Journal of Simulation (2013) 7, 69–82 1747-7778/13

www.palgrave-journals.com/jos/

Reference modelling in support of M&S—foundations and applications

A Tolk[1]*, SY Diallo[2], JJ Padilla[2] and H Herencia-Zapana[3]

[1]Engineering Management and Systems Engineering, Old Dominion University, Virginia, USA; [2]Virginia Modeling Analysis and Simulation Center, Old Dominion University, Virginia, USA; and [3]National Institute of Aerospace, Hampton, Virginia, USA

Whether by design or by practice, systems engineering (SE) processes are used more and more often in Modeling and Simulation (M&S). While the two disciplines are very close, there are some differences that must be taken into account in order to successfully reuse practices from one community to another. In this paper, we introduce the M&S System Development Framework (MS-SDF) that unifies SE and M&S processes. The MS-SDF comprises the SE processes of requirements capture, conceptual modelling, and verification and validation (V&V), and extends them to M&S. We use model theory as a deductive apparatus in order to develop the MS-SDF. We discuss the benefits of the MS-SDF especially in the selection between federation development and multi-model approaches and the design of composable models and simulations. Lastly, a real life application example of the framework is provided.

Journal of Simulation (2013) **7,** 69–82. doi:10.1057/jos.2013.3;

Keywords: conceptual modelling; model theory; systems engineering; verification & validation

1. Introduction

Although Modeling and Simulation (M&S) is often regarded more as an art form than an engineering discipline in many application domains, systems engineering (SE) practices are often used for systematically developing M&S applications. Practitioners' guides such as the Verification, Validation, and Accreditation (VV&A) Recommended Practices Guide (RPG) (MSCO, 2006) or textbooks like Law and Kelton (2000), Zeigler *et al* (2000), and Fishwick (2007) show that M&S relies on SE to create models of systems in order to solve problems. Three major components are part of this SE-applied-to-M&S process: user *requirements* to capture stakeholders' needs, *conceptual modelling* to capture system's parts and relationships, and *verification and validation* (V&V) to evaluate the closeness of a solution (solution in this case is a simulation or system[1]) to a client's need. However, while these activities are supported successfully in SE through the SE process, they do not necessarily extend to M&S. SE relies mostly on modelling to capture requirements in order to build a system that solves a well-defined problem. Further, SE's V&V process is the evaluation of the physical/software system against requirements. In contrast, M&S adds these additional nuances:

- M&S does not only capture requirements, but also a modelling question that drives the purpose of the model and corresponding simulation.
- M&S V&V focuses on answering the modelling question about a phenomenon of interest. In SE, software V&V focuses on comparing the developed software/system against requirements.
- In addition to well-defined problems, M&S is in particular concerned with problem situations or problems whose specification is not universally agreed upon (Vennix, 1996).

While the discipline of M&S is very close to that of SE, the difficulty in evaluating the applicability of processes and practices across the disciplines lies in the fact that M&S is understood differently by different communities of practitioners where some view M&S as a tool (the simulation view) and some as an engineering discipline or even as a scientific endeavour. In short, we lack a model of M&S that can serve as the basis for comparison between M&S and its sister disciplines such as SE and computer science. If we see M&S as an atomic discipline in which M&S cannot be disassociated, the development of models and simulations must be viewed in a holistic approach as contributors to *one* system and not independent activities that are conducted individually and are aligned after the fact. This paper proposes a model of M&S that ties together traditional M&S activities using model theory and uses this model to merge M&S and SE processes.

*Correspondence: *A Tolk, Department of Engineering Management and Systems Engineering, Old Dominion University, 241 Kaufman Hall, Norfolk, Virginia, 23529, USA.*

---

[1] A comprehensive exploration of the term 'system' is outside of the scope of this paper. Informally and generally speaking, a system is a collection of elements (related to one another) exhibiting a collective behaviour. Informally and specifically speaking, a system in systems engineering is a man-made solution to a well-defined problem. There are many other working definitions but until we formally define terms, we choose not to abide by one and consider that informal definitions are equivalent.

The paper is organized as follows: Section 2 presents a brief overview of the state of the art in the activities of requirements capture, conceptual modelling, and V&V. In addition, we evaluate the applicability of these activities to M&S. In Section 3, model theory is used as the mathematical foundation to provide a model of M&S and define relevant M&S terms. In addition, we propose an M&S System Development Framework (MS-SDF) that links reference modelling, requirements capture, conceptual modelling, and V&V. Section 4 presents a use case where the MS-SDF is used to build a simulation that helps stakeholders make decisions of vacating, investing, or maintaining a region when flooding due to sea level rise (SLR) is imminent. Finally, Section 5 highlights key implications of the MS-SDF on designing M&S systems, the V&V process, and interoperability and composability.

2. State of the art

While SE practices are widely reused in M&S, there are no comprehensive studies on how much and to what extent activities such as requirements engineering, conceptual modelling, and software and system V&V transfer to M&S. In order to further elaborate on this transference, we first discuss how these activities take place in SE and contrast them with an M&S view.

Requirements are the cornerstone of each SE approach. There are many definitions for requirements, but most of them focus on the following understanding: *a requirement defines a characteristic of a system including measurable performance*. In this case, measurable performance implies the measurement of desired characteristics with the expectation that as all requirements are satisfied, the system performs as planned. Examples for such definitions are given in detail with references in Buede (2009, pp 151–210). Generally speaking, requirements can be partitioned in four categories:

- *Function and input/output-specific requirements*: This category defines inputs, outputs, functions, and interface requirements.
- *System-wide technology requirements*: This category addresses the 'ilities' of a system—such as reliability, availability, serviceability, usability, and maintainability—as a whole.
- *Qualification requirements*: This category defines the measures of the performance and effectiveness of the system. They are the foundation for VV&A.
- *Trade-off requirements*: This category establishes the gains or losses ratio between cost, schedule, and performance when building the system.

Ultimately, the current practice is to produce a series of documents that provide decision makers with the necessary information to manage and govern their portfolios, make critical design decisions, and establish trade-offs between existing and new solutions. Documents such as Operational Concepts, System Definition Documents, and System Architecture are typical examples of artefacts that are produced.

However, while requirements are accepted as the best way of capturing a client's need in a traditional system design and development project, they are not necessarily the best for an M&S project. According to Tolk *et al* (2011), *M&S has the additional requirement that the execution of the model will provide additional insight into some problem while the traditional view mainly focuses on satisfying the identified requirements* (p 356). In other words, insight in M&S is gained from answers to a research or modelling question asked to a model and corresponding simulation. In this sense, SE provides a strong framework for designing a model of a system. However, it needs to be extended so that it provides the capability to capture a phenomenon of interest and questions can be asked of it. In short, the system must eventually become a simulation that provides the solution space where one can find answer to a modelling question.

In M&S, it is strongly recommended to capture the SE process in the form of a *conceptual model*. A conceptual model is a set of artefacts (diagrams in most cases) that captures the desired system in an implementation-independent model. Robinson (2008) defines a conceptual model as *a non-software specific description of the computer simulation model (that will be, is or has been developed), describing the objectives, inputs, outputs, content, assumptions and simplifications of the model* (p 283). This definition suggests that a conceptual model is the specification of a simulation to be developed and is consistent with the SE idea of a system-as-a-solution-of-a-problem approach. Zeigler *et al* (2000) reflect the same SE approach by defining the system to be modelled within an experimental frame, as the reference for the simulation to be developed. However, a conceptual model is not only about a simulation design, it is also about capturing a system of interest (SE view). Moreover, a conceptual model is also about capturing an abstraction of the phenomenon of interest with the goal of answering a modelling question (M&S view). These views must be reconciled and we will do so in the next section.

Finally, V&V ensures that systems are modelled correctly, and that during the modelling and implementation process no mistakes were made. These mistakes result in incomplete or inappropriate simulation solutions in regard to the modelling question. Very often, validation is understood as the processes of 'modelling the right thing' while verification is understood as the process of 'modelling the thing right'.

It can be argued that this view of V&V of simulation evolved from an SE-driven perspective that found its way into software engineering approaches and was eventually reflected in M&S approaches. Generally, in order to show validity, a real system must be available and perfectly observable under an agreed-upon perspective in order to generate a comparison between what is built and the real thing. The problem is that in many cases, M&S is used where a real system is not available, or not perfectly observable or where the problem is viewed with different perspectives, which leads to different solutions (human social behaviour modelling for instance). Very often, M&S is used to look into possible future developments or to gain insight into situations that are not accessible through real-world experimentation. Even in the case where systems are available, all relevant attributes, relations, and processes may not be known or knowable within a reasonable timeframe so assumptions must be made. In M&S, there is a constant tradeoff between the correspondence to the real world, the computability of the model and the ability to answer the modelling question, which does not exist to the same extent in SE. Consequently, we need to define M&S, V&V such that it captures assumptions due to the lack of full knowledge and reflects different perspectives about what is being modelled.

In summary, three relevant activities, capturing system requirements, conceptual modelling, and V&V, are relatively mature in themselves in SE, yet a *formal* connection showing how they are related and how they can work together in support of M&S has not been established. SE relies on well-defined problems with an agreed-upon solution. The idea of departing from modelling a solution, although applicable to SE, does not transfer to M&S. M&S focuses on studying problem situations, which are problems where there is no agreement on a single specification or even whether there is a problem. Hence, the idea is to explore a solution space through M&S, not to focus on one solution.

In the next section, we introduce the concept of reference modelling and propose a modelling framework that captures all three activities individually and holistically.

3. Reference modelling

In order to integrate the viewpoint of conceptual modelling as a modelling activity with the viewpoint of conceptual modelling as a simulation activity, we introduce the notation of reference modelling to capture the SE view and use conceptual modelling as a way to derive specific M&S views of the reference model. Informally, we define *a reference model as an explicit model of a real or imaginary referent, its attributes, capabilities, and relations, as well as governing assumptions and constraints under all relevant perceptions and interpretations*. The reference model captures what is known and assumed about a problem situation of interest.[2] It captures requirements and theories and allows the identification of inconsistencies as well as under- or over-defined areas. It is complete in the sense that it captures what is known and lends itself to multiple and even competing interpretations. A conceptual model, in contrast, focuses on consistency because it builds the foundation for a computer implementation (in a constructive simulation that is) that provides consistent answers. As such, a conceptual model is derived from a reference model focusing on the consistent elements of a reference model and thus facilitating a specification of a simulation.

Both models—the reference model and the conceptual model—are needed and useful, as non-trivial formal models are neither consistent nor complete, but cannot be both (Gödel's incompleteness theorem establishes the inherent limitations of all but the most trivial axiomatic systems capable of, for example, being complete and consistent). Therefore, for non-trivial cases it is likely that we will end up with the choice for completeness (capturing and addressing all requirements, even when they are not consistent) and consistency (being the foundation for the computer implementation). This logical argument already motivates the use of two formal models, one for completeness in support of governance (reference model), and another one for consistency in support of implementations (conceptual model). In the highly unlikely case that one single simulation model is sufficient to address all requirements, only one layer is needed and in that case conceptual model and reference model merge into one common model.

As we begin to informally understand the distinction and interrelation between reference modelling and conceptual modelling, we need to explore in a more rigorous manner the M&S process in order to understand how it relates to the SE process. In order to formally define reference and conceptual modelling formally, we need to model M&S itself and formally define what we mean by model, simulation, and simulator. In the next section, we use model theory as the foundational theory for defining M&S.

3.1. Mathematical foundation

The main challenge we want to address with M&S is that the same statement about a real or simulated object can be true under one viewpoint, but false under another viewpoint especially in problem situations. This challenge is more apparent in interoperability and composability, for instance, since, as stated by Tolk *et al* (2011), the conceptualization of a referent as captured in the reference model becomes the reality for the simulation. Composability of models copes with the question of whether the assumptions and constraints of two conceptualizations are consistent, or if the resulting model of combining conceptualizations remains consistent. That is why model theory is of particular interest, as the *fundamental tenet of a model theory is that mathematical truth, like all truth, is relative. A statement may be*

---

[2] The term reference model, as introduced in this paper, is addressing a collection of all relevant real-world referents and the perceptions thereof in form of a collection of statements using mathematical logic. This approach of capturing real-world referents has been adapted from semiotics and has been applied to address M&S challenges by Turnitsa and Tolk (2008) and generalized by Hofmann *et al* (2011). It should not be confused with the reference model used in systems and software engineering that are abstract frameworks of common concepts.

true or false, depending on how and where it is interpreted (Weiss and D'Mello, 1997, p 1).

In general, model theory is a branch of mathematics that deals with the interpretation of formal languages using set-theoretic structures. Model theory deals with the equivalency of interpretations in different formal languages. The interested reader is referenced to standard literature such as Weiss and D'Mello (1997) for further exploration. The fundamental terms of model theory are the 'formal' *languages* that are used to express the concepts to be evaluated. In order to interpret a *sentence* of a language, a *structure* is needed. This structure interprets sentences to be true or false. The set of sentences that are interpreted to be true builds the *theory*. Structures are therefore understood as the *model* of a language. The theories of these models are the sets of true statements in these models.

The formal definitions according to Weiss and D'Mello (1997) are the following.

Definition 1 A *language* L is a set of logical symbols, including constant symbols, function symbols, and relational symbols.

Definition 2 A *structure* U for a language L is an ordered pair $U = < A, I >$, where A is a non-empty set of logical symbols and I is a function that maps constant symbols to constants, function symbols to functions, and relation symbols to relations. The set A is called the *universe* of a language. The function I is called the *interpretation function*. Combining universe and interpretation function results in the *model* of the language. For each universe A many interpretation functions I can exist; and each resulting structure $U = < A, I >$ is its own model.

Definition 3 A *sentence* σ is an assertion that can be assigned the Boolean value true or false. A language is generated by a set of its elementary sentences and its logical operators.

In particular for computable formal languages, the language can equivalently be defined by enumerating all sentences instead of using a production system approach as shown in Definition 3 and proven among others in Dowek (2011).

The most fundamental concept of model theory is that a sentence σ of a language L is interpreted in a model U to be true or false. If the interpretation is true, we write $U \Vdash \sigma$. This concept is extended to define the additional terms.

Definition 4 Let Σ be a set of sentences. U is a model of Σ whenever $U \Vdash \sigma$ for each $\sigma \in \Sigma$. This is written as $U \Vdash \Sigma$. Σ is *satisfiable* if and only if there is a structure U for which $U \Vdash \Sigma$.

Definition 5 A *theory* T is a set of sentences. If T is a theory and σ is a sentence then we write $T \Vdash \sigma$ whenever we have that for all U we can show that if $U \Vdash T$ then $U \Vdash \sigma$. We define σ to be a consequence of T. A theory is defined to be closed whenever it contains all of its consequences.

Definition 6 If U is a model of L then we define the *theory of the model* U, named ThU, as the set of all sentences of L which are true in U, or $ThU = \{\sigma \in L\text{: } U \Vdash \sigma\}$.

Definition 7 If $\Sigma \subseteq T$ fulfills that $\Sigma \Vdash \sigma$ for every $\sigma \in T$, in other words $\Sigma \Vdash T$, then Σ is a set of axioms of the theory T.

This set of definitions forms the basis for a formal specification of M&S terms allowing the application of theorems and other insights of model theory to M&S.

3.2. Towards an M&S formalism using model theory

Model theory allows defining M&S terms in a very precise way. In this section the authors propose definitions for reference model, model, modelling question, conceptual model, simulator, and simulation.

Definition 8 A reference model is a structure U. A reference model contains the elements of interest and the relations between them. With respect to SE, every requirement, known fact, assumption, and observation can be captured in this structure. As such, the structure U is a complete representation of what we know, require, or assume. However, as pointed out before, requirements can be inconsistent. Similarly, applicable theories can be inconsistent. This is not an issue for the structure, as model theory allows capturing various sets of R_n—defined as interpretations in Definition 2—in the same structure. As a simple example one system can interpret a symbol as a constant while another can interpret it as a function. Another possibility is that different people have different assumptions (axioms), which can lead to different interpretations of the same statement or different conclusions all together.

When we define the conceptual model, we want to build a simulation system that is consistent. We want the same interpretation of truth in this model (a constant is a constant and not a function or both, the assumptions and constraints are unique and consistent), so that we have to exclude interpretations with different results. As a simple example, we want to make sure that all simulation components are based on a consistent interpretation of the laws of physics to avoid inconsistent results.

Definition 9 A model is a language L, denoted $L_{M\&S}$.

Definition 10 A modelling question is a collection of sentences Σ.

Definition 11 A conceptual model is a language $L_{M\&S}$ satisfiable under a reference model.From a model theoretic perspective, a model is a collection of sentences that may or may not have truth value assigned, a modelling question is a collection of sentences without truth value (subset of $L_{M\&S}$), and a conceptual model is a set of sentences produced by the reference model. Following Definition 4, in order for a conceptual model to represent a consistent subset of the reference model U, the language $L_{M\&S}$, which is the set of sentences, must be satisfiable meaning every sentence can be evaluated as true or false. If the conceptual model is made of only true sentences, it is a theory of the reference model (ThU). In other words, reference modelling, conceptual modelling, and validity are indivisibly interconnected, which mandates the explicit capturing of the reference model and the conceptual model as separate components. If the language/model $L_{M\&S}$ is satisfiable by the subset of the structure/reference model U, the conceptual model is valid under the reference model. In other words, we formally equate validity with satisfiability. The intuitive understanding of 'modelling the right thing' is reflected by the existence of a structure under which the conceptual model is satisfied.

As the focus of many M&S projects of interest is on computer simulation, we introduce a series of definitions that allow extending these insights to the M&S terms digital simulator and simulation.

Definition 12 A digital simulator is a finite state machine (FSM).

An FSM is a triple $\{I, S, O\}$, where I is the set of inputs, S is the set of states, and O is the set of outputs. Several papers show the applicability of the Discrete Event System Specification (DEVS) to this problem, among them Zheng and Wainer (2003), but this discussion goes beyond the scope of this paper. Obviously, each FSM produces a language L_{FSM} when being executed: the triple defining the FSM—or in some cases a subset thereof—can be mapped to sentences that are interpreted in a structure U. In model theory, the sentences produced by a simulator under this definition result in the definition of—digital—simulation.

Definition 13 A digital simulation is the FSM realization of $L_{M\&S}$.

As we already showed in the explanation of Definition 11, validity equals satisfiability, which applies to the simulation as well. A simulation is therefore valid if the language L_{FSM} is satisfiable as evidenced in Definition 4. The conceptual model and the implementing simulation must therefore be equivalent languages, which explain the state of the art's intuitive definition of verification as a way to ensure loss-free transformation between specification and implementation. In other words, if the transformation is loss-free, the languages defined by the artefacts are equivalent and satisfy the same reference model. Verification is simply the ability of the simulator to produce a simulation, which corresponds to the intuitive understanding of 'modelling the thing right'.

This subsection on mathematical foundations for M&S motivates the explicit definition of reference models and conceptual models as complementary components within the M&S formalism. Only if one single interpretation relation exists for all application cases, stakeholders, or agile requirements, reference modelling and conceptual modelling are identical. In all other cases, the reference model is the complete collection of all universes and the potentially inconsistent interpretations while the conceptual model and the implementing simulation are both languages satisfiable under the consistent subset of the structure they implement. It is important to note that the use of model theory in this paper is consistent with that of authors such as Mario Bunge (1973, 1974, 1998) in how the concept of model is defined. In our case, we note that:

- M&S itself is considered a problem situation, or referent, whose main terms need unambiguous definitions. The M&S formalism provides those definitions to not only facilitate their communication, but also serve as a deductive apparatus.
- The formalism is not intended to explain how to capture models using model theory. Model theory is used as the means to generate the formalism. In this case, M&S is the object of interest and model theory the apparatus to create the formalism. Using Bunge's terminology, model theory is used to develop a model object (M&S formalism), which paired with the theory of computation as a general theory as we did in this paper to define digital simulation provides the basis to derive a theoretical model (specific or interpreted theory) of M&S.
- The resulting theoretical model, by being formal, would allow the deduction of properties of models and simulations, namely, composability, interoperability, and validity. A full discussion of these properties is outside the scope of this work and is proposed as future work.
- Model theory is used to define, in addition to the concept of model, concepts like simulator and simulation and to establish differences between concepts such as reference and conceptual model.

While this section motivates the need for reference and conceptual modelling for requirement capture, conceptual modelling, and V&V, the following section recommends some supporting methods to conduct this work coherently.

3.3. An M&S framework for system development

From the last subsection it became apparent that a systemic approach is needed in order to capture a reference model,

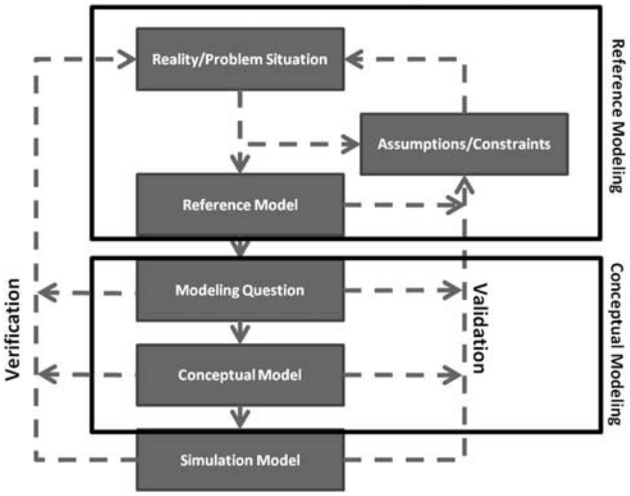

Figure 1 The M&S System Development Framework (MS-SDF).

a conceptual model, a modelling question, and a simulation. In this subsection, we propose an MS-SDF that provides the basis for migrating existing specifications into the formal realm. In order to generalize, we use the notion of a problem situation as the object of the M&S study and organize the items defined in the M&S formalism into a cohesive whole. On the basis of the M&S formalism, we group M&S activities into reference modelling, which is focused on moving from a problem situation to a well-defined problem based on assumptions and capturing it in the form of a reference model. This reference model is used as the basis for conceptual modelling, which results in a conceptual model based on a modelling question.

The MS-SDF (Figure 1) has the following steps:

- *Reality/problem situation.* The problem situation is the phenomenon of interest whose participants diverge on what the problem is. This constitutes the anchor point for the M&S process. In general, we are either trying to reproduce some aspect or study some properties of the problem situation. The main goal at this stage is to find every key stakeholder's view of the problem situation and capture this knowledge in the form of sentences believed to be true of false about the problem situation;
- *Establish assumptions/constraints.* Every stakeholder has assumptions about what they know of the problem situation. These assumptions need to be made and explicitly captured in the form of additional sentences that can be either true or false. The process of coupling knowledge with assumptions is iterated to find which assumption to keep (reasonable assumptions for example) and which to discard. This iteration process leads to the creation of a reference model that holds knowledge and assumptions about the problem situation.
- *Capture the reference model.* The reference model is what we currently know and assume of the problem situation. It is acceptable and highly likely for the reference model to have contradictive statements about knowledge and assumptions describing the problem situation. The goal of reference modeling is to completely frame the problem situation, i.e., capturing all viewpoints of all stakeholders. It is noted that by completeness we mean sufficiency; that is the reference model sufficiently captures the understanding of everyone involved. From this point forward the reference model becomes our constructed reality on which all subsequent artefacts rely upon. In SE, there is a set of documents such as the architecture design document that captures the design and architecture of the system. In M&S, we do not have a direct counterpart for these types of documents even though in some cases there is a high-level description of the model that is provided *ad hoc*. For M&S, we recommend that a reference model be captured formally. A formal representation provides the following advantages:

  ○ Syntax and semantics rigorously defined (no emergence).
  ○ Precise form, logical, and computable.
  ○ Eliminate imprecision and ambiguity (V&V).
  ○ Provide basis for mathematically verifying equivalence between specification and implementation.

  In our use case example, we use the web ontology language (OWL) in combination with rules and inference engines to capture the assumptions and constraints and be able to reason over the ontology. The resource description framework is another possibility and there are open source enterprise toolkits (Protégé-OWL, Swoop) that integrate OWL, Resource Description Framework (RDF), rules engines, and reasoners in a coherent whole. There is also ongoing work on a System Modeling Language (SysML) to an OWL bi-directional translator. These tools can help us semi-automatically generate the reference model.

- *Formulate a modelling question.* As previously mentioned, the modelling question is a sentence or a collection of sentences that can be asked of the reference model. In other words, it is a collection of queries about entities and their relationships as captured by the reference model. These modelling questions can be stored as part of the reference model, conceptual model, or created separately to check whether the reference model can be used to answer them. The reference model might contain the required entities, properties, rules, and the necessary assumptions to answer a question. In case that the reference model does not have these elements, a decision has to be made into whether or not it needs to be extended. In either case, the Manchester OWL syntax can be used to formulate modelling questions. Protégé-OWL support Manchester OWL as part of its framework.
- *Create the conceptual model.* The conceptual model captures how we intend to answer the modelling question. Several alternatives are possible but it should be noted

that the conceptual model should not violate any of the assertions made in the reference model. Further it must remain consistent, which means that if there are inconsistencies in the reference model they have to be resolved by a subject matter expert (SME) if possible or further assumptions have to be made. These new assumptions are called 'constraints' to distinguish them with the existing assumptions. Each conceptual model can have a set of constraints based on the modelling question they intend to answer. Constraints have to be included with the reference model to ensure completeness. In essence, the conceptual model has to be evaluated against the reference model until a consistent conceptual model is reached. The other reason we seek a consistent conceptual model is an issue of logic. The conceptual model represents a theory of the reference model, which means it should not contain any false statements because from false we can logically arrive at true statements. Arriving to true from false statements is of concern because false statements can lead to emergence and accounts for strange and unexplainable behaviours in simulations. Once it is consistent, the conceptual model can reflect the paradigm chosen by the modeller to answer the modelling question. In order to capture the conceptual model we can use a standardized syntax such as the Unified Modeling Language (UML) or its system's engineering extension known as the SysML. In addition to providing a standardized syntax, UML and SysML remove the ambiguity associated with natural language and allows for basic consistency checking. However, these standards also allow for imprecise semantics, which implies that other sources of error and misunderstanding might still exist. Formal languages can also be used to define the conceptual model. For instance, the DEVS can be used if the intent is the use a discrete event approach.

- *Create the simulation model.* The simulation model is the realization of the conceptual model derived from the reference model based on the modelling question. In the case of a digital simulator, the simulator is the FSM realization of the conceptual model; the modeller can select a software package of choice and use it to generate the model. It is important to note that all software components that are used to generate the simulation are not part of the simulation, which is precisely defined as the FSM realization of the conceptual model. *Compromise is a key componen*t of this framework. We do not assume that all reference models can be conceptualized and all conceptual models can be simulated. For non-trivial cases, there are several compromises that have to be made in the form of assumptions and constraints at all levels. The key is to find the balance between the number and types of assumptions/constraints that will keep the model valid and 'simulatable'.
- *Verification and validation.* Understanding M&S as an atomic concept and applying the MS-SDF framework theory results in the unification of validation and verification into one atomic concept as well. Validation is a relation between the simulation, conceptual model, modelling question, reference model, and the assumptions and constraints captured in the form of axioms in the reference model as there is no direct access to the problem situation. Verification is a relation between simulation, conceptual model, modelling question, and reference model. Notice that, on the basis of this approach, verification is equivalent to validation as direct access to reality is not possible. Also note that validity is built-in as simulation is derived from conceptual model and conceptual model from referent model. Further work is suggested in order to facilitate the validation process against the reference model by using model checking tools to test that conceptual models are satisfiable under the reference model.

The current state of the art recommends differentiating between model specification activities and the resulting models (static model, dynamic model, logical model, conceptual model, etc.) and V&V activities, which focus primarily on the simulation model. These activities are usually part of an organizational M&S approach and they are often conducted in sequence or parallel without a concerted approach. Recent tendencies have been to move to an ontological approach where it is recommended to differentiate between methodological ontologies to address the question 'How we model?' and referential ontologies to address the question 'What do we model?' and show which models currently used fall under which category (Hofmann *et al*, 2011). The proposed MS-SDF utilizes these recommendations and improves them by ensuring that the identified challenges are not only addressed, but that they are within one consistent approach manipulated through task-specific facets.

On the basis of the MS-SDF, we can identify the main activities of M&S as the specification of a model (modelling activity) and the generation of the sentences of a model (simulation activity). From the model theoretic view, modelling is the creation of a language that describes a problem situation. This language can have several levels of abstraction and expressiveness to cope with multiple aspects of the problem situation. The MS-SDF unifies the reference model, the modelling question, the derived conceptual model, and the simulation model under the same modelling activity resulting in transparent models. The simulation activity is simply the enactment or execution of the model, producing the output data producible by the model. The MS-SDF also incorporates V&V activities as consistency checking and satisfiability proving between the artefacts produced by the modelling activity. The proposed approach allows for these activities being automated for the most part. M&S and V&V activities are linked within the MS-SDF.

In the next section, we present a use case example where we apply the MS-SDF to show the applicability of these ideas.

Table 1 Sample list of actors involved in sea level rise (SLR) decision

Actor	*Description*
Decision maker	Moral or physical entity who has to make decisions about the SLR situation
Person	A physical human being
Area	A geographical location that has one or more infrastructures in which one or more person lives
Critical infrastructure	A physical entity or a service that is affected by SLR
Mitigation factors	Physical entities that can be used to negate the physical effects of SLR
Resource	An entity required to make something work
Investment	A government or private business entity

4. Use case: designing an SLR decision support simulation

The use case reflects a project where we use the proposed MS-SDF to capture the problem situation and rely on the concept of reference model to build in the validity of the conceptual model and respective simulation. Semi-automated validation is part of future work. In this use case we focus on applying the framework and purposefully do not provide specific results as it is out of scope of this paper. Instead, we focus on showing traceability and consistency.

4.1. Problem situation

The problem situation is to provide stakeholders of the Hampton Roads, VA region a decision support simulation in order to decide today what areas need to be evacuated or protected due to flooding 50 years from now. The Hampton Roads region of southeastern Virginia is one of the most threatened areas in America. Most land within the region lies less than 15 feet above mean sea level and the combined impacts of rising waters and continental plate sink result in forecast relative sea level increases of 1½–2 feet over the next 100 years. Numerous studies have projected the extent of potentially flooded land areas and mapped threatened communities. Others have listed decision-impacting influences from particular perspectives. However, no studies have objectively identified a comprehensive list of factors that must be considered in the decision-making process and organized these factors in a manner that considers all simultaneously.

Making these decisions about evacuation or protection, before imminent flooding due to sea lever rise, is everything but simple. As there are many stakeholders involved, there are many different and divergent criteria used to define both the problem and an acceptable solution.

4.2. Reference model

The reference model is the starting point of modelling activity. From the problem situation standpoint, we know that SLR affects people, the environment, and infrastructures within an area. We also know that SLR has caused some resources to be expanded in order to protect certain assets and we know that these decisions are made by elevated officials or decision makers. In addition, we know that people and businesses are affected by the outlook of an area and the decision made by the leadership regarding SLR. From this situation, we are asked to model the SLR problem. Table 1 shows an example of the main actors that play a role in this problem situation. We capture the SLR problem situation in an ontology using the OWL.

Each actor is examined further in order to capture what is known or assumed. For instance, we use the Department of Homeland Security Infrastructure Data Taxonomy as the basis for modelling the critical infrastructure because it is used to enable transparent and consistent communication about critical infrastructure and key resources between government and private sector partners. The taxonomy contains 18 factors listed below:

(1) Agriculture and food
(2) Banking and finance
(3) Chemical and hazardous materials industry
(4) Defense industry base
(5) Energy
(6) Emergency services
(7) Information technology
(8) Telecommunications
(9) Postal and shipping
(10) Healthcare and public health
(11) Transportation
(12) Water
(13) National monument icons
(14) Commercial facilities
(15) Government facilities
(16) Dams
(17) Nuclear facilities
(18) Manufacturing

Starting from this set of actors, we add logical rules about how the factors relate to one another, on the basis of theory, assumptions, and SME input. As a result, theories, assumptions,

Table 2 Assertions and theories in the reference model

Assertion—Decision makers	*Theory*	*Implementation*
Decision/policymakers have several options for dealing with sea level rise (SLR)	There are several options for mitigation: ● Building dikes ● Raising land ● Relocate people/assets ● Dredge coastlines	Every decision is made by at least one decision maker
Each decision option has a cost	There are several types of cost for a decision: ● Monetary cost ● Environmental cost ● Political cost	Every decision has some cost
Decision makers can have different goals for a given area	Decision makers decide what asset to protect based on the homeland security taxonomy	Only decision makers can select what they want to protect and an asset is protected if and only if a decision maker wants to protect it
Decision makers have different terms of employment	If SLR is not expected to affect an area during a decision makers term in office, then SLR may be overlooked	Decision makers can decide to do nothing in response to SLR
Decision makers do not have unlimited funds to allocate for SLR	Decision makers can allocate funding from their budgets or try to borrow money	Decision makers can only allocate money that they have or that they can potentially collect

and SME input are made explicit in the ontology. The ontology allows us to make logical deductions and flag inconsistent statements. As an example, Table 2 shows a list of assertions from decision makers having to deal with SLR. In addition, we show the theory that models this assertion and how that theory is eventually captured in the reference model. We focus on capturing only capturing assertions that are true. We could equivalently capture sentences that are false by using the 'Not' operator. For instance if every decision is made by at least one decision maker we can equivalently state that *no* decisions are made without at least decision maker being involved or it is *false* to have decisions made without a decision maker being involved.

In Table 3 we have a list of assumptions that we need to make about the SLR problem in order to define a manageable problem space. While an assertion is a statement that we know is true because the SME or some other source has justified it, an assumption is a statement that we accept as true without having to justify it. For instance, we know that a decision maker has several options for dealing with SLR but we have to assume that not all options are available and that we cannot compute all possible types of costs associated with a decision.

At this stage, the reference model is an organized collection of what we know about SLR including the main actors, roles, and relationships involved. Relevant theories are included where necessary. For instance, we use *supply-side economics* as the underlying economic theory that governs the economic life of an area and use *Maslow's Hierarchy of Needs* to represent how individuals select where to live.

It is also worth noting that there are several seemingly contradictory assertions that have to be further specified. For instance the fact that decisions makers are involved in every decision could contradict the assertion that decision makers can decide to do no nothing depending on the modelling question. We will discuss this point further during the conceptual modelling process. As shown by the MS-SDF, the reference model is a living model that can be amended at any moment in the modelling process. Once we have a reference model, we can begin to formulate a set of modelling questions

4.3. Modelling question

In order to create a simulation that can model the effects of SLR on an area, a consensus must first be reached that forms a specific question or view relating to SLR. On the basis of the different perspectives surrounding SLR and discussion with SMEs, we determine that there are three main options when dealing with SLR. These options include vacating, maintaining, or investing in the area. Vacating an area means that the area will continue to receive its normal upkeep but no additional funds will be allocated to protect or reduce the effects of SLR on the area. Maintaining an area means that funding within the area's available budget can be allocated to protecting or mitigating the effects of SLR on the area. Investing in the area means that additional funds will be procured in order to mitigate the effects of SLR in the area.

Table 3 Assumptions and constraints in the reference model

Assumption—Decision makers	*Theory*	*Implementation*
Decisions that deal with sea level rise (SLR) can only be grouped into three categories	Decisions are driven solely by monetary concerns	The least costly decision is the one always selected
Decision maker can decide to 'Vacate' an area	It is possible to vacate an area	A vacated area does not receive any investments (No budget entries for vacated areas)
People choose to live in an area based on their worldviews	A worldview is the set of parameters that must be present in the area in order for the person to choose to live there (ie meeting some basic needs)	A person only stays in an area if his or her needs are met
Decision maker can decide to 'Maintain' an area	It is possible to apply mitigation measures from funds within the available budget	The budget for maintained areas reflects only the maintenance amount (no additional investments)
Decision maker can decide to 'Invest' in an area	It is possible to apply mitigation measures that require borrowing addition funding	Additional capital is added from borrowed money
The decision to vacate, maintain, or invest can be applied to individual categories within an area	A decision maker can decide to invest in protecting an infrastructure while refusing to protect another	An infrastructure is protected only if it is directly targeted for protection (no secondary protection due to geographic proximity for instance)

Table 4 Sample set of modelling questions

Actor	*Modelling question*
Decision maker	What is the best combination of factors that would make an *area* as attractive as possible for as long as possible
Person	What is the best *area* (area that meets most of the needs of most people) for a person to live? How about businesses?
Area	What types of areas is most/lest attractive; under what conditions
Critical infrastructure	What is the combination of critical infrastructure that should be protected in order for an *area* to survive or thrive?
Mitigation factors	What combination of mitigation factors is most/least useful in protecting an area?
Resource	What is the best use of resources in order for an *area* to thrive?

However, the determination on whether to vacate, maintain, or invest in an area is based on how important the area is from the viewpoint of the decision maker. Therefore, it is important to define the value of an area. Defining the concept of value is difficult because it is a subjective term. For this model to be of use in modelling the problem situation, value must be defined objectively by not only incorporating the traditional and quantitative metrics, for example, GDP, economic activity, population, commodity production, and consumption, etc., but also social welfare factors such as healthcare, cultural, and historical landmarks, etc. In this case these intangibles can be evaluated side by side with the more concrete causes. All of the metrics that are associated with the value of an area are captured in the reference model so that they can be referred to as the model is constructed. The simulation needs to support what-if scenarios and course of action analysis in order to establish the combination of factors that would lead to either one of the three outcomes (a Vacate/Invest/Maintain heretofore VIM decision). This leads to one possible modelling question (many are likely): what is the combination of factors, under each stakeholder's perspective, that leads to the outcome of a VIM decision? This modelling question can be further specified as summarized in Table 4.

Each of these modelling questions could warrant a very detailed model on its own but as we focus on the area and the VIM outcome, it is important to model the different perspectives of an area as represented by each of the questions. In this case, we have to ensure that our reference model contains sufficient information to answer these questions otherwise we have to augment it with either assertions, assumptions, or both. At this point, we assume that we have sufficient information and attempt to formulate a conceptual model based on the reference model and the set of modelling questions. As we move forward in the process we will revisit the reference model to ensure satisfiability, that is each new assertion can be traced to the reference model either directly or by a combination of already existing assertions.

4.4. Conceptual model

Since the conceptual model is ultimately what is going to be implemented, we decide to select the modelling paradigms that will help us answer the modelling questions. We also ensure that the conceptual model is consistent by either further refining the assertions made in the reference model or adding assumptions. For instance, we decide to add an assertion to the reference model stating that a decision maker can decide to vacate, invest, maintain, or do nothing. By doing so, we *explicitly decide* not to directly model the cost associated with doing nothing and instead let the consequences of that alternative emerge. In addition, we *decide* to model several areas in parallel and examine the role played by competition or cooperation if any. Once these types of decisions are added to the reference model we return to the modelling question and begin to formulate the conceptual model such that it is a consistent subset of the reference model. We capture the conceptual model in UML, which becomes our *de facto* meta-language. The detailed discussions on how to transfer knowledge from OWL into UML is outside the scope of this paper, but the implications will be addressed in the following section. While we purposefully do not show the UML different artefacts, we will discuss the high level view of the conceptual model.

The modelling questions formulated above address the problem situation from a micro (person), meso (mitigation factors), and macro level (critical infrastructure). This requires a multi-paradigm modelling approach to create the appropriate environment for the simulation (simulated environment). Agent-based modelling (ABM) is used to create autonomous entities and to define rules for the entities within the system and system dynamics (SD) modelling is used to represent the behaviour of the areas (ie cities, regions, states, etc.) within the simulated environment. ABM allows for the beliefs, desires, and intentions of each of the entities to be set and then allows for the behaviour of the system to be observed with respect to the actions of the agents. SD helps in understanding the long-term effects of the model over long periods of time. The agents of interest for the simulation are the 'Person' in the system and the 'Investments' in the system. The Person agents represent the general population that is moving into or out of an area while the Investment agents represent businesses or governmental Investments moving into areas. These agents contain predefined values for how they interact with the area while they are in it. This system interaction will be explained further in the SD model. Each of these agents follows a specific set of rules for determining if they want to remain in their current area or if they want to leave and find a new area. The rules unique to each of these agents were defined in the reference model.

The Person agents have a world view to follow in the simulation. The world view represents an initial set of conditions that must be met in order for a person to move into a new area. The Person agents move through the simulated environment looking for new areas and when they arrive at the area they check their requirements (different for each Person agent) against the conditions of the area. If these conditions are not met then the agent leaves the area where it is currently and looks for a new area. If the conditions are met, then the agents check a more specific set of conditions (called impact attributes) against the values of the area. The impact attributes represent the specific factors within the area that the agent will have the greatest interaction with while in the area. The impact attribute values per agent are assigned as a random distribution so that each of the agents has a different set of values. This is designed to represent the different types of Person that move into an area, such as a banker or a farmer. These conditions are different for the Person agents and the Investment agents.

The Person agent first looks to see if the area can meet its basic need requirements as defined by *Maslow's Hierarchy of Needs*. The area must be able to provide food, water, and shelter for all agents within the area and the agents have a threshold that the area must be greater than in order for the agent to consider joining the area (ie the area must be able to provide a minimum of X gallons of water per person per day based on the value required by each agent). If these conditions all pass then the agent compares its impact attribute values against the values of the area. The impact attribute is the main component in deciding if the agent finds the area acceptable to live in because it represents the factors that the agent finds most valuable to itself. For example, if the agent's impact attributes want an area with a high agricultural base then that agent will refuse to stay in an area with a low agricultural presence even if every other component in the area is prospering. The agent continues to recheck its world view and impact attribute values while within the area in order to make sure that the area remains an acceptable living environment.

The Investment agent also checks a world view value and an impact attribute value against the areas but these conditions are created differently for the Investment agents. The Investment agent represents a wide variety of businesses and government expenditure values based on the values that are randomly assigned to the entities. In terms of the world view for the Investment, the area must be able to financially support the addition of new businesses. In terms of the impact attribute, the area must have the necessary infrastructure to justify the new Investment joining the area. For example, if the Investment agent represents a government expenditure on dam repair then there has to be at least one dam in need of repair in the area in order for the Investment agent to join the area.

SD modelling is used to handle the changing dynamics within the areas of interest. An SD model is created for every area that is being represented within the environment. As discussed in an earlier section of this paper, this model represents the 18 factors that are of greatest interest within the area. These factors are agriculture and food, banking and finance, chemical and hazardous material industry, defense industrial base, energy, emergency services, information technology, telecommunications, postal and shipping, healthcare and public health, transportation, water, national monuments and icons, commercial facilities, government facilities, dams, nuclear facilities, and manufacturing. The SD model is designed to represent how each of these factors is related to each of the other factors. These factors are also affected by the number of Person agents and Investment agents that have entered the area.

By relating all of the factors of the area to each other, the effects on the system of drastically changing one of the factors can be observed. This allows for the area to be tested under different conditions in order to observe how the dynamics of the area respond. For example, if an area has a large energy infrastructure but the SLR is threatening to cause serious damage to that particular infrastructure then the effects can be observed in the simulation. Then the simulation can be rerun to view what happens if money is invested in protecting the energy infrastructure from the potential damage. This can then allow for the costs of saving the energy infrastructure to be compared against the cost of not saving the infrastructure to provide an estimate of how the system will be affected either way. This same test can be run for all of the factors within the area to help understand how the system will respond under different conditions.

Changing the dynamics of the system also serves to make the area more or less attractive to the agents that are living in the areas. If the conditions in the area become unbearable to the agent then the agent will leave the system. As more agents leave the system, thus removing their financial contributions to their areas (ie disposable income), the area may become even less desirable and an even greater number of Person will start to leave until the economic value of the area completely collapses. This affect can also be observed while testing changes in the dynamics of the system in order to provide a check on the emergent behaviour of the system. Protecting one aspect of the area completely while completely ignoring other parts of the system may show the best results for the area financially, but the ABM portion of the model helps to determine if the change will cause a reaction forcing all of the population to vacate the area and cause the area to die in the future. At the same time, it could also show that making certain changes will make the area more desirable to live in and increase the number of Person and Investment agents into the area.

4.5. Simulator and simulation

The simulator is a computer running Anylogic 6.6. We selected Anylogic because it provides support for combining agents and SD simulations. As such, Anylogic is the FSM capable of generating the conceptual model. The simulation is the FSM realization of the conceptual model. Figure 2 shows a screenshot of a running simulation prototype.

The prototype shows three agents: Person, Investments, and Areas. Each area includes the 18 above-mentioned factors. Each factor is an SD simulation. Depending on how appealing an Area is, there is going to be an inflow or outflow of Person into or from that Area. It is important to capture these dynamics as VIM decisions are made for the long term (decisions must be made today that affect the Area 50 years from now when the flood occurs).

5. Implications

In addition to requirements, conceptual models, and V&V, the MS-SDF facilitates composability. By specifying the reference model separately, *all conceptual models built from the reference model are composable* as they depart from the same foundation. Although the maximum benefit of this approach will be the consistent definition of software derived from a consistent subset of the reference model in a top-down approach, it is also possible to re-engineer many assumptions and constraints in a bottom-up approach. In other words, conceptual models captured in UML or SysML can be used to generate parts of the reference model. Therefore, the framework supports future developments as well as reverse engineering of legacy solutions.

The approach of combining UML, SysML, and OWL to increase collaboration based on common concepts and better knowledge sharing capabilities is used in other domains. Successful examples are presented among others during the annual Product Data Exchange workshops between the National Aeronautics and Space Administration (NASA) and the European Space Agency (ESA), accessible via http://step.nasa.gov/.

To give some practical examples, the MS-SDF can be applied to the following areas.

- *Federate selection*: Selecting the best federate to support an M&S effort such as an international computer-assisted exercise is a challenging task. By having the subset of the reference model formally available, a federate selection can now be supported by reasoning over the tasks to be

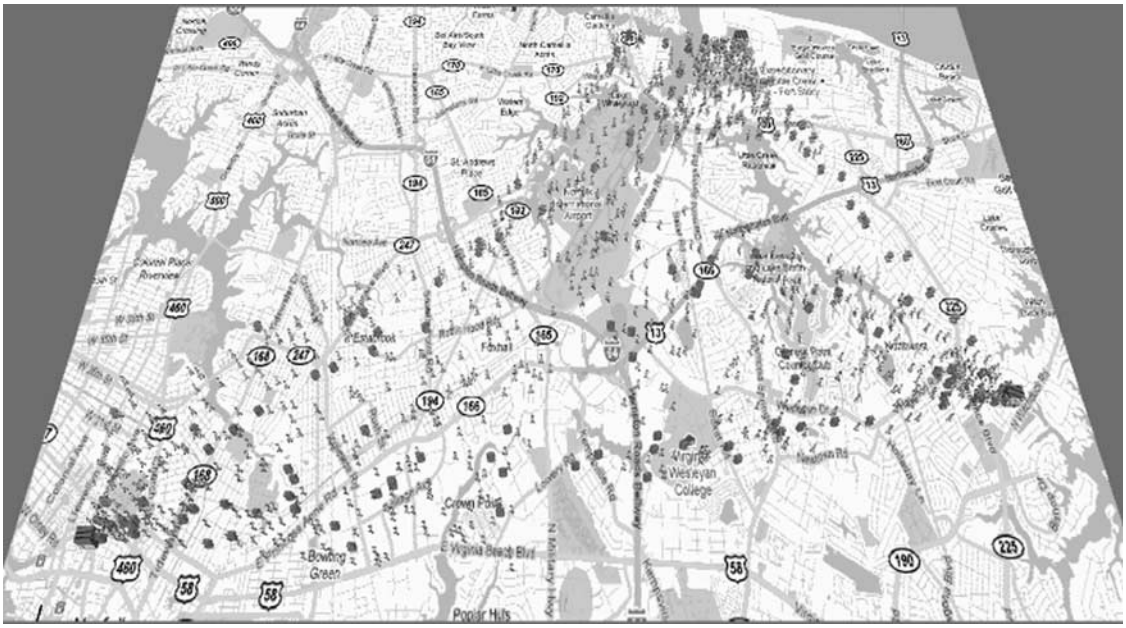

Figure 2 SLR PS simulation.

supported and the suitability of potential solutions. In other words, if the conceptual model is satisfiable under the reference model of the exercise, the solution can support the exercise. Conceptual models can be federated if they can be satisfied under the same subset of the reference model.

- *Federation versus multi-simulation*: If conceptual models are derived from the same reference model, then the federation of solutions (corresponding to those conceptual models) is possible. This is because these conceptual models are generated from the same structure and therefore can be satisfied under a reference model. If conceptual models are generated from different reference models (or they cannot be traced to a common reference model or they share universes, but not the same interpretations), then a multimodel solution or the development of a common higher-level reference model is needed. As such, the proposed work is not only the logical continuation of the work described by Yilmaz *et al* (2007), it also provides the theoretic justification of multimodeling approaches. Many current efforts target the establishment of a 'common ontology' for a common domain. These efforts are directed at specifying a common reference model and can be integrated to facilitate the tasks of the systems engineer.
- *Multi-resolution modelling*: The term multi-resolution modelling is overloaded as it addresses several different aspects of misaligned data structures between two models that represent the same real-world referent. By explicitly modelling the real-world referent in the reference model, where each concept is unique, we already avoid the problem of synonyms (different word referring to the same concept) and homonyms (same word referring to different concepts).

It is important to note that although reuse is not defined, it is supported. We understand under reuse to *use a model for a different purpose than the one it was originally designed for*. However, if the model is applied in a new context to answer new questions, is it still valid? So far heuristics and SME judgement are the only options to rely on. With formal models of the referent and the modelling questions the proposed MS-SDF helps to formalize the *validation* of a model regarding its reuse. To this end, the new context of the reuse establishes the reference model. Consistently with the model of the referent, the modelling questions to be answered are used to build a conceptual model of a solution that could address the questions.

Following Definition 13, it needs to be decided if the language of the current simulation is satisfiable under the new reference model. Alternatively it can be evaluated if the new conceptual model is part of the language of the current simulation solution. Both ways are possible and supported by the proposed MS-SDF.

6. Summary and future work

This paper proposes the use of an MS-SDF that ties together SE and M&S. We propose the specification of a reference model, an explicit model of a real or imaginary referent, its attributes, capabilities, and relations, as well as governing assumptions and constraints. The reference model captures what is known and assumed about a situation of interest. It captures requirements and theories and allows the identification of inconsistencies as well as under- or over-defined areas. It is complete in the sense that it captures what is known by the stakeholders and lends itself to multiple and even competing interpretations. A reference model would never be complete in the general sense so it can be expanded as soon as new knowledge becomes available or discovered. We recommend the use of formal languages such as OWL to capture the reference model as it facilitates reasoning over a knowledge base. Conceptual models are the basis for a computer implementation. They have to be consistent as otherwise the resulting system would be based on contradictive interpretations. Conceptual models are subsets of a reference model known to be consistent. Reasoners facilitate the identification of such inconsistencies. Further, it was established that conceptual models derived from the same reference model facilitate composition. Languages such as UML and SysML can be used to capture conceptual models.

Lastly, the recommended MS-SDF ties current independent activities together namely, *gathering requirements* (using SysML structures that can be imported into OWL), V&V (as the reference model becomes the authoritative structure to validate against while most of the current verification efforts become subsumed by semi-automated transformation as demonstrated in the domain of executable architectures), and conceptual modelling.

We propose as part of future work to fully elaborate the M&S formalism. The formalism would not only provide a common language when referring to models and simulations, but also it would provide more details on properties such as interoperability.

References

Buede DM (2009). *The Engineering Design of Systems: Models and Methods*. Wiley Series in Systems Engineering and Management, 2nd edn, John Wiley and Sons: Hoboken, NJ.

Bunge M (1973). *Method, Model and Matter*. D. Reidel Publishing Company: Dordrecht, Holland.

Bunge M (1974). *Treatise on Basic Philosophy Vol. 2*. D. Reidel Publishing Company: Dordrecht, Holland.

Bunge M (1998). *Philosophy of Science: From Problem to Theory*. Transaction Publishers: New Brunswick, NJ.

Dowek G. (2011). *Proofs and Algorithms: An Introduction to Logic and Computability*. Springer: Palaiseau, France.

Fishwick PA (2007). *Handbook of Dynamic System Modeling*. CRC Computer & Information Science Series, Chapman & Hall: Boca Raton, FL.

Hofmann M, Palii J and Mihelcic G (2011). Epstemic and normative aspects of ontologies in modeling and simulation. *Journal of Simulation* **5**(3): 135–146.

Law A and Kelton WD (2000). *Simulation Modeling and Analysis*. 3rd edn, Industrial Engineering and Management Science Series; McGraw-Hill Science/Engineering/Math: New York, NY.

Robinson S (2008). Conceptual modelling for simulation part I: Definition and requirements. *Journal of the Operational Research Society* **59**(3): 278–290.

Tolk A, Diallo SY, Padilla JJ and Turnitsa CD (2011). How is M&S interoperability different from other interoperability domains? *Proceedings of the Spring Simulation Interoperability Workshop*, Boston, MA, April, 2011, pp 12–20.

Turnitsa CD and Tolk A (2008). Knowledge representation and the dimensions of multi-model relationship. *Proceedings of the Winter Simulation Conference*, Washington, DC, December, pp 1148–1156.

US Modeling and Simulation Coordination Office (MSCO) (2006). *Verification, Validation, and Accreditation (VV&A) Recommended Practices Guide (RPG)*. US Modeling and Simulation Coordination Office (MSCO): Alexandria, VA, accessible via http://vva.msco.mil.

Vennix J (1996). *Group Model Building*. Wiley: New York, NY.

Weiss W and D'Mello C (1997). *Fundamentals of Model Theory*. University of Toronto: Toronto, Ontario.

Yilmaz L, Ören T, Lim A and Bowen S (2007). Requirements and design principles for multisimulation with multiresolution, multistage multimodels. *Proceedings of the Winter Simulation Conference*, Washington DC, December, 2007, pp 823–832.

Zeigler BP, Praehofer H and Kim TG (2000). *Theory of Modeling and Simulation*, 2nd edn, Academic Press: San Diego, CA.

Zheng T and Wainer GA (2003). Implementing finite state machines using the CD++ toolkit. *Proceedings of the Summer Computer Simulation Conference*, Montreal, QC, Canada; July.

Received 9 February 2012;
accepted 17 January 2013 after four revisions