Journal of Defense Modeling and Simulation: Applications, Methodology, Technology
1–14

DOI: 10.1177/1548512914568657
dms.sagepub.com

Towards a framework for more robust validation and verification of simulation models for systems of systems

Bernard P. Zeigler[1] and James J. Nutaro[2]

Abstract

We present a framework for verification and validation of simulation models of System of Systems that is based on an existing framework for modeling and simulation. The framework addresses problems arising especially in recently emerging Systems of Systems such as cyber-physical autonomous cooperative systems. The design of such systems presents challenges to the currently employed independent use of simplified models for formal verification or brute-force simulations which are severely limited in the range of conditions they can test. The proposed framework is applied to integration of formal analytic and simulation verification methods where there is a need to have confidence that the properties proved for idealized abstract models also hold in more realistic models which gave rise to the abstractions. Taking both logical and probabilistic perspectives clarifies the situation and suggests where more research is needed.

Keywords

Validation, verification, modeling and simulation, experimental frames, intended use, Discrete Event System Specification

1. Introduction

The use of simplified models in simulation model development has long been advocated as necessary in trading accuracy for practicality of execution.[1–3] Whether referred to as reduced order (derived from linear system theory[3]), abstraction (under computer science influence[4]) or metamodeling,[5] the basic idea is that the model in question is conceived to be a simplified version of a more complex model that has beneficial attributes such as being easier to analyze, simulate, or compare with reality. Especially in the realm of system design, formal Verification and Validation (V&V) methods rely on appropriately formulated abstractions that lend themselves to proofs of properties using logic-based analysis.[6] However, the ever-increasing complexity of autonomous, cyber-physical, cooperative multi-agent systems (such as unmanned aircraft of the US Air force and self-driving cars) has raised questions concerning the limitations of verification based solely on analysis of abstractions. For example, model checking, a well-known formal verification method, systematically explores the state space of a system model to check that states satisfy specified behavioral properties.[6] Model checking methods encounter state space explosion in analyzing *autonomous* systems that require complex logical processes to perform complex decision-making tasks. Moreover, because they are limited in their expressive capability to restricted logics, such methods must typically make stringent assumptions about physical components and environments. These assumptions and idealizations greatly reduce the methods' applicability to *cyber-physical systems* where the interplay of physical and computational elements is paramount. Finally, *cooperative* multi-agent systems raise the state space explosion exponentially through the cross-product of their individual state

[1]RTSync Corp. and Arizona Center for Integrative Modeling and Simulation, AZ, USA
[2]Oak Ridge National Laboratory, TN, USA

Corresponding author:
Bernard P. Zeigler, RTSync Corp. and Arizona Center for Integrative Modeling and Simulation, AZ, USA.
Email: zeigler@ece.arizona.edu

spaces. In the absence of workable simulation approaches to enable virtual testing, the only recourse for V&V of cyber-physical autonomous cooperative systems, as examples of Systems of Systems (SoS),[7] is to brute-force methods which are severely limited in the range of conditions they can test.

We claim that a key root cause of limitations in current V&V approaches to SoS is that they are not based on a general dynamic systems modeling and simulation framework. Such a framework should be capable of expressing the interaction of decision logic, discrete events, and continuous dynamics that are the hallmarks of such systems. We therefore propose that the Discrete Event System Specification (DEVS) formalism, as the computational basis for a general dynamic systems theory,[2,8] provides a sound and practical foundation for enhancing existing V&V methods to address their limitations in addressing SoS.

The value of Modeling and Simulation (M&S) in defense and other application areas is well known.[9] A DEVS model is a system-theoretic concept specifying inputs, states, and outputs, similar to a state machine.[10] Critically different, however, is that it includes a time-advance function that enables it to represent discrete event systems, as well as hybrids with continuous components[11] in a straightforward platform-neutral manner.[12,13] A recent dissertation[14] presents a multi-paradigm model-driven approach to design, verification and deployment of software intensive systems, another formulation of cyber-physical systems. It shows that DEVS provides excellent features for modeling such systems. The thesis provides a list of properties of DEVS and their mapping to properties of automotive software and systems, here viewed as instances of SoS.

- *Concurrency*: Multiple processors and communication links are concurrent in a SoS system. The semantics of DEVS coupled models supports concurrency by appropriate interleaving of the discrete-event behavior of individual sub-models.
- *Time*: Real-time performance is a crucial property of SoS embedded software. End-to-end latencies are part of the requirements for these applications. The time advance function of an atomic DEVS model can be used to model latency.
- *Events*: Event-triggered and time-triggered architectures use triggers in the form of either external events or timing events to start certain pieces of functionality. DEVS implements reaction to events using the external transition functions.
- *Priorities*: Some real-time communication channels use priority-based and other mechanisms for arbitration. DEVS supports such arbitration by means of explicit specification of executable events from the set of simultaneous events.
- *Simulation of the physical parts of the system*: DEVS is a very general formalism and is able to include different other formalisms. This generality stems from the infinite possible states that DEVS allows to model and the (continuous) time elapse between the different state transitions. Hierarchical coupling techniques are used to integrate the different formalisms using DEVS as a common denominator.

Faced with the complexity of V&V for SoS, there has been recognition of the need to establish a standard specification for intended use (IU) of a model that contains a comprehensive listing of specifics in relation to the problem that the model is intended to address for SoS.[15] As a greatly expanded formulation of modeler objectives and context, the introduction of the IU specification begs the question of how to reconcile IUs with experimental frames (EFs) and how to apply the resulting concepts to V&V of SoS. In this paper, we work toward an extended framework for V&V of simulation models of SoS that views the problem from the "modeling in the large" perspective of Zeigler.[16] This framework is based on the framework for M&S emphasizing abstractions called partial or lumped models as first proposed by Zeigler[2,17] The extended framework includes the concept of IU as a characterization of modeler objectives and an extension of the EF concept.[2,18] The proposed framework is applied to integration of formal analytic and simulation verification methods where there is a need to have confidence that the properties proved for idealized abstract models also hold in more realistic models which gave rise to the abstractions. Taking both logical and probabilistic perspectives clarifies the situation and suggests where more research is needed.

In the sequel, we first review basic systems theory that underlies the subsequent presentation of the proposed modeling and simulation framework and its application to integration of formal methods for verification within the simulation context. Discussion of more robust validation and verification of simulation models for systems of systems then follows with conclusions on where further research is needed.

We remark that a significant volume of literature has developed in the area of V&V since the 1970s that includes branches that stem from the original framework and branches that do not (and some that re-combine.) To place the current approach in context of these works enables better elucidating its contribution. This will be done in the discussion following the main presentation to exploit the clarity introduced by the framework definitions.

Table 1. Levels of system specifications.

Level	System specification	Description
4	Coupled	Hierarchical system with coupling specification. System of systems is defined at this level.
3	Atomic	State space with transitions. Behavior and internal structure are specified at this level.
1	I/O function	State space with defined initial state. Component temporal behavior with respect to initial state is defined at this level.
1	I/O behavior	Collection of I/O pairs defining temporal behavior is defined at this level.
0	I/O frame	Defines I/O variables with associated ports over a time base.

I/O: input/output

2. Systems theory background

Concepts for organizing models and data for simulation based on systems theory[18,19] and implementable in Model Based Systems Engineering[20,21] are a necessary background for discussing the *modeling and simulation framework* (MSF). The system specification hierarchy (as outlined in Table 1) provides an orderly way of establishing relationships between system descriptions as well as presenting and working with such relationships. Pairs of system can be related by morphism relations at each level of the hierarchy. A *morphism* is a relation that places elements of system descriptions into correspondence. For example, at the lowest level, two observation frames are isomorphic if their inputs, outputs, and time bases respectively, are identical. In general, the concept of morphism tries to capture similarity between pairs of systems at the same level of specification. Such similarity concepts have to be consistent between levels. When we associate lower level specifications with their respective upper level ones, a morphism holding at the upper level must imply the existence of one at the lower level. The morphisms are set up to satisfy these constraints.

The most fundamental morphism, called *homomorphism*, resides at the State Transition level Consider two systems specified at level 3, S and S', where S may be bigger than S' in the sense of having more states. S could represent a complex model and S' a simplification of it. Or S could represent a simulator and S' a model it is executing. If such a homomorphism holds for all states of S', then any state trajectory in the S' will be properly reproduced in S. Often, we require that the correspondence hold in a step-by-step fashion and that the outputs produced from corresponding states be the same. In this type of homomorphism, the values and timing of the transitions and outputs of the big system are preserved in the small one. Thus, in this case, the state and output trajectories of the two models, when started in corresponding states, are the same. However, in general the relation between state and output trajectories at different levels is more complex.

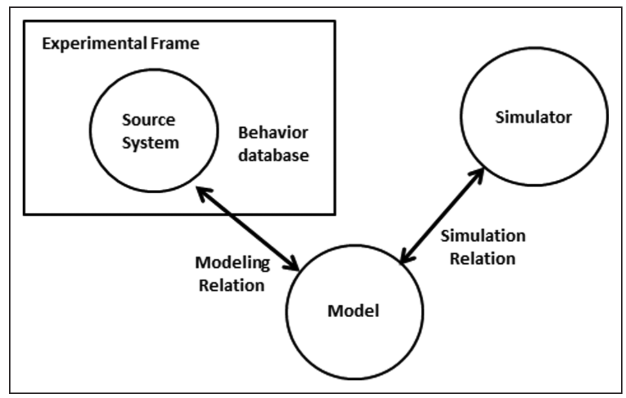

Figure 1. Basic entities in M&S and their relationships.

3. MSF

The MSF presents entities and relationships of a model and its simulation. The basic entities of the framework are: source system, model, simulator and the Experimental Frame (EF). As illustrated in Figure 1, the basic entities in M&S are the actual system (the "Source System"), the "Model," and the mechanism for executing the Model (a "Simulator") when the model generates a description of events over time. It is important to understand the relationship between the Model and the Source System (the "Modeling Relation") and the relationship between the Model and the Simulator (the "Simulation Relation"). The adequacy of the model must be judged with respect to the context of use, which includes the domain of input values, the range of output values, and the intent of the user. The EF was originally introduced to operationalize this contextual dependence of adequacy as an object in a status equal to real system, model, and simulator. In the simulation theoretic usage we employ here the MSF separates models from simulators as entities that can be conceptually manipulated independently and then combined in a relation which defines correct simulation. In addition, the EF defines a particular experimentation process for model input, state, and outcome measurements in accordance

Table 2. Defining the basic entities in M&S and their usual levels of specification.

Basic entity	Definition	Related system specification levels
Source system	Real or artificial source of data	Known at level 0
Behavior database	Collection of gathered data	Observed at level 1
Experimental frame	Specifies the conditions under which system is observed or experimented with	Constructed at levels 3 and 4
Model	Instructions for generating data	Constructed at levels 3 and 4
Simulator	Computational device for generating behavior of the model	Constructed at level 4

with specific analysis objectives. The EF formally recognizes that the IU of a model is a fundamental determinant of its validity with respect to the source system. Modular reuse, validity, and executability of simulation compositions are common aspirations among enterprises regularly relying on M&S of SoS throughout their lifecycles. Such enterprises invest significantly not only in development and experimentation, but also in V&V. The MSF helps clarify many of the issues involved in such activities. The MSF underlies the DEVS simulation protocol,[2,8,11] which provides provably correct simulation execution of DEVS models thereby obviating commonly encountered sources of errors in legacy simulations.

The basic inter-relationships among entities are the *modeling* and the s*imulation* relationships. The entities are defined in Table 2. This table also characterizes the level of system specification that typically describes the entities. The level of specification is an important feature for distinguishing between the entities, which is often confounded in practice. Based on this framework, the basic issues and problems encountered in performing M&S activities can be better understood and coherent solutions developed.

3.1 The entities of the framework

The *source system* is the real or virtual environment viewed as a *source of observable data*, in the form of time-indexed trajectories of variables. The data that have been gathered from observing or otherwise experimenting with a system are called the *system behavior database*. This concept of source system is a specification at level 0 and its database is a specification at level 1. These data are viewed or acquired through EFs of interest to the modeler. In *data-rich* environments, such data are abundant from prior experimentation or can easily be obtained from measurements. In contrast, *data-poor* environments offer meagre amounts of historical data or low-quality data (whose representativeness of the system of interest is questionable). The modeling process can direct the acquisition of data to those areas that have the highest impact on the IUs of the M&S.

In its most general guise, a *model* is a system specification at any of the levels of the Hierarchy. However, in the traditional context of M&S, the system specification is usually done at levels 3 and 4. Thus the most common concept of a simulation model is that it is a set of instructions, rules, equations, or constraints for generating input/output (I/O) behavior. In other words, we write a model with a state transition and output generation mechanisms (level 3) to accept input trajectories and generate output trajectories depending on its initial state setting. Such models form the basic components in more complex models that are constructed by coupling them together to form a level 4 specification. The definition of model in terms of system specifications has the advantages that it has a sound mathematical foundation and it has a definite semantics that everyone can understand in unambiguous fashion.

4. Intended uses and experimental frames

Figure 2 shows the application of the framework to V&V for M&S for SoS. Given the importance of the EF concept to V&V, we review it in more depth based on detailed developments found in Zeigler[16] and Traoré and Muzy.[23] Roughly, an EF is a specification of the conditions under which the system is observed or experimented with. As such EFs are the operational formulation of the IUs that motivate a modeling and simulation project. Many EFs can be formulated for the same system (both source system and model) and the same EF may apply to many systems. There are two equally valid views of an EF. In the first **data storage view**, a frame is a definition of the type of data elements that will be stored in the database to be tagged with the frame for later extraction. In the second, **data acquisition view**, a frame is a system that interacts with the SoS to obtain the data of interest under the specified conditions that are part of the frame. In this view, the frame is characterized by its implementation as a measurement system or observer. In this implementation, a frame typically has three types of components: a *generator* that generates input segments to the system; an *acceptor* that monitors an experiment to see if the desired experimental conditions are met; and a *transducer* that observes and analyzes the system output segments.

Balci[24] emphasized the role of ‘‘intended uses’’ in directing the construction of an M&S application and

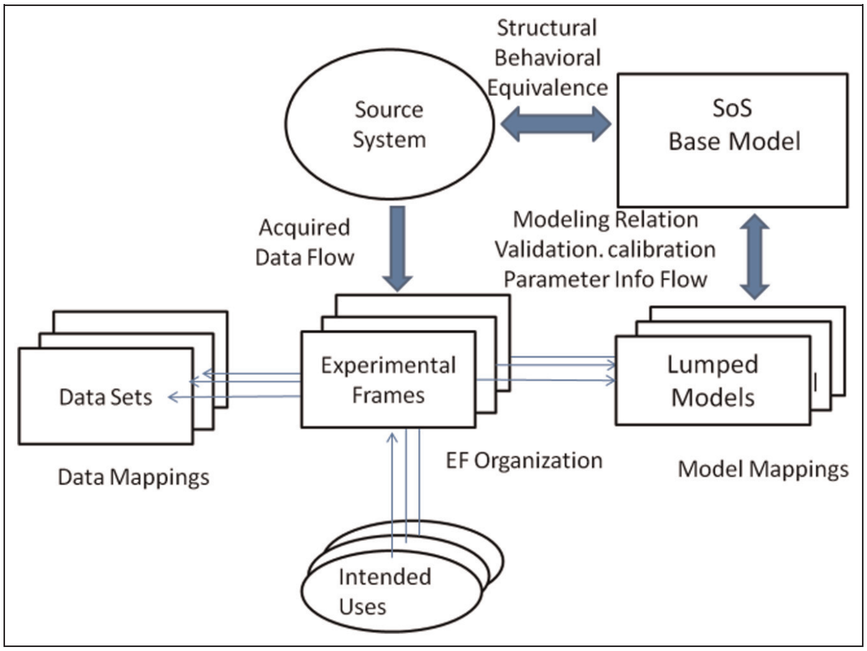

Figure 2. Architecture for SoS V&V based on M&S framework. (SoS: Systems of Systems)

consequently how it should be subject to V&V. He provided by way of illustration a hierarchy of intended uses for a particular M&S application but did not attempt a general formulation of such a hierarchy. Recently, an organization with a heavy reliance on M&S for SoS, the US Missile Defense Agency, is establishing a **standard IU specification** that contains a comprehensive listing of specifics in relation to the analysis problem that the model is intended to address.[15] Indeed this specification significantly expands the set of elements that characterize the objectives of the user. The fundamental elements of an IU include pertinent analyst tasks, model inputs and outputs, experimental designs, calibration methods and data, test objectives and concepts of operations (CONOPS). In addition the specification requires characterization of Key Attributes, i.e. aspects and values that identified stakeholders and developers agree on. These attributes include: Focus (from narrow consideration of a component, such as a specific radar, to the broad scope of end-to-end SoS evaluations), Simulation Type (Constructive, Virtual, or Live), Fidelity, Uncertainty Quantification, Interoperability, Level of detail and relation to operator training or exercise experience.

Our formulation based on such a specification is that once the IU is known, suitable EFs can be developed to accommodate it. Such frames translate the IU elements into more precise experimentation conditions for the SoS base model and its various abstractions and aggregated models called *Lumped Models*. A model developed for an application is expected to be valid in each frame associated with the IU specification that formalizes that application. An IU specifies a focus, fidelity, and a level of detail to support the problems and tasks it concerns. Different foci, fidelities, and levels of details may both require and allow different models that exploit these factors to enable optimal set up and run time attributes. The basic concept in Figure 2 is that IUs act as keys to all data and models that have been acquired and developed thus far. In the storage process, a new data set or model is linked to the IU that motivated its development. In retrieval, given an application of interest to the user, the system supports formulating a representative IU and finding the closest IU matching the newly formulated IU. If the user is unsatisfied with the match, or wishes to explore further, the system supports synthesizing a composite IU using available lattice-like operations (upper and lower bounds, decomposition, etc.).

4.1 Review of validity and partial order relations

The basic modeling relation, *validity*, refers to the relation between a *model*, a *system* and an *EF*. Validity is often thought of as the degree to which a model faithfully represents its system counterpart. However, it makes much more practical sense to require that the model faithfully captures the system behavior only to the extent demanded by the objectives of the simulation study. In our formulation, the concept of validity answers the question of whether it is impossible to distinguish the model and system *in the EF of interest*. The most basic concept, *replicative validity*, is affirmed if, for all the experiments possible within the EF, the behavior of the model and system agree within acceptable tolerance. Thus replicative validity requires that the model and system agree at the I/O relation level 1 of the system specification hierarchy.

Stronger forms of validity are *predictive validity* and *structural validity*. In predictive validity we require not only replicative validity, but also the ability to predict as yet unseen system behavior. To do this the model needs to be set in a state corresponding to that of the system. Thus predictive validity requires agreement at the next level of the system hierarchy, that of the I/O function level 2. Finally, structural validity requires agreement at level 3 (state transition) or higher (coupled component). This means that the model not only is capable of replicating the data observed from the system but also mimics in step-by-step, component-by-component fashion, the way that the system does its transitions.

The term *accuracy* is often used in place of validity. Another term, *fidelity*[25] is often used for a combination of both validity and detail. Thus, a high fidelity model may refer to a model that is both high in detail and in validity (in some understood EF). However, when used this way there may be a tacit assumption that high detail alone is needed for high fidelity, as if validity is a necessary consequence of high detail. In fact, it is possible to have a very detailed model that is nevertheless very much in error, simply because some of the highly resolved components function in a different manner than their real system counterparts.

Besides the two fundamental relationships, there are others that are important for understanding modeling and simulation work. These relations have to do with the interplay and orderings of models and EFs. The inescapable fact about modeling is that it is severely constrained by complexity limitations. Complexity is measured typically on resource usage in time and space relative to a particular simulator, or class of simulators. However, properties intrinsic to the model are often strongly correlated with complexity independently of the underlying simulator. Successful modeling can then be seen as *valid simplification*. We need to *simplify*, or reduce the complexity, to enable models to be executed on resource-limited simulators. But the simplified model must also be *valid*, at some level, and within some EF of interest. As in Figure 3, there is always a pair of models involved, call them the *base* and *lumped* models. Here, the base model is typically ''more capable'' and requires more resources for interpretation than the lumped model. By the term ''more capable,'' we mean that the base model is valid within a larger set of EFs (with respect to a real system) than the lumped model. However, the important point is that within a particular frame of interest the lumped model might be just as valid as the base model. The concept of morphism, introduced above, affords criteria for judging the equivalence of base and lumped models with respect to an EF.

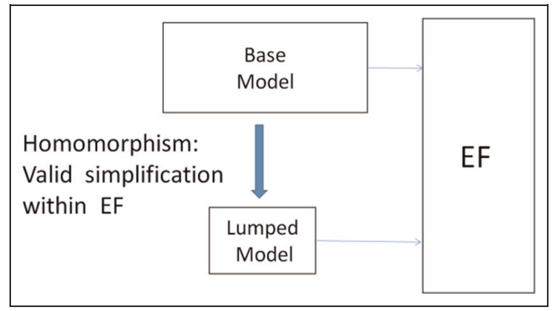

Figure 3. Validity of base and lumped models in EF. (EF: experimental frame)

Table 3 gives the fundamental relations that underly the organization of models and data in the architecture of Figure 2. In this conception, the data architecture for SoS V&V contains a repository of models and EFs keyed by IUs. Focusing on EFs, it is critical to have an ability to ask whether there are any frames that meet our current objectives and whether there are models that can work within such frames. The relation that determines if a frame can logically be applied to a model is called *applicability* and its converse is called *accommodation*. Notice that validity of a model in a particular EF requires, as a precondition, that the model accommodates the frame. The degree to which one EF is more restrictive in its conditions than another is formulated in the *derivability* relation. A more restrictive frame leaves less room for experimentation or observation than one from which it is derivable. So, as illustrated in Figure 4, it is easier to find a model that is valid in a restrictive frame for a given system. It has been shown that applicability may be reduced to derivability. Briefly, to see this, define the *scope* frame of the model to represent the most relaxed conditions under which it can be experimented with (this is clearly a characteristic of the model). Then a frame is applicable to a model, if it is derivable from the scope frame of the model.

5. Systems morphisms to integrate simulation and formal verification

Extending the work of Zeigler[2,16], Traoré and Muzy[23] formalized EFs as a basis for specifying the context in which simulation models are built with the goal of integrating formal methods for verification of a model's consistency, composability, reuse, and validity using model checkers and theorem provers. Foures et al.[26] provide a metric-based method which aims to guide EF and/or model definition to assist finding the right model and the right EF for a

Table 3. Definition of fundamental partial order relations.

Relationship	Definition
Experimental frame applies to a model (or 'is applicable to')	The conditions on experimentation required by the frame can be enforced in the model
Model accommodates experimental frame	Frame is applicable to the model
Experimental frame 1 is derivable from experimental frame 2	Any model that accommodates experimental frame 2 also accommodates experimental frame 1

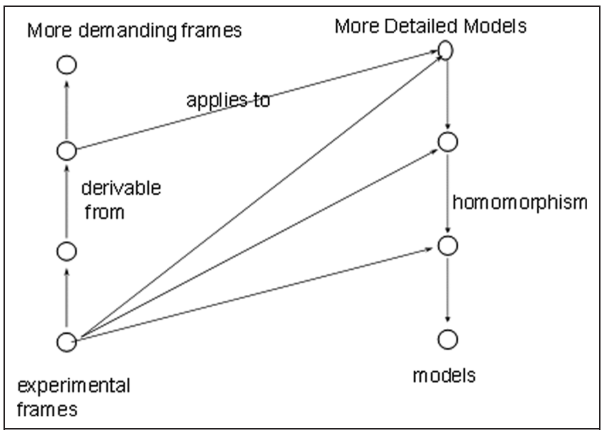

Figure 4. Fundamental ordering relations for data architecture.

given intended use. Zeigler and Nutaro[27] discuss the use of morphisms that builds upon recent extensive work on verification combining DEVS and model checking for hybrid systems.[14,28,29] The mathematical concepts within the DEVS formalism encompass a broad class of systems that includes multi-agent discrete event components combined with continuous components such as timed automata, hybrid automata, and systems described by constrained differential equations. System morphisms can map a model expressed in a formalism suitable for analysis (e.g. timed automata or hybrid automata) into the DEVS formalism for the purpose of simulation. Conversely, it is also possible to go from DEVS to a formalism suitable for analysis for the purposes of model checking, symbolic extraction of test cases, and reachability, among other analysis tasks.

5.1 Model abstraction in formal verification

Humphrey[30] explored the use of Linear Temporal Logic, the SPIN model checker, and the modeling language PROMELA[30] for high-level design and verification in Unmanned Autonomous Vehicles (UAV) related applications. She reported some success while suggesting limitations and needed extensions. Table 4 shows three UAV related cases she discussed.

In each case, the focus of model is shown along with a simplifying assumption. Because they are oriented to verification, model checking tools tend to lack many functions that exist in DEVS environments and require abstractions that fit the tools' operation. This forces an abstraction of the real system that on the one hand enables the modeler to better understand the model, and on the other hand entails numerous assumptions to enable the model checker to verify the focal requirement. Despite these drastic simplifications, state space explosion prevents employing more than a handful of UAVs and sensors.

As discussed earlier, several DEVS methodologies have been developed which incorporate non-DEVS verification methods.[13,14,21,28] These methodologies attempt to employ DEVS to enable loosening the simplifying assumptions typically made by non-simulation models. In another variation, functional and temporal properties of Timed Stream Petri Net models are checked using exhaustive verification or DEVS-based simulation.

The combination of simulation and formal verification gives a much more powerful capability to test designs than can be achieved with either alone. In a design process that incorporates both types of analysis, verification models can be used to obtain absolute answers concerning system behavior under idealized conditions. Failures in this verification stage should indicate a need to find and correct fundamental flaws in the system design. On the other hand, if a successfully verified model can be formally extended into a simulation model for which the verification model is a homomorphic simplification, the simulation model might retain the properties that were verified with the simpler model, and then can be used to explore scenarios that are necessarily outside the scope of formal verification.

5.2 Integration of verification and simulation

The representation of this idea within the organization of relations just described is shown in Figure 5. Here the lumped model represents an analysis model that we are seeking to verify for a set of requirements and assumptions represented by EF_{Assum}. The base model represents a simulation model that has more of the structure and behavior representative of the real-world system and

Table 4. Example applications of model checking to a UAV SoS.

Model #1 A centralized UAV controller that coordinates the actions of multiple UAVs performing a monitoring task.
Focus of model checking Assuring that all sensors are eventually visited.
Sample simplifying assumptions Communication between UAVs and sensors can only occur when in the same location and is error free.
Model #2 A leader election protocol for a decentralized system of unattended ground sensors sending estimates of an intruder's position to a UAV.
Focus of model checking At least one leader exists at every time step.
Sample simplifying assumptions The sensors all use sampling epochs of the same length enabling a single time step for time advance.
Model #3 Verification of high level UAV mission plans for a scenario in which multiple UAVs must be used to safely escort an asset across a road network.
Focus of model checking The path travelled by the asset is safe, i.e. all road segments in the path have been scanned by UAV.
Sample simplifying assumptions UAVs and asset were assumed to travel at the same speed.

UAV: Unmanned Automated Vehicle

accommodates a ''larger'' frame, EF_{Scope}. The fact that the latter is ''larger,'' i.e. more inclusive, than the latter is captured by the derivability relation. Then, if we can demonstrate a homomorphism from the base to the lumped model EF_{Assum} one might suspect that any property proved to hold for the lumped model would also hold for the base model in that same frame. Unfortunately, as we will show, the literature shows this preservation of properties is not a necessary consequence of such a homomorphism. Moreover, it is more questionable whether the property continues to hold in the larger frame, EF_{Scope}.

To illustrate, let EF_{Assum} contain the first simplifying assumption of Table 4''Communication between UAVs and sensors can only occur when in the same location and is error free'' and suppose the lumped model of the UAV system has been shown to meet the requirements that ''all sensors are eventually visited.'' Now suppose that in addition, we construct a base model that allows for realistic communication (less spatially constrained and error-prone). Under what circumstances would it be possible to show that all sensors are still eventually visited? Presumably, the base model would have to be extended in such a way as to have the ''same'' structure and behavior as the lumped model, which would require that a strong morphism hold between them. It would be natural to first consider that the morphism holds when the simplifying assumption is made (i.e. within EF_{Assum}) and then whether it still holds when the condition is relaxed (i.e. within EF_{Scope}).

A full framing of the problem for SoS is obtained by returning to the architecture of Figure 2 where for the UAV multiagent SoS in Table 4, we have at least 3 EFs (each representing different simplifying conditions) and 3 lumped models (each representing a verified analyzable model). What can we say about a base (simulation) model that attempts to be compatible with each of the simplified models and therefore retain the desired properties they

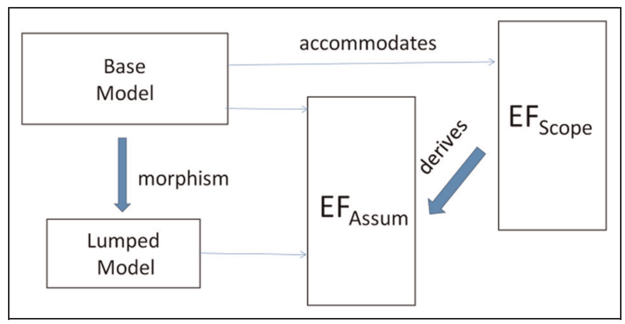

Figure 5. Relation between verification and simulation. (EF: experimental frame)

satisfy? The fundamental ordering relations of Figure 4 and the system theory hierarchy of specifications and morphisms give us a means to frame the problem and develop a methodology to approach its solution. In the last section, we address the question of what kind of morphic relation is needed between base and lumped models in order for proven properties of the lumped model to also hold for the base model.

6. Morphisms and preservation of properties

What kind of morphic relation between base and lumped models justifies inferring that a base model has a property just established for a lumped model? Consider base and lumped models with a homomorphism holding in an EF as in Figure 4. Assume the lumped model has property P. Is it true that the base model must have property P as well? Consider Figure 6, where for convenience we will use S (or System) to refer to the base model and M (or Model) to refer to the lumped model. We call *upward* preservation, or structural inference, the inference: ''if M has P

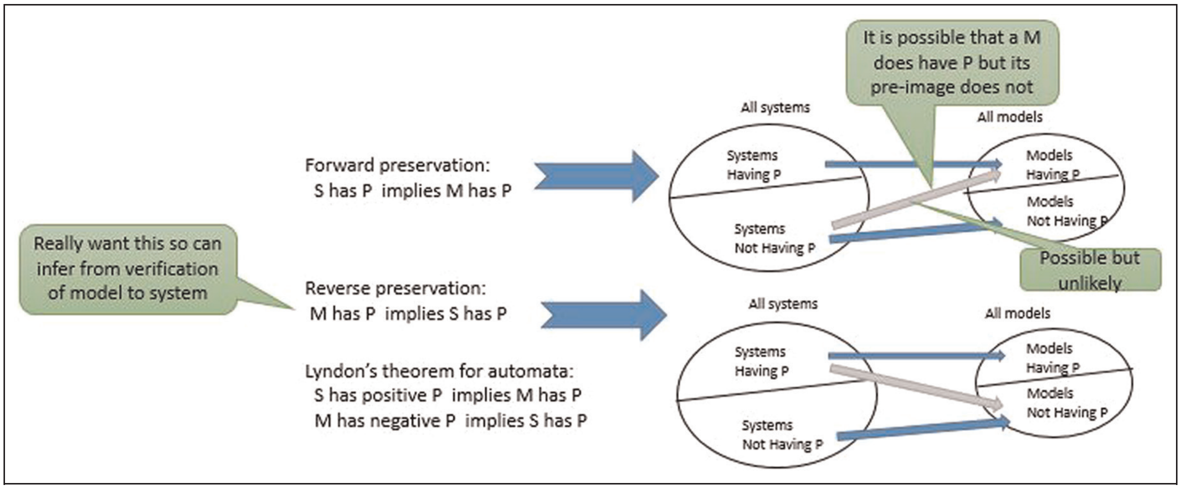

Figure 6. Downward and upward preservation of properties

then S has P'' and as emphasized, it represents the kind of preservation we are focusing on here. However, *downward* preservation, where a property P is inherited from S to M is of interest as well. The problem of downward preservation was raised by Foo[31] who provided sufficient conditions for inheritance of stability properties for continuous systems and pointed out that downward preservation of P implies upward preservation of –P, the negation of P. Unfortunately, properties of interest seem not to be expressible in negative form. Indeed, Sierocki[32] clarified the situation by applying a well-known universal algebraic formulation of the logician Lyndon to finite automata. Informally, a positive property is one expressible in first order logic without use of negation. Conversely, a negative property is one that requires negation to express it. Sierocki enumerates a number of properties of automata that are of interest (e.g. relating to reachability, connectedness, and reversibility) and are positive by direct statement. Applying Lyndon's theorem, Sierocki shows that upward inheritance of positive properties holds for the usual homomorphism of automata. Moreover, downward inheritance holds for negative properties.

The general situation is unsettled. Saadawi and Wainer[28] shows that some properties transfer upwards from Safety Timed Automata models verified in UPPAAL to real-time advance DEVS models under a strong form of bi-simulation similar to isomorphism. Zeigler et al.[8] provide examples of morphisms and properties where it is both possible and not possible to make structural inferences. Given this situation, one direction for research is to look at more special cases. Another is to formulate the problem within a probabilistic, rather than logical, framework. It turns out we can get a more robust approach to making structural inferences as well as gaining more insight into the role of morphisms in the process.

6.1 Probabilistic perspective: Bayesian reasoning

The thin arrows in Figure 6 denote paring of Systems and Models corresponding to morphisms. Consider that in the situation for downward preservation, with the blue arrows, that Systems having P map to Models having P. The problem is caused by the possibility, shown as a gray arrow, that that Systems not having P also map to Models having P. In other words, a morphism may be ***deceptive*** even though it preserves behavior and/or structure at some level of the hierarchy. Certainly, if we set up a random pairing of Ss and Ms where Ss with P always map to Ms with P, we would not expect that Ss without P also map to Ms without P. However, there may be constraints on the Ss, Ms, and mappings that pull it away from pure randomness and make it less likely that a deceptive cross-mapping exists. Indeed, we can show that under conditions consistent with downward preservation of hard-to-prove properties, the posterior probability that S has P given that M has P is greater than the prior probability that S has P. In other words, demonstrating a homomorphism, or more generally a strong morphism, can **significantly increase** the likelihood that a lumped model property is truly reflective of the more complex base model. The set-up and proof proceed using the reasoning underlying Bayes' theorem and are provided in the Appendix.

7. Historical trace of V&V streams

After several decades of development, there are still a wide variety of V&V terms, concepts, products, processes, tools and techniques.[33] This variety derives from different streams of development and motivates the following effort to place the framework for more robust V&V methodology presented here in historical perspective.

Table 5. Conceptual definitions of objects and modeling and simulation framework (MSF) equivalents.

Conceptual definition of object	MSF formalization
A **simuland** is the real-world system of interest. It is the object, process, or phenomenon to be simulated	Real-world system is a source of data and can be represented by a system specification at a behavioral level
A model is a representation of a simuland, broadly grouped into conceptual and executable types.	Model is a set of rules for generating behavior and can be represented by a system specification at a structural level. A modeling formalism, e.g. DEVS, enables conceptual specification and is mapped to a simulation language for execution by a simulator.
Simulation is the process of executing a model over time	A simulator is a system capable of generating the behavior of a model; simulators come in classes corresponding to formalisms, e.g. an abstract DEVS simulator provides the rules for implementable simulators of DEVS models
The **results** of simulation are the output produced by a model during a simulation	The behavior of a model generated by a simulator constitutes a specification at the behavior level.

DEV: Discrete Event System Specification

Table 6. Conceptual definitions of activities and modeling and simulation framework (MSF) equivalents.

Conceptual definition of activity	MSF formalization
Verification is the process of determining if an implemented model is consistent with its specification	There is a relation, called simulation correctness, between models and simulators. Verification is the process of proving this correctness for a simulator generating the behavior of the model. When this is done for a formalism, it certifies a simulator as correct for any model of the associated class.
Validation is the process of determining if a model behaves with satisfactory accuracy consistent with the study objectives within its domain of applicability to the simuland it represents.	There is a relation, called *validity in a frame*, between models and real systems within an experimental frame. Validation is the process of establishing that the behaviors of the model and real system agree in the frame in question. The frame can capture the intended objectives (extended to intended uses), domain of applicability, and accuracy requirements.
Abstraction is the omission or reduction of detail not considered necessary in a model.	Abstraction is the process of constructing a lumped model from a base model intended to be valid for the real system in a given experimental frame.

MSF: modeling and simulation framework

7.1 Informal V&V concepts and processes

The work of Balci[34–36] Balci and Sargent,[38] and Kleijnen[39] on V&V provided a starting point for many future efforts in the defense[40] Pace[41,42] and other M&S communities.[42] In this branch of the literature terminology is defined conceptually but not in the mathematically precise manner of the MSF presented here. A rough equivalence between objects in this literature and entities of the framework can be established in Table 5.

Processes involved in M&S activities are likewise defined informally in this stream of literature. As the MSF defines its entities as mathematical systems, it can define such processes as mathematical relations. A rough equivalence is given in Table 6.

7.2 Theory-based and model-driven developments

Research and development on V&V based on the conceptual definitions has been focused on process rather than fundamentals of theory to supply solid foundations for such processes. On the more theoretical side, the use of simplified (reduced order, more abstract) meta-models in Verification, Validation and Acreditation (VV&A) was discussed in Caughlin.[3] Weisel et al.[43–45] employed the idea of bi-simulation to ensure fundamental ordering relations and to evaluate them for existing systems. Multi-resolution modeling, in which a family of models at several levels of resolution is maintained[22] has been employed in distributed simulations of military systems (see e.g., Weisel et al.[46]) Baohong[47] formalized such multi-resolution model families employing a DEVS representation. He provided closure under coupling for such families which provides a foundation for research into consistency between levels of resolution, related to the problem of model simplification discussed here. Vangheluwe[5] summarized the theory of M&S and emphasizes DEVS as a common basis for multi-formalism modeling using formalism transformation graphs. He emphasizes the notion that much of V&V can be done virtually by model-checking, simulation, and optimization prior to any actual ''bending of metal,'' at least for systems design. This ''doing it right the first time'' leads to significant cost savings and improved quality. The mathematical foundations for model-based systems engineering originate with Wymore[20] (see also Mittal and Risco-Martín J[21] and Davis and Hillestad[22]). Model transformation as the basis for V&V as well as for simulation system development has been evaluated in great detail in the recent work at TU Delft.[48,49] Cetinkaya et al.[50] incorporated MDD4MS, a model drive development framework for modeling and simulation into the broader framework of the DEVS Unified Process.[21,51,52]

7.3 Generic methodology processes and best practice guides

The wide variety of V&V terms, concepts, and processes, tools or techniques has a negative consequence in the same way that lack of standardization inhibits other areas of M&S. This was the motivation behind the recent development of the Generic Methodology for Verification and Validation (GM-VV) within the Simulation Interoperability Standards Organization (SISO). Paraphrasing Rosa et al.,[33] as a generic methodology, the GM-VV comprises an abstract technical framework that consists of three parts: 1) the conceptual framework provides unifying terminology, concepts and principles, 2) the implementation framework translates these concepts into a building blocks for the development of concrete and consistent V&V solutions, and 3) the tailoring framework utilizes these building blocks to develop and cost-efficiently apply such V&V application instantiations. GM-VV can be tailored to more concrete guides and recommendations. The US Modeling and Simulation Coordination Office (US MSCO) compiled the VV&A Recommended Practice Guide (RPG) to facilitate the application of its directives and guidelines and to promote the effective application of VV&A.[53,54] The High Level Architecture (HLA) standard entails the Federation Development Process (FEDEP) which provides definitions and descriptions of the steps, tasks, and activities that are standardized for the development of a federation. An extension of the standard was developed to provide an overlay to the FEDEP that adds the required phases and activities for verification, validation, and accreditation at the same level of detail.[39] This complies with Ören's[55] broader concept of Quality Assurance where any element or activity involved in the M&S enterprise can be subject to the kinds of consistency checking, evaluation and comparison activities, typically associated with verification and validation of models.

The GM-VV acceptance goal is to develop a recommendation that shows why an M&S asset is, or not, acceptable for the stakeholder. Intended use is a primary factor in such acceptability criteria. However, intended use is not defined in GM-VV nor in Roza's thesis[25] to which it refers for background. In relation to the MSF definition of EF, GM-VV defines a V&V EF as a set of experiments, tests and conditions used to observe and experiment with the M&S system to obtain V&V results which are the collection of data items produced by applying the frame. The implementation framework of the GM-VV specifies the use of an *argumentation structure* to capture the derivation of the V&V EF specification from the criteria for acceptability of the M&S system which themselves derive from the goal of V&V.

8. Conclusions and recommendations

We have presented a framework for V&V of simulation models of SoS based on the hierarchy of system specifications and associated theory of M&S. The framework reintroduces foundational theory for current literature in V&V that is process oriented. This theory influenced the development of a GM-VV, a generic methodology for V&V. The extended framework includes the more elaborated formulation of Intended Use, a characterization of modeler objectives that enhances the conceptual underpinning for EF theory. Such enhancement significantly contributes to the clarity with which M&S methodology can be executed. Research is needed to extend the derivability and other relations defined for frames and models to support intended use constructs. The GM-VV argumentation structure described above could benefit from such support.

We also considered issues involving the use of simplified models as valid abstractions for real systems in EFs. Here, we considered the kind of morphic relations that are needed between base and lumped models that justify inferring that a base model actually has a property of interest that has been proven for a lumped model. As a concrete application, we considered the case of integrating model checking with simulation where there is a need to have confidence that the properties proved for idealized models also hold in more realistic simulation models strongly related to them. We used both logical and Bayesian perspectives to clarify the situation. This approach suggests that similar hybrid research should be undertaken to elaborate the framework presented here for more robust validation and verification of SoS simulation models.

Finally, it is important to consider V&V within the larger context of model engineering that has been defined by Zhang et al.[56] as the formulation of theories, methods, technologies, standards and tools relevant to a systematic, standardized, quantifiable engineering methodology that guarantees the credibility of the full lifecycle of a model with the minimum cost. V&V is an important component, but its full treatment needs to be considered in interaction with the other components of the development lifecycle.

Acknowledgment

The authors wish to thank the reviewers of this paper for their helpful comments in clarifying its contribution to the advance of theory in support of the process of V&V.

Declaration of Conflicting Interest

The authors declare that there is no conflict of interest.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors. This manuscript has been co-authored by an employee of UT-Battelle, LLC, under Contract DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes

Notes

1. Models are also conceived as idealizations which are derived independently of a more complex model in mind. Nevertheless to test such models for validity it is helpful to formulate a more complex model providing a context in which their idealizations can be considered. We illustrate this approach after reviewing the basic framework.
2. The full definition of morphism relationships involving aggregation, lumping and other simplification processes is discussed in Zeigler[2] and Zeigler et al.[8] A more informal discussion that offers important insights is discussed by Davis and Bigelow.[22]
3. As technology advances, such constraints may continue to be loosened. Nevertheless, fundamental limitations to computation are well established in computer science. Moreover, we conceive of human comprehension of models as involving mental simulation and therefore cognitive computational constraints are also included in such limitations.
4. Human understanding of models formulated as a class of cognitive simulators is included, as per the footnote.
5. This definition is a synthesis of various definitions in the literature that separately relate the model to a real system, the purposes of model construction, the domain of applicability and the accuracy required. For example, Balci[24] defines validation as the assessment of behavioral or representational accuracy and then later conditions accuracy on intended use. Our intent is to best represent the conceptual literature for the purposes of relating it to the MSF.
6. The approach we take here differs from the formulation of homomorphisms of probabilistic systems (Aggarwal[1]) also called stochastic systems (Castro and Kofman, 2010). Such morphisms are predicated on the formulation of systems as stochastic, whereas the formulation here applies to all systems considered as points in a space where the probability of having a property has been defined. The relation between the two levels of consideration is also a possible topic or research.

References

1. Aggarwal S. *Ergodic Machines—Probabilistic and Approximate Homomorphic Simplifications*. PhD Thesis, The University of Michigan, Ann Arbor, Michigan, USA, 1975.
2. Zeigler BP. *Theory of modeling and simulation*. 1st ed. New York: John Wiley and Sons, 1976.
3. Caughlin D. Verification, validation, and accreditation (VV&A) of models and simulations through reduced order metamodels. In: *Proceedings of the 27th conference on winter simulation*. IEEE Computer Society, 1995.
4. Lee K and Fishwick PA. Dynamic model abstraction. In: *Proceedings of 1996 winter simulation conference*, San Diego, CA, December 1996, pp. 764–771.
5. Vangheluwe H. Foundations of modelling and simulation of complex systems. In: *ECEASST* 10, 2008.
6. Baier C and Joost-Pieter K. *Principles of model checking*. Cambridge, MA: MIT Press, 2008.
7. Jamshidi M. *Systems of systems—innovations for the 21st century*. New York: John Wiley and Sons, 2008.
8. Zeigler BP, Praehofer H and Kim TG. *Theory of modeling and simulation*. 2nd ed. Boston, MA: Academic Press, 2000.
9. Shaffer AR. The value of modeling and simulation for the department of defense. *M&S Journal* 2012; (Fall): 2–3.
10. Mittal S and Risco-Martín JL. Model-driven systems engineering for netcentric system of systems with DEVS unified process. In: Pasupathy R, Kim S-H, Tolk A, Hill R and ME Kuhl ME (eds) *Proceedings of the 2013 winter simulation conference*, 2013.
11. Nutaro JJ. *Building software for simulation: Theory and algorithms with applications in C++*. New York: John Wiley and Sons, 2011.
12. Zeigler BP, Song H., Kim T and Praehofer H. DEVS framework for modeling, simulation, analysis, and design of hybrid systems. In: Antsaklis P, Kohn W, Nerode A and Sastry S (eds) *Proceedings of hybrid systems II (Lecture Notes in Computer Science, vol. 999)*. Berlin: Springer-Verlag, 1995, pp. 529–551.
13. Zeigler BP and Sarjoughian HS. *Guide to modeling and simulation of systems of systems*. New York: Springer, 2012.
14. Denil J. *Design, verification and deployment of software intensive systems: A multi-paradigm modelling approach*. PhD Dissertation, University of Antwerp, 2013.
15. Piersall CH and Grange FE. The necessity of intended use specification for successful modeling and simulation of a

systems-of-systems (Crosstalk Online). http://m.crosstalkonline.org/issues/15/130/

16. Zeigler BP. *Multifaceted modelling and discrete event simulation*. New York: Academic Press, 1984.
17. Zeigler BP. Structuring the organization of partial models. *Int J Gen Syst* 1978: 4: 81–88.
18. Ören TI and Zeigler BP. System theoretic foundations of modeling and simulation: A historic perspective and the legacy of A. Wayne Wymore. *Simulation* 2012; 88: 1033–1046.
19. Wymore AW. *A mathematical theory of systems engineering: The elements*. New York: John Wiley and Sons, 1967.
20. Wymore AW. *Model-based systems engineering: An introduction to the mathematical theory of discrete systems and to the tricotyledon theory of system design*. Boca Raton, FL: CRC Press, 1993.
21. Mittal S and Risco-Martín JL. *DEVS net-centric system of systems engineering with DEVS unified process (CRC–Taylor & Francis Series on System of Systems Engineering)*. New York: Taylor & Francis, 2012.
22. Davis PK and Hillestad RJ. *Experiments in multiresolution modeling*. RAND Report MR1004, RAND Corp., 1998.
23. Traoré MK and Muzy A. Capturing the dual relationship between simulation models and their context. *Sim Model Practice Theory* 2006; 14: 126–142.
24. Balci O. Golden rules of verification, validation, testing, and certification of modeling and simulation applications. *M&S Mag* 2010; n4(Oct): 1–7.
25. Roza ZC. *Simulation fidelity theory and practice: a unified approach to defining, specifying and measuring realism of simulations*. Delft, The Netherlands: Delft University Press Science, 2004.
26. Foures D, Albert V and Nkesta A. Formal compatibility of experimental frame concept and FD-DEVS model. In: *Proceedings of 9th international conference of modeling, optimization and simulation—MOSIM'12*, 2012.
27. Zeigler BP and Nutaro JJ. Combining DEVS and model-Checking: Using systems morphisms for integrating simulation and analysis in model engineering. In: *EMSS* 2014, Bordeaux, France, 2014.
28. Saadawi H and Wainer G. Principles of discrete event system specification model verification. *Simulation* 2013; 89(1): 41–67.
29. Saadawi H, Wainer G and Moallemi M. Principles of models verification for real-time embedded applications. In: Popovici K and Mosterman P (eds) *Real-time simulation technologies: principles, methodologies, and applications*. Boca Raton, FL: CRC Press, 2012.
30. Humphrey LR. Model checking for verification in UAV cooperative control applications recent advances in research on unmanned aerial vehicles. *Lecture Notes in Control and Information Sciences* 2013; 444: 69–117.
31. Foo N. Stability preservation under homomorphisms. *IEEE Trans SMC* 1977; 7: 750–754.
32. Sierocki I. 1986. A note on structural inference in systems theory. *Int J Gen Syst* 1986, 13: 17–22. DOI: 10.1080/03081078608934951.
33. Roza M, Voogd J and Sebalj D. The generic methodology for verification and validation to support acceptance of models, simulations and data. *JDMS* 2013; 10: 347–365.
34. Balci O. Verification, validation, and accreditation. In: *Proceedings of the 30th conference on winter simulation*. IEEE Computer Society Press, 1998.
35. Balci O. A life cycle for modeling and simulation. *Simulation* 2012; 88: 870–883.
36. Balci O and Ormsby W. Well-defined intended uses: an explicit requirement for accreditation of modeling and simulation applications. In: *Proceedings of the 2000 winter simulation conference*. IEEE Computer Society Press, 1998
37. Balci O and Sargent RG. A methodology for cost–risk analysis in the statistical validation of simulation models. *Commun ACM* 1981; 24: 190–197.
38. Kleijnen JPC. Verification and validation of simulation models. *Eur J Op Res* 1995, 82: 145–162.
39. Tolk A. Verification and validation. *Engineering principles of combat modeling and distributed simulation*. Hoboken, NJ: John Wiley & Sons, 2012, pp. 263–294.
40. Pace DK. 2004. Modeling and simulation verification and validation challenges. *John Hopkins APL Tech Digest* 2004; 25(2): 163–172.
41. Pace DK. 2012 Comprehensive consideration of uncertainty in simulation use. *JDMS* 2013; 10: 367–380. DOI: 10.1177/1548512912455471.
42. Obaidat MS and Boudriga N. *Fundamentals of performance evaluation of computer and telecommunications systems*. New York: John Wiley & Sons, 2010.
43. Weisel EW. Towards a foundational theory for validation of models and simulations. In: *Proceedings of the spring 2011 simulation interoperability workshop*, 11S–SIW–074. Orlando, FL: Simulation Interoperability Standards Organization (SISO), 2011. http://www.sisostds.org/.
44. Weisel EW. Decision theoretic approach to defining use for computer simulation. In: *Proceedings of the 2012 autumn simulation conference*. San Diego, CA: Society for Modeling and Simulation International (SCS), 2012.
45. Weisel EW, Petty MP and Mielke RR. Validity of models and classes of models in semantic composability. In: *Proceedings of the fall 2003 simulation interoperability workshop*, 03F–SIW–073. Orlando, FL: Simulation Interoperability Standards Organization (SISO), 2003. http://www.sisostds.org/.
46. Reynolds RA, Iskenderian H and Ouzts SO. Using multiple representations and resolutions to compose simulated METOC environments. In: *Proceedings of 2002 spring simulation interoperability workshop*, 2002.
47. Baohong L. A formal description specification for multi-resolution modeling based on DEVS formalism and its applications. *JDMS* 2007; 4: 229–251.
48. Cetinkaya D. *Model driven development of simulation models: defining and transforming conceptual models into simulation models by using metamodels and model transformation*. PhD Thesis, Technical University Delft, The Netherlands, 2013.
49. Cetinkaya D, Verbraeck A and Seck MD. MDD4MS: A model driven development framework for modeling and

simulation. In: *Proceedings of the summer computer simulation conference*. SCS Press, 2011, pp. 113–121
50. Cetinkaya D, Mittal S, Verbraeck A and Seck MD. Model driven engineering and its application in modeling and simulation. In Mittal S and Martin JLR (eds) *Netcentric system of systems engineering with DEVS unified process*. Boca Raton, FL: CRC Press, 2013, pp. 221–248.
51. Zeigler BP and Moon Y. DEVS representation and aggregation of spatially distributed systems: speed versus error tradeoffs. *Trans SCS*, 1996; 13: 179–190.
52. MSCO 2014 Verification, validation and accreditation: Recommended Practices Guide. Available at: http://msco.mil/VVA_RPG_RandR.html.
53. Petty MD. Verification and validation. In: Sokolowski JA and Banks CM (eds) *Principles of modeling and simulation: a multidisciplinary approach*. Hoboken, NJ: John Wiley & Sons, 2009.
54. Ören TI. Artificial intelligence in quality assurance of simulation studies. In Elzas MS, Ören TI and Zeigler BP (eds) *Modelling and simulation methodology in the artificial intelligence era*. Amsterdam: North-Holland, 1986, pp. 267–278
55. Zhang L, Shen YW, Zhang XS, Song X, Tao F and Liu Y. The model engineering for complex system simulation. In: *Proceedings of the 26th European modeling and simulation symposium (simulation in industry)*, Bordeaux, France, 10–12 September 2014.
56. Castro, R. and E Kofman, G Wainer (2010) A formal framework for stochastic discrete event system specification modeling and simulation, *Simulation* 86 (10), 587–611

Appendix: Proof of the increase in probability that a system has a given property

We demonstrate that a homomorphism, or more generally a strong morphism, can significantly increase the likelihood that a lumped model property is truly reflective of the more complex base model. Using the reasoning underlying Bayes' theorem, we consider a model M of a system S, and define the following probabilities.

P(S), the probability that the system S has the property.

P(-S) = 1-P(S), the probability that the system S does not have the property.

P(M), the probability that model M has the property.

P(M/S), the probability that model M has the property given that system S has the property.

P(M/-S), the probability that model M has the property given that S does not have the property.

We assume that **P(M/S)** > **P(M/-S)**, that is, we assume it is more likely for the model to have the property when the system has the property than for the model to have the property when the system does not have the property. This is consistent with a morphism existing from S to M that preserves the property. It allows for the possibility that models can have properties not possessed by the system but that there a smaller chance that this will happen.

From this assumption it will follow that P(S/M)/P(S) > 1, i.e., the posterior probability that S has the property (given that its model has the property) is greater than the prior probability that S has the property.

Proof: Using the definition of conditional probability (underlying Bayes theorem) we can write:

$$P(S|M)/P(S) = P(SM)/[P(M)P(S)] = P(M|S)/P(M)$$

It follow from operations to both sides that

$$\begin{aligned} P(S|M)/P(S) &= P(M|S)/P(M) \\ &= P(M|S)/[P(M|S)P(S) + P(M|-S)P(-S)] \\ &= P(M|S)/[P(M|S)P(S) + P(M|-S)(1-P(S))] \\ &> P(M|S)/[P(M|S)P(S) + P(M|S)(1-P(S))] \\ &= 1/[P(S) + 1 - P(S)] \\ &= 1 \end{aligned}$$

To illustrate the argument, suppose P(M|S) = 0.1, P(M|S) = 0.5, and so

$$\begin{aligned} P(S|M)/P(S) &= 0.5/[0.5P(S) + 0.1(1-P(S))] \\ &= 0.5/[0.4P(S) + 0.1] \end{aligned}$$

If P(S) is small (i.e. if our prior belief that S has the property is small but not zero), then a proof that M has the property increases our confidence in the system by a factor of approximately 0.5 / 0.1 = 5. On the other hand, if P(S) is large (i.e. if we believe that S has the property) then P(S/M) / P(S) $\approx$ 1: in this case, the proof merely confirms what we already knew.

Author biographies

Bernard Zeigler is Professor Emeritus from the University of Arizona and Chief Scientist at RTSync Corp., a university spinoff. He was founding editor of the *Journal of Defense Modeling and Simulation* and recognized by Simulation Archive as a Computer Simulation Pioneer. His biography appears in Wikipedia.

James Nutaro is a member of the M&S group at Oak Ridge National Lab. He holds a B.S., M.S., and Ph.D. in Computer Engineering from the University of Arizona. He is the author of Building Software for Simulation: Theory and Algorithms, with Applications in C++.