be sit or stand.
Measuring blood pressure is physical.
Measuring blood pressure occurs if Patient is sit.
Measuring blood pressure requires PressureMeasurementDevice.

Fig. 3. Example of an (left) OPD and (right) the corresponding OPL.

This profile is particularly tailored for military acquisition programs.

The OPM, which was founded by Dori in 2002, and the corresponding graphical and textual representations, object process diagrams (OPDs), and object process language (OPL), enlarge the domain of object-oriented-modeling tools for SE. The provided bimodality (i.e., graphical and textual) facilitates the understanding of complexity since it is very similar to the power of both sides of the brain, i.e., the right side that acts like the visual interpreter and the left side that acts like the language interpreter. According to Grobshtein and Dori [36], this intuitive dual notation provides a single model that is comprehensible to the different stakeholders (both technical and nontechnical) involved in the development process.

They are available at the software environment object process CASE tool (OPCAT). According to Booch *et al*. [34], OPM "is a comprehensive novel approach to SE. Integrating function, structure, and behavior in a single, unifying model, OPM significantly extends the system modeling capabilities of current object-oriented methods." Fig. 3 provides an example of an OPD and the corresponding OPL.

The OPM is based on three fundamental aspects of a system: the structure (how it is made), the function (what it does), and the behavior (how it changes over time). The function is enabled by the architecture of the system that combines the structure and the behavior. The graphics (i.e., OPDs) and the natural language (i.e., OPL) express these characteristics in a unified frame of reference that corresponds to an integrated single model.

The SysML and the OPDs/OPL constitute the current state-of-the-art systems modelling languages. SysML being a more "institutionalized/standardized" language with the support of the OMG and the INCOSE, and the OPDs/OPL a more intuitive simpler language with less training effort, it seems interesting to combine the advantages of both languages creating synergies between them [36]. This integration can strongly contribute to a common understanding of the system and to improved communications between different stakeholders, as well as to a proficient SE collaborative development environment.

According to Wilkiens [37] there will be, in the future, a great demand to model languages since systems will become increasingly complex and there are considerable advantages in modeling and simulating before using them in practice. The author synthesizes the advantages of the modeling languages with the following idea: "The modeling language allows me to move on different abstraction levels. The more abstract I get the simpler the system appears to be. This is the art of being concrete on an abstract level."

III. MODEL-BASED SYSTEMS ENGINEERING FOR MODERN SYSTEMS

Model-based SE (MBSE) is an emerging approach in the SE field [6], [38], and can be described as the formalized application of modeling principles, methods, languages, and tools to the entire lifecycle of large, complex, interdisciplinary, sociotechnical systems. A simplified definition of MBSE is provided by Mellor *et al*. [39] as "...is simply the notion that we can construct a model of a system that we can transform into the real thing."

This model-centric approach, which main artifact is a coherent model of the system being developed, contrasts with the traditional document-based one [15]. The emergence of computers in the 1950s and 1960s has strongly contributed to this paradigm shift in a considerable range of engineering disciplines like the mechanical and the electrical ones, but in the SE field the transitioning process, while becoming prevalent, is still immature [6], [15], [40].

As pointed out by Bahill and Botta [41], as a fundamental principle of good system design, the essence of MBSE relies on the application of appropriate formal models to a given domain.

In the next decade, it is expected that MBSE will play an increasing role in the practice of SE and that will extend its application modeling domains beyond hardware and software systems, including social, economical, environmental, and human-performance components [9].

A. Main Features

The emergent model-based approach aims to facilitate the SE activities through the development of a unified coherent model as the main artifact. The SE process is accomplished with increasing detailed models that are all part of the system model. The major potential advantages of this approach include enhanced communications between the stakeholders and team members as well as a true shared understanding of the domain, improved knowledge capture, design precision and integrity without disconnections among the representations of data, better information traceability, enhanced reuse of artifacts, and reduced development risk. As Friedenthal *et al*. [15] stated, "the emphasis is placed on evolving and refining the model using model-based methods and tools"; therefore, the prominence of controlling documents is now replaced by controlling the model of the system.

It is expected that this paradigm will become a standard practice in the SE field in the next years. The standards evolution in the field, including the SysML, the ISO 10303: AP233, the XMI, and the MDA are impelling the proliferation of MBSE. According to the INCOSE Vision for 2020, the future of SE will be model-based, embracing high-fidelity static and dynamic models at different levels of abstraction. The MBSE approach will expand its boundaries and all the application domains (e.g., defense, industrial, pharmaceutical and healthcare, transportation, telecommunications, energy, etc.) will be potential targets for a model-based development.

The *system model* is the main artifact of MBSE and is typically developed in a modeling language, which is available in a modeling tool (for example, SysML in Artisan Studio, OPDs/OPL in OPCAT), depicted on graphical diagrams, and contained in a model repository. This integrated-model repository so that "everyone draws from the same well" [18] will embrace all the relevant information for the system and will enable marketing research, decision analysis, environmental impact analysis, social and economical modeling, biological modeling, and other appropriate analyses.

The system model is made by interconnected modeling elements that represent the key aspects of the system, namely, its requirements, its structure, its behavior, and its parametrics [15]. This integrated specification is usually in interaction with other engineering models (e.g., simulation models, analysis models, hardware models) that address multiple aspects of the systems, originating a complete coherent development environment. This environment is, nowadays, a global one without physical barriers and geographical constraints. Consequently, the collaborative world teams must "speak" the same language and must work on the same "matter" that, in an MBSE approach, corresponds to the system model.

The potential advantages of MBSE are critical to cope with the complexity of the global development environment of modern systems. This environment demands for adaptive and accurate communication mechanisms that can support considerable dimension and interdisciplinarity, geographically dispersed teams, people, and technology as inherent parties of the systems, cooperation and concurrency of different subsystems, the integration of legacy and Commercial-Off-The-Shelf (COTS) systems, and "personalized" standards and system descriptions. The coexistence of these features and their integration along with "the system's big picture" can be enabled by an MBSE approach. Particular care must be taken in order to ensure that completeness, integration, and synchronization is aligned with focus and simplicity ("managers prefer simple models that they understand and trust, to more realistic ones" [42]). The transitioning to MBSE implies a considerable investment in processes, methods, tools and, obviously, in training [15]. The MBSE approach requires a new way of thinking and a new set of skills. The community working with the modeling tools and languages must include language/tool experts that will develop the system model and that are able to train other team members.

The MBSE metrics can be used to assess design quality, development progress, risk, and they provide an indication if the process is moving in a successful way in order to achieve a successful outcome. The metrics to evaluate the design quality embrace, typically, the satisfaction of requirements, the critical performance properties to be monitored such as reliability, and the partitioning of the design.

The development progress can be assessed, for example, by the number of use case scenarios completed, the number of requirements satisfied, the percentage of logical components that have been allocated to physical components, the completeness of the specification of interfaces and properties, the number of test cases, and verification procedures that have been accomplished. The development effort and risk can be managed through the COSYSMO model that aims to accurately estimate the time and effort associated with the SE activities [43].

B. Formalisms, Methodologies, and Applications

By the present time, the theory and formalisms of MBSE are quite inexistent. The first standards in the field are now emerging and an established MBSE body of knowledge is expected to be achieved in ten years. Nevertheless, there already exist three main formalisms that deserve special attention. One of them is more elementary and is related with the SE field, while the other two are mainly devoted to the MBSE discipline.

The first formalism is a *semantic glossary and model for SE concepts* proposed by Oliver *et al.* [33]. They provided a set of definitions and a graphical model for the SE concepts that aims to introduce rigorous and consistent definitions in the field, which are critical to support an MBSE approach.

The second formalism refers to an *information model for system design* proposed by [44] and helps to understand the MBSE approach from the perspective of the kinds of information to be used and the associated relationships. The model suggests four main kinds of information that are interrelated: model, requirements, components, and design alternatives. The requirements specify components, the requirements may be decomposed into other requirements, components may be decomposed into other components, design alternates satisfy requirements, design alternates represent components, models execute design alternates, and models represent components. By the end of the design (i.e., after a concurrent incremental process), there should be only one design alternate (i.e., the best according to given criteria) and the models must become sufficiently faithful for compliance assessment.

The third formalism corresponds to a *mathematical model for SE and MBSE* that was introduced in 1993 by Wymore in his book *Model-Based SE: An Introduction to the Mathematical Theory of Discrete Systems and to the Trycotyledon Theory of System Design* and is informally known as Wymorian theory. The book provides a rigorous mathematical framework as the basis for the development of models and designs for large-scale, complex systems. Since each person has an internalized notion of system his seminal work was devoted to establish a (universal) mathematical formalization of the concept of "system" based on set theoretic concepts and based on system models. A system model is a description that separates the perceived universe into two parts: the "inside" of the system, which is described by states, and the "outside" of the system from, where the inputs come and to where the system delivers its outputs [45].

These contributions help to establish coherent and unambiguous foundations for the MBSE paradigm. They should evolve in the next years and provide the desired body of knowledge required to elevate the MBSE approach to a truly scientific discipline.

The *methodologies* for MBSE are implementations of specific processes. According to Friedenthal *et al.* [15], a methodology is "a set of related activities, techniques, and conventions that implement one or more processes and is generally supported by a set of tools." According to Estefan [46], an MBSE methodology is a set of related processes, methods, and tools used to support the discipline of SE in a model-based context. The process is the set of interacting activities that transform the inputs into outputs or, in other words, the WHAT activities are to be performed. The method specifies the techniques to perform the tasks of the process, i.e., the HOW to execute. The tools are the resources applied to the method in order to improve the efficiency of the tasks, thus enhancing the WHAT and the HOW. An MBSE methodology gathers all these pieces, thus implementing a given process, which is supported by a given method, which is enhanced by a set of tools. The capabilities and limitations of the surrounding environment, including the technologies and the people, enable or disable the methodology and the resulting success or failure of the system's development. One of the primary artifacts of an MBSE methodology is the system model.

Analyzing the main methodologies presented in [46] and [47], one can see that they are particularly focused on the implementation of the concept and development phases of the SE process. In fact, it is in these stages that SE (and MBSE) can provide considerable value-added. A synthesis of the main characteristics of these methodologies is presented in the next paragraphs and is organized according to the following structure: [name of the methodology and origin: 1) main development approach, 2) main task flow, 3) predominant modeling language, and 4) software-tool support].

Harmony SE from IBM Telelogic:

1) Consistent with the Vee model (i.e., classical top-down approach) and service-request-driven approach.
2) Requirements analysis, system functional analysis, and design synthesis.
3) SysML.
4) Rhapsody TAU.

OOSEM from INCOSE:

1) Consistent with the Vee model (i.e., classical top-down approach) incorporating object-oriented concepts and Scenario-driven approach.
2) Analyze stakeholders needs, define systems requirements, define logical architecture, synthesize allocated architectures, optimize and evaluate alternatives, validate, and verify system.
3) SysML.
4) OMG SysML tools (integrated with other engineering tools).

RUP SE from IBM Rational:

1) Consistent with the spiral model (i.e., iterative and incremental development) and object-oriented concepts.
2) Inception, elaboration, construction, transition, and use case flow down activities.
3) UML/SysML.
4) Rational method composer with RUP SE plug-in.

Vitech MBSE methodology from Vitech Corporation:

1) Concurrent design, incremental approach (i.e., "onion model").
2) Requirements analysis, behavior analysis, architecture/synthesis, and design V&V.
3) System definition language (SDL) (which is based on the ERA model), EFFBDs.
4) CORE.

OPM from Prof. Dori:

1) Object-oriented/process-oriented approach and reflective methodology.
2) Requirement specifying, analysis and designing, implementing, using, and maintaining.
3) OPDs/OPL.
4) OPCAT.

The MBSE methodologies are not, by this time, covered by formal standards but it is expected that will occurs as soon as they prove their value in real-world contexts.

The *applications* of the MBSE paradigm to real-world scenarios are beginning to be published to the community. The scientific journals and the new dedicated conferences in the field confirm it.

Probably, the first MBSE applications have arisen from the Defense and aerospace industries that are typically characterized by SoS. The dimension and complexity of these systems, with a strong technological facet, had impel the evolution of engineering solutions to deal with cost overruns, schedule delays, technology constraints, and interoperability issues. Bell Labs in the 1940s, the U.S. Department of Defense in the 1950s, and NASA in the 1960s were possibly the first ones to recognize the importance of the SE interdisciplinarity to manage and integrate large complex engineering projects.

The increasing complexity of these systems, with people, technologies, hardware, software, processes, and enterprises acting as interacting agents, demand the utilization of "intelligent and intuitive model-based SE techniques" [48].

The "MBSE challenge" team (i.e., collaboration between the INCOSE and the European Southern Observatory) is one of the most active initiatives in the application of MBSE principles to contemporary complex systems. The "telescope-modeling" project and the "space systems" project, in current development, are examples that belong to this initiative. The major goals are to apply the SysML to solve the modeling problems, to demonstrate its adequacy to support MBSE, and to create modeling guidelines for future MBSE projects. The "telescope-modeling" project involves the development of a next-generation optical telescope that must provide a continuous mirror surface. The "space systems" project is working on the FireSat system whose mission is to detect, identify, and monitor forest fires from orbit.

The project "excavator model," which will evaluate interoperability issues between modeling and simulation, is being developed by the Georgia Institute of Technology. The project involves the integration of SysML models leveraged with conventional modeling and simulation tools like mechanical CAD, factory CAD, spreadsheets, math solvers, finite element analysis (FEA), discrete event solvers, and optimization tools [49].

The Global Earth Observation System of Systems (GEOSS), to monitor and collect information related to Earth's resources is another application example of MBSE. Rao *et al.* [50] demonstrated the use of SysML to define the GEOSS architecture and the combined utilization of colored petri nets to develop the executable simulation model. Butterfield *et al.* [51] proposed an MBSE process to develop the architecture model and system specifications, thereby emphasizing the SoS perspective.

Mandutianu *et al.* [52] described an example of a pilot application of the OOSEM methodology to design a space mission.

The study reveals some encouraging potential benefits of using an MBSE approach, such as the improved communications among model designers and stakeholders, the consistent and complete representation of system models across different missions and phases, the reduction of errors and ambiguity, the reduction of design and maintenance costs, and the saved time and resources.

Simpkins *et al*. [53] presented a practical application of MBSE, using the Vitech methodology, thereby leading to an integrated and convergent solution for an automated parking system. The major benefits pointed out involve a better insight of the problem, a faster response to stakeholders' inquiries, a more rigorous traceability, and automated consistency checking and documentation.

Soyler and Diakanda [54] proposed the adoption of the MBSE holistic approach to capture the structure and behavior of disaster-management systems and to deal with their complexity. The SysML is used to realize the model-based paradigm.

The utilization of MBSE principles in the manufacturing domain is discussed in [55]. There are several case studies analyzed that aim to discover if the workers want to move from a document-based approach to model-based working. The results disclose the need to implement different modeling levels and strategies to engage domain workers in modeling activities.

Andersson *et al*. [56] described the lessons learned when introducing an MBSE approach, using UML/SysML, at Saab Aerosystems. The approach is considered to have high potential to improve engineering productivity and quality. However, there is a clear need to instigate effective modeling training programs, with particular emphasis on model-based methods and tools.

Haan [57] described an application of MBSE to the health-management field. An SysML model is used to demonstrate the potential competitive advantage of prognostics and health management. As the author stated "...MBSE methods are clearly applicable and should be highly sought by enterprises wishing to finesse a competitive advantage from PHM technologies."

The INCOSE MBSE initiative is also working on the urban transportation field, along with the Florida Department of Transportation, using the cases of an urban traffic signal and a highway-maintenance system. These projects are quite immature and require further advances.

The importance of these diverse real-world applications and the true essence of MBSE is quite highlighted in the following statement "...the specific tool, or language, or approach, is not the important thing; rather, systems engineers should model to understand the problem, and to communicate with others about the problem. If your modeling approach helps you accomplish that, it is a good thing" [58]. The idea is corroborated by Rasmussen as "the benefit of formal modeling is that we can finally stop being ambiguous and say exactly what we mean" [59].

A balance provided in [60], which is a resultant from experiences of pioneer applications of the MBSE approach, points out the following major guidelines for the successful implementation of an MBSE environment: The MBSE cultural change must be supported by an organizational change and continuous improvements principles; a well-defined MBSE methodology is decisive to support the development of the system model; adequate and customized training in languages, methods, and tools is vital as well as continuous mentorship; pilot projects are required to test and validate the model-based approach; well-defined modeling purposes, objectives, and scope are essential to properly manage the stakeholders' expectations, which are "the most-noticeable measure of the MBSE project success."

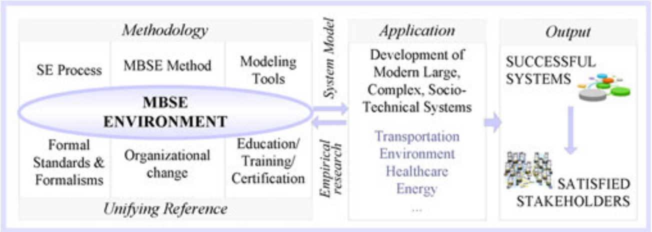


Fig. 4. Integrated MBSE environment.

IV. VISION FOR THE FUTURE

The different MBSE aspects that have been discussed through the previous sections (i.e., standards, formalisms, methods, modeling tools, applications, etc.) can be considered as interacting dimensions that must work together to achieve the main result, which is an MBSE environment able to lead to a successful system. This success is measured by the fulfillment of the stakeholders' expectations. This integrated vision is illustrated in Fig. 4.

The future of MBSE will be facilitated by the continuously evolving information technologies (computing power, storage and analysis capacities, distributed capabilities, virtual networking, etc.) as well as by the fine-tuned profile of modern systems engineers (the proliferation of SE courses at the various graduation levels and the adaptive profile innate to the new generations will contribute to the systems engineer of the future).

The emergence of the MBSE discipline is now well visible in the new dedicated conferences that are flourishing such as the International Conference on SE and Modeling sponsored by the IEEE, the Technion, and the INCOSE (with a second edition that took place in 2009 under the designation MBSE 2009), and the Association for Computing Machinery (ACM)/IEEE International Conference on Model-Driven Engineering Languages and Systems (MODELS) that is, in 2010, in its 13th edition, but has been, until 2007, under the designation of the International Conference on the UML.

The potential of MBSE can only be realized if the required cultural and technical challenges will be overcome. The market forces and the field visionaries must "push the envelope to demonstrate value, exploiting opportunities, and setting an example for others to follow" [9].

Some concrete recommendations for advancing MBSE, from several specialists in the field, include the development of metrics and a value model for MBSE, the promotion of the use of modeling tools and interoperability support/standards, the development of a "human centric" MBSE establishing the bridge between cognitive and systems engineers, the identification of MBSE best practices, the advancement of standards such as OMG SysML and AP233, the integration between SysML and simulation standards, the sharing of knowledge across the SE/MBSE community, and the development of an MBSE certification in education.

According to our vision, the fundamental research lines in the field will be related with 1) the development of simple and agile MBSE methodologies and 2) the effective utilization of graphical modeling languages able to support collaborative development environments and successful stakeholders' communication/interactions thus successful systems.

In the first case, the development of an integrated methodology with simple, lean, and customizable processes and methods is of paramount importance to enable the widen utilization of MBSE practices. The SE process (i.e., WHAT) must be intuitive, logical, universal, and easy to use and tailor. According to our opinion, the ISO/IEC 15288 processes standard requires some integration that can be provided by the SIMILAR process model. The SIMILAR acronym stands for *s*tate the problem, *i*nvestigate alternatives, *m*odel the system, *i*ntegrate, *l*aunch the system, *a*ssess performance, and *r*eevaluate. In 1998, in this same journal, Bahill and Gissing [61] had suggested this general process as a universal way of planning and problem solving closely related to human thinking. After a decade, the process stills extensive and straightforward, but must be contextualized in the framework of the international SE processes standard that has emerged since then. Some seminal work in this subject is provided in [62]. The MBSE method specifies the HOW to execute the process and relies on the development of a coherent model of the system. This area has significant research opportunities since the existing methods (e.g., OOSEM, RUP SE, and OPM) are still immature and require a proof of value in real-world contexts. The methods based on more agile iterative and incremental development approaches and supported by the state-of-the-art modeling languages will probably be the ones that can lead the way to formal standards.

In the second case, the challenge will be to integrate the existing benchmark graphical modeling languages, such as SysML and OPDs/OPL, to create an effective collaborative development environment. These two languages are considerably different in terms of size and complexity. SysML is fairly large, rich, and comprehensive, appropriate to provide a detailed description of the system, and uses a standard notation supported by several commercial tools but is cumbersome and requires significant learning efforts (usually, the nontechnical stakeholders are not able to work with this language). The OPDs/OPL is a language more compact, simple, and easy to learn and use and is more adequate to model the high-level concepts.

The synergies between these two languages can strongly contribute to a common understanding of the system and to improved communications between different stakeholders. As we know, the communication is as more difficult as we bring to the dialog people with different skills, points of view, responsibilities, and interests. Some automation mechanisms to convert one language into another are already being worked by Grobshtein and Dori [36]. We think that the creation of agile tools, like matrices, to assist the system's modeling process will be important. For example, we can develop matrices for the SIMILAR process entailing a classification of stakeholders and models that will help the system's developer(s) to choose the appropriate model(s) to use in a given phase with a particular group of stakeholders (e.g., to *state the problem* and to incite a discussion with local governors and academic researchers the system's engineer will look at the "S" matrix, for the right entry (governors × researchers) and will pick the indicate model(s) like an OPD system diagram to define the boundaries of the system and the main constituents and an SysML requirement diagram to describe system's requirements and their relationships).

We believe that the major developments (which will contribute to the establishment of a reliable MBSE unifying reference, made of formal standards, organizational culture, and high-quality education/training) will be accomplished through accredited SE/MBSE-centric programs and noteworthy empirical research.

The centric programs, at the basic, master, and doctoral levels, will be fundamental to provide systems engineers with the technical, communicational, modeling, and leading skills and competences that are critical to connect people and information, to cope with holism, flexibility, multidisciplinarity, human behavior, scalability, and risk, and to solve problems creatively delivering value to society. This holistic education should be complemented by domain-specific disciplines, such as energy and environment or healthcare. The empirical research will be essential to drive the evolution of MBSE knowledge and to help to establish a coherent unifying reference. The experimental observations are fundamental to understand the real modern complex systems, and they can be used to test MBSE hypotheses, to develop MBSE standards, and to create MBSE theories. It is our opinion that this empirical work will have as target domain the complex super systems that aim to deliver world sustainability. The traffic and environment, the energy, and the healthcare are examples of these large, complex, and heterogeneous systems.

We are convinced that MBSE will be, in the next decade, a fundamental paradigm for the development of modern 21st century complex systems and will be crucial to support effective collaborative development environments. The main challenge will be to ensure that the *system model* reflects the stakeholders' ideas and positions acting as a shared working platform, and the resulting *system* satisfies their expectations.

REFERENCES

[1] M. Jamshidi, "System of systems engineering–new challenges for the 21st century," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 23, no. 5, pp. 4–19, May 2008.

[2] J. Lane and B. Boehm, "System of systems lead system integrators; where do they spend their time and what makes them more or less efficient?" *Sys. Eng.*, vol. 11, pp. 81–91, Spring 2008.

[3] C. Haskins, "Using patterns to transition systems engineering from a technological to social context," *Syst. Eng.*, vol. 11, pp. 147–155, Summer 2008.

[4] H. de Bruijn and P. Herder, "System and actor perspectives on sociotechnical systems," *IEEE Trans. Syst., Man, Cybern. A, Syst, Humans*, vol. 39, no. 5, pp. 981–992, Sep. 2009.

[5] J. Tien, "On integration and adaptation in complex service systems," *J. Syst. Sci. Syst. Eng.*, vol. 17, pp. 385–415, Dec. 2008.

[6] D. Rhodes, "Addressing systems engineering challenges through collaborative research," in *SEARI—Systems Engineering Advancement Research Initiative*. Cambridge, MA: MIT Press, 2008.

[7] S. Sheard and A. Mostashari, “Principles of complex systems for systems engineering,” *Syst. Eng.*, vol. 12, pp. 295–311, Winter 2009.

[8] A. Wymore, *Model-Based Systems Engineering*. Boca Raton, FL: CRC, 1993.

[9] INCOSE, in *Systems Engineering Handbook—A Guide for System Life Cycle Processes and Activities*, C. Haskins, K. Forsberg, and M. Krueger, Eds. San Diego, CA: Int. Council Syst. Eng., 2007.

[10] D. Hybertson and S. Sheard, “Integrating and unifying old and new systems engineering elements,” *INSIGHT-INCOSE J.*, vol. 11, pp. 13–16, Feb. 2008.

[11] E. Honour, “Systems engineering and complexity,” *INSIGHT-INCOSE J.*, vol. 11, pp. 20–21, Feb. 2008.

[12] W. Rouse, “Engineering complex systems: Implications for research in systems engineering,” *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 33, no. 2, pp. 154–156, May 2003.

[13] S. Arnold, “Where is standardization guiding us?” *INSIGHT-INCOSE J.*, vol. 10, pp. 41–43, Apr. 2007.

[14] R. Valerdi and H. Davidz, “Empirical research in systems engineering: Challenges and opportunities of a new frontier,” *Syst. Eng.*, vol. 12, pp. 169–181, Summer 2009.

[15] S. Friedenthal, A. Moore, and R. Steiner, *A Practical Guide to SysML, The Systems Modeling Language*. Burlington: Morgan Kaufmann/OMG, Elsevier, 2008.

[16] R. Cloutier and D. Verma, “Applying the concept of patterns to systems architecture,” *Syst. Eng.*, vol. 10, pp. 138–154, Summer 2007.

[17] A. Tang, J. Han, and P. Chen, “A comparative analysis of architecture frameworks,” Centre Component Software Enterprise Syst., School Inf. Technol. Swinburne Univ. Technol., Melbourne, Vic., Australia, Technical Rep. SUTIT TR2004.01, 2004.

[18] T. Browning, “The many views of a process: Toward a process architecture framework for product development processes,” *Syst. Eng.*, vol. 12, pp. 69–90, Spring 2009.

[19] M. Richards, N. Shah, D. Hastings, and D. Rhodes, “Managing complexity with the department of defense architecture framework: Development of a dynamic system architecture model,” Eng. Syst. Div., Mass. Inst. Technol., Cambridge, MA, Working Paper ESD-WP-2007-09, 2007.

[20] P. Kruchten, “The 4+1 view model of architecture,” *IEEE Softw.*, vol. 12, no. 6, pp. 42–50, Nov. 1995.

[21] J. Ludewig, “Models in software engineering—An introduction,” *Softw. Syst. Modelling*, vol. 1, pp. 5–14, Mar. 2003.

[22] J. Rumbaugh, I. Jacobson, and G. Booch, *The Unified Modeling Language Reference Manual*. Reading, MA: Addison-Wesley Longman, 1999.

[23] J. Sussman, *Introduction to Transportation Systems*. Dedham, MA: Artech House, 2000.

[24] D. Buede, *The Engineering Design of Systems—Models and Methods*, 2nd ed. New York: Wiley, 2009.

[25] Stanford Encyclopedia of Philosophy. *Models in Science.* [Online]. (2006). Available: http:/plato.stanford.edu/entries/models science

[26] R. Hughes, “Models and representations,” *Philos. Sci.*, vol. 64, pp. 325–336, 1997.

[27] N. Karcanias. (2004). Modelling and simulation in technological and emerging fields: Emerging challenges. Workshop: SIM SERV Modelling and Simulation Challenges, Working Group Roadmap for Continuous and Hybrid Simulation, White Paper, Greece [Online] Available: http:/www.sim-serv.com/wg_doc/WG7_General_roadmap.pdf

[28] T. Di Carlo, B. Khoshnevis, and F. Udwadia, “Whole brain thinking in systems architecting,” *Syst. Eng.*, vol. 12, pp. 265–273, Autumn 2009.

[29] S. Soliman (2005). Systems and creative thinking. Center for Advancement of Postgraduate Studies and Research in Engineering Sciences, Faculty of Engineering, Cairo University, Cairo, Egypt [Online]. Available: http:/www.pathways.cu.edu.eg/subpages/Creativity-Engine.htm

[30] P. Senge, *The Fifth Discipline—The Art & Practice of the Learning Organization*. New York: Doubleday, 1990.

[31] D. Ross, “Structured analysis (SA): A language for communicating ideas,” *IEEE Trans. Softw. Eng.*, vol. SE-3, no. 1, pp. 16–34, Jan. 1977.

[32] C.-H. Kim, R. Weston, A. Hodgson, and K.-H. Lee, “The complementary use of IDEF and UML modelling approaches,” *Comput. Ind.*, vol. 50, pp. 35–56, 2003.

[33] D. Oliver, J. Andary, and H. Frisch, “Model based systems engineering,” in *Handbook of Systems Engineering and Management*, 2nd ed., A. Sage and W. Rouse, Eds. Hoboken, NJ: Wiley, 2009, pp. 1361–1399.

[34] G. Booch, J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*. Reading, MA: Addison-Wesley Longman, 1999.

[35] D. Dori, *Object-Process Methodology: A Holistic Systems Paradigm*. New York: Springer, 2002.

[36] Y. Grobshtein and D. Dori, “Creating SysML views from an OPM model,” in *Proc. 2nd Int. Conf. Model Based Syst. Eng.*, Herzelya and Haifa, Israel, Mar. 2009, pp. 36–45.

[37] T. Wilkiens, *Systems Engineering with SysML/UML—Modeling, Analysis, Design*. Burlington, MA: Morgan Kaufmann/OMG, 2007.

[38] J. Grady, “Universal Architecture Description Framework,” *Syst. Eng.*, vol. 12, pp. 91–116, Summer 2009.

[39] S. Mellor, A. Clark, and T. Futagami, “Model driven development,” *IEEE Softw.*, vol. 20, no. 5, pp. 14–18, Sep./Oct. 2003.

[40] J. Andary and D. Oliver, “Models in systems engineering and software engineering,” *INSIGHT-INCOSE J.*, vol. 10, pp. 26–27, Oct. 2007.

[41] A. Bahill and R. Botta, “Fundamental principles of good system design,” *Eng. Manag. J.*, vol. 20, no. 4, pp. 9–17, 2008.

[42] J. Little, “Models and managers: The concept of a decision calculus,” *Manag. Sci.*, vol. 16, no. 8, pp. B466–B485, 1970.

[43] R. Valerdi, *The Constructive Systems Engineering Cost Model (COSYSMO): Quantifying the Costs of Systems Engineering Effort in Complex Systems*. Saarbrücken, Germany: VDM Verlag, 2008.

[44] L. Baker, P. Clemente, B. Cohen, L. Permenter, B. Purves, and P. Salmon, “Foundational concepts for model driven syst. design,” INCOSE Model Driven Syst. Design Interest Group, Int. Council Syst. Eng., San Diego, CA, White Paper.

[45] A. Wymore, “Systems movement: Autobiographical retrospectives, contributions to the mathematical foundations of systems science and systems engineering,” *Int. J. General Syst.*, vol. 33, no. 6, pp. 593–610, 2004.

[46] J. Estefan, “MBSE methodology survey,” *INSIGHT-INCOSE J.*, vol. 12, pp. 16–18, Apr. 2009.

[47] INCOSE, “Survey of model based systems engineering (MBSE) methodologies,” *Model-Based Systems Engineering Initiative* of the International Council on Systems Engineering [published online INCOSE TD 2007 003 01], 2008.

[48] J. Garcia, Jr., “Executable and integrative whole-system modeling via the application of OpEMCSS and Holons for model-based systems engineering,” *INSIGHT-INCOSE J.*, vol. 12, pp. 21–23, Dec. 2009.

[49] R. Peak, C. Paredis, L. McGinnis, S. Friedenthal, and R. Burkhart, “Modelling & simulation interoperability team status update,” presented at INCOSE Int. Workshop, Phoenix, AZ, 2010.

[50] M. Rao, S. Ramakrishnan, and C. Dagli, “Modeling and simulation of net centric system of systems using systems modeling language and colored petri-nets: A demonstration using the global earth observation system of systems,” *Syst. Eng.*, vol. 11, pp. 203–220, Autumn 2008.

[51] M. Butterfield, J. Pearlman, and S. Vickroy, “A system-of-systems engineering GEOSS: Architectural approach,” *IEEE Syst. J.*, vol. 2, no. 3, pp. 321–332, Sep. 2008.

[52] S. Mandutianu, M. Moshir, and K. Donahue, K., “Conceptual model for space mission systems design,” presented at 19th Ann. Int. Symp. INCOSE, Singapore, 2009.

[53] P. Simpkins, A. Kleinholz, and J. Maley, “A practical application of MBSE—The automated parking system,” presented at 3rd Asia-Pacific Conf. Syst. Eng., Singapore, 2009.

[54] A. Soyler and S. Diakanda, “A model-based systems engineering approach to capturing disaster management systems,” in *Proc. 4th Ann. IEEE Syst. Conf.*, San Diego, CA, Apr. 2010, pp. 283–287.

[55] H. Tellioglu, “Practicing modelling in manufacturing,” in *Proc. 2nd Int. Conf. Model Based Syst. Eng.*, Herzelya and Haifa, Israel, Mar. 2009, pp. 75–82.

[56] H. Andersson, E. Herzog, G. Johansson, and O. Johansson. (2009). Experience from introducing unified modeling language/systems modeling language at Saab aerosystems, *Syst. Eng*. [Online]: DOI 10.1002/sys.20156, 2009.

[57] B. Haan, “A model of the competitive advantage of prognostics and health management,” in *Proc. Ann. Rel. Maintainability Symp.*, Jan. 2009, pp. 442–447.

[58] R. Cloutier, “Introduction to this special edition on model-based systems engineering,” *INSIGHT-INCOSE J.*, vol. 12, no. 4, pp. 7–8, Dec. 2009.

[59] C. Delp, L. Cooney, C. Dutenhoffer, R. Gostelow, M. Jackson, M. Wilkerson, T. Kahn, and S. Piggott, “The challenge of model-based systems engineering for space systems, year 2,” *INSIGHT-INCOSE J.*, vol. 12, pp. 36–39, Dec. 2009.

[60] S. Friedenthal, “SysML: Lessons from early applications and future directions,” *INSIGHT-INCOSE J.*, vol. 12, no. 4, pp. 10–12, Dec. 2009.

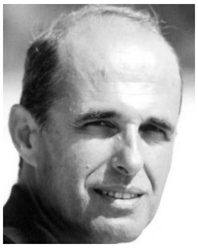
[61] A. Bahill and B. Gissing, “Re-evaluating systems engineering concepts using systems thinking,” *IEEE Trans. Syst., Man Cybern. C, Appl. Rev.*, vol. 28, no. 4, pp. 516–527, Nov. 1998.

[62] A. Ramos, J. Ferreira, and J. Barceló, "Revisiting the SIMILAR process to engineer the contemporary systems," *J. Syst. Sci. Syst. Eng.*, vol. 19, no. 3, pp. 321–350, Sep. 2010.

Ana Luísa Ramos (M'10) was born in Portugal in 1974. She received the M.Sc. degree in computer engineering from the University of Coimbra, Coimbra, Portugal, in 2002. She is currently working toward the Ph.D. degree in industrial management with the University of Aveiro, Aveiro, Portugal.

She is also an Assistant Lecturer of management and industrial engineering (MIE) with the University of Aveiro, where she was the Vice-Director of the MIE undergraduate program from 2003 to 2006 and the Coordinator of the Socrates/Erasmus Program from 2003 to 2006. She is a coauthor of the articles "Combining heuristic procedures and simulation models for balancing a PC camera assembly line," which was published in *Computers and Industrial Engineering*, and "Revisiting the SIMILAR Process to Engineer the Contemporary Systems," which was published in the Journal of Systems Science and Systems Engineering. Her current research interests include modeling and simulation and model-based systems engineering, mainly in the industrial and transportation sectors.

Dr. Ramos is a member of the International Council on Systems Engineering (INCOSE) and received the INCOSE Foundation/Stevens Institute Doctoral Award for promising research in Systems Engineering and Integration in 2009. She is also a member of the Portuguese Society for Operational Research.



José Vasconcelos Ferreira was born in Portugal in 1960. He received the M.Sc. degree in operational research and systems engineering from the Technical University of Lisboa, Lisboa, Portugal, in 1989 and the Ph.D. degree in engineering sciences from the University of Porto, Porto, Portugal, in 2005.

Since 2005, he has been an Assistant Professor of management and industrial engineering with the University of Aveiro, Aveiro, Portugal, where he is a member with the Department of Economics, Management, and Industrial Engineering Council and a member of the Industrial Engineering Doctoral Program Council. He was an Assistant Lecturer with the University of Porto. He is a coauthor of the articles "A Travelling Salesman Model for the Sequencing of Duties in Bus Crew Routes," which was published in the *Journal of the Operational Research Society,* and "Revisiting the SIMILAR Process to Engineer the Contemporary Systems," which was published in the *Journal of Systems Science and Systems Engineering*. He has more than 20 years of experience collaborating with the main urban mass-transit companies in Portugal. He works in partnership with a spin-off company dedicated to transportation planning and optimization. His current research interests include systems engineering, logistics, multivariate data analysis, and operational planning at mass-transit companies.

Jaume Barceló was born in Spain in 1944. He received the Degree in physics from the University of Valencia, Valencia, Spain, in 1967 and the Ph.D. degree in physical sciences from the Autonomous University of Barcelona, Barcelona, Spain, in 1974.

Since 1986, he has been a Full Professor of operations research with the Technical University of Catalonia (UPC), Barcelona, where, since 2007, he has been a Scientific Director of the Area of Information and Communication Technologies and Mobility. In 1985, he set up the research group that developed the microscopic traffic simulator Aimsun, and he was a cofounder and a Scientific Director of the Transport Simulation Systems (TSS), which is a spin-off company of UPC. He has been a Coordinator of several projects of the R&D programs of the European Union on Intelligent Transportation Systems. He is an author or a coauthor of more than 100 papers published in technical journals, conference proceedings, and chapters in transportation technical books. He is currently an Associate Editor of *Transportation Science and of Transportation Research Part C—Emerging Technologies*. His current research interests include optimization and simulation techniques for transportation problems.

Prof. Barceló was the Director at Large of the Institute for Operations Research and the Management Sciences (INFORMS). In 1997, he received the Narcís Monturiol Medal of the Catalan Government to Technological Innovation, and in 2001, he received the Honorific Mention Prize Ciudad de Barcelona to Technological Innovation.