Proceedings of the 2011 Winter Simulation Conference
S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.

MODEL THEORETIC IMPLICATIONS FOR AGENT LANGUAGES IN SUPPORT OF INTEROPERABILITY AND COMPOSABILITY

Andreas Tolk

Old Dominion University
Eng. Management & Systems Engineering
Norfolk, VA 23529, USA

Saikou Y. Diallo
Jose J. Padilla

Old Dominion University
Virginia Modeling Analysis & Simulation Center
Suffolk, VA 23435, USA

Heber Herencia-Zapana

National Institute of Aerospace
100 Exploration Way
Hampton, VA 23666, USA

ABSTRACT

This paper evaluates the implications of model theory for agent languages. The tasks of ambassador agents are to represent simulations and identify potential contributions, select the best solutions in light of the question, compose the selected best solutions to provide the new functionality, and orchestrate their execution. Model-based data engineering can help to identify the information that needs to be exchanged between systems, existential and transformational dependencies can be identified using graph theory, and Petri nets can represent the availability of required information. All structures can be computed and fall under the realm of formal languages. Model theory is a subset of mathematics that focuses on the study of formal languages and their interpretations. Interpreting the terms model, simulation, and data of the modeling and simulation community using model theoretic terms allows the application of model theoretic insights. This allows to formally and unambiguously capture requirements for interoperability and composability.

1 INTRODUCTION

Many ideas have been proposed in recent conferences to support interoperability and composability more efficiently. The use of ambassador agents, i.e., agents that represent simulation solutions as a placeholder and that negotiate the applicability of this particular simulation solution in a new context, was first discussed by Yilmaz and Paspulleti (2005). They extended the idea of Logan and Theodoropoulos (2001) to look at multi-agent systems as a challenge of distributed simulation. However, while Logan and Theodoropoulos distributed their multi-agent system, Yilmaz and Paspulleti used agents to represent distributed components of simulation solutions. In their approach, agents had to find out if the represented simulation solutions could meaningfully interoperate with each other.

Ambassador agents are intelligent software agents that have an understanding on assumptions and constraints of the conceptualization that builds the foundation of the simulation solution as well as of the implementation details thereof. As discussed by Tolk and Diallo (2010), agent languages supporting this task must be and can be rooted in a general theory of interoperability. The approach used set theory to motivate such languages.

Within this paper, a more general view on agent languages for ambassador agents is described. As ambassador agents are software, their language must be a formal language. The mathematical branch that deals with expressiveness of formal languages and their interpretation is model theory. As model theory uses set-theoretic structures, a direct connection to the work presented in Tolk and Diallo (2010) is given. Furthermore, this allows to generalize the findings presented in this earlier paper.

The findings derived from this research confirm several so far unrelated results, such as the concepts of machine understanding as described among others by Zeigler (1986), the complexity problem of component selection as described by Overstreet and Nance (1985) and updated for composability discussion by Page and Opper (1999), the idea of languages for ambassador agents introduced by Yilmaz and Paspulleti (2005) by extending the concepts documented in Yilmaz and Ören (2004), and model-based data engineering to construct common reference models for information exchange between heterogeneous simulation solutions (Tolk and Diallo 2005).

2 TASKS OF AMBASSADOR AGENTS

The overall task of an ambassador agent is to represent the abilities and constraints of a simulation solution in the context of a research question (Tolk and Diallo 2010). Given a question of interest to the customer, which can also include the question how to configure a training system to provide best training under given circumstances, ambassador agents communicate with each other to

- identify potential contributions,
- select the best solutions in light of the question,
- compose the selected best solutions to provide the new functionality, and
- orchestrate their execution.

Many currently proposed approaches assume that the distributed components have been implemented as part of a common model and therefore build a partition of the overall distributed system, or system of systems, being modeled. A prototypical example is the work presented by Ahn et al. (2010) on hierarchical federation composition for information hiding. Their idea is that each distributed simulation component represents a technical components of the modeled system, so that implementation details can efficiently be encapsulated without hindering the interoperability of the overall distributed system.

Recent work of the authors of this paper showed that such an assumption can become dangerous when the distributed simulation components are ‘legacy systems,’ i.e., that they have been independently developed for different task and research questions (Tolk et al. 2011). While in other interoperability domains two components are separated and independent by their definition and interoperability can focus on the information exchange between these components over their interfaces, model-based solutions require that the underlying models are aligned as well. Other interoperability domains can use the unique real-world referents that are supported to align their activity. Model-based solutions are not based on such real-world referents, but on a conceptualization thereof that was designed with the research question in mind. If in the real world the referent changes, all services that rely on him are triggered accordingly. If two indecently developed simulation solution use a conceptualization of this real-world referent it is part of the task to align the models to ensure that the conceptualization reflects the change accordingly. In the work of Ahn et al. (2010), the referent are represented as one of the distributed components and interactions with other components are made explicit; information hiding only applies to the internals of each component itself. This, however, is not the general case.

The following example shall help to better understand the task of aligning models. In support of conducting an optimization study for critical infrastructures, three independently developed simulation systems shall be coupled. The first one models the electricity grid within a city, the second one models drinking water and waste water flow, and the last one looks at transportation. All three models make use of bridges, as the bridge provide means for electricity, drinking and waste water flow, and obviously for transportation. However, the information that bridges are used may be hidden within the systems. What is exposed are the resulting nets that show where and how electricity, drinking and waste water, and transportation are distributed over the city. If the optimization study can only take this provided information into account, this can lead to problems. A traffic accident on a bridge in the transportation system would potentially influence the electricity grid as well as water management, but as the real-world referent bridge is independently modeled using different conceptualizations in all three models, an accident in model three may not be noted in models one and two.

As pointed out by Tolk et al. (2011), simulation developers often assume that all simulations supporting a common domain are ultimately derived from the same model that represents the truth as described within this domain. Such a viewpoint assumes that truth exists on its own, that truth is independent of the observer, and reality is separated from the individual who observes it. The term used for this viewpoint by science philosophy is positivism. For a positivist, problems as described above happen because someone made a mistake: an important detail was not implemented, a model was over-simplified, an important relation was overlooked, etc. Such a viewpoint can be justified in models that are based on one common theory, such a physics based models.

However, positivism is not the only viewpoint that can be supported by models. Of particular interest in the context of this paper is constructivism that holds the belief that truth is constructed within the perception of the observer. Reality is relative and is part of the individual who observes it. The majority of social and human sciences subscribe to constructivism. Even the world of physics experiences tremendous challenges on the quest to find the truth: relativity on the big scale and quantum physics on the small scale are extremes that show that even physics-based models can run into conceptual contradictions, depending on which theory is applied. Consequently, if two simulation solutions are rooted in different theories, they will produce incompatible results. The reason for this inconsistency is then not a mistake within the simulation; the reason is that the underlying conceptualizations are different and the models are conceptually not alignable. No technical interoperability standard can overcome this problem.

This insight implies that ambassador agents do not only have to understand and communicate the implementation details on the simulation level and the specifications of supported interfaces and data that can be exchanged, but that they also have to know about the underlying model and the conceptualizations. As stated in Tolk et al. (2011): *"As we are connecting simulated things we need transparency of what we are simulating, as the real world referent use in other interoperability domains has been replaced in the modeling phase by its representing conceptualization in the M&S interoperability domain."* As ambassador agents are intelligent software agents, their option to express conceptualizations is limited to what computers can express.

This task of communicating the underlying conceptualization as well as the chosen implementation is not completely new. Information resource dictionary systems (IRDS) faced a similar challenge, namely that data used to represent referents of the supported domain come in different formats, supporting different structures, and may have been derived by different methods. The supporting ideas to cope with these challenges were standardized in support of metadata registries (MDR) in the standard ISO/IEC 11179 (ISO 2004, 2005), a multipart standard including parts defining the framework, classification, a registry metamodel and basic attributes, the formulation of data definition, naming and identification principles, and registration. The ideas were generalized in the standard ISO/IEC 19763 that defines a metamodel framework for interoperability (MFI) for data in several parts, with part 3 (ISO 2010) extending the principles of MDR to ontology descriptions.

In the context of this paper, the registry metamodel is of particular interest, as it standardizes in machine understandable form how to document conceptualization and implementation of data structures. ISO/IEC 11179 (ISO 2010) defines quadruples to document the conceptualization and implementation details of data.

Figure 1 shows them as well as their relation. The four elements depicted with their relations are defined as follows:

- The *Conceptual Domain* is a set of valid value meanings that can be enumerated or expressed via a description. This set also defines a controlled vocabulary of terms that can be used to express the concepts of the universe of discourse.

- A *Data Element Concept,* a concept that can be represented in the form of a data element, described independently of any particular representation.
- A *Data Element,* a unit of data for which the definition, identification, representation and permissible values are specified by means of a set of attributes.
- The *Value Domain* is a set of permissible values. This set can be enumerated or expressed via other mathematical specifications, such as intervals.

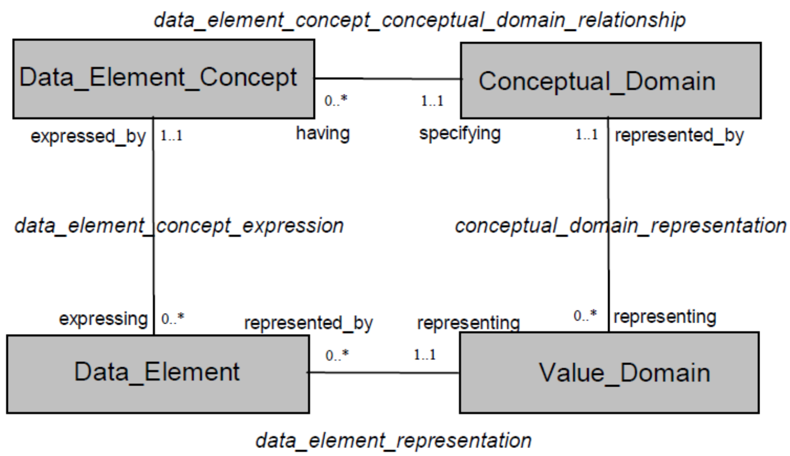

Figure 1: High-level metamodel from ISO/IEC 11179 Part 3 (ISO 2004)

While the first two elements are used to describe the conceptual details, the last two elements describe the implementation details. If we want to talk about countries (conceptual domain), we can talk about different ways to do this, e.g., using the "name" or alternatively a "country code" (data element concepts). The permissible values (value domain) specify what is allowed to define a valid data (data element) that represents a country. Each data is therefore defined as a quadruple of definition (conceptual domain), identification (data element concept), representation (data element), and permissible values (value domain).

For each alternative of how to model a common concept of the domain equivalence classes regarding their expressiveness can be defined. It makes no difference if different value domains are used to express the same data element concepts. To capture the twelve months of a year, we can use their English names ("January," "February," …, "December") as well as three letter abbreviations (JAN, FEB, …, DEC), number (01, 02, …, 12), or any other enumeration of twelve clearly distinguishable values. As long conceptual domains and data element concepts are identical and a bijective function exists, data elements can be lossless mediated into each other. In order to make the assessment if two data elements are equivalent, the conceptual domains and data element concepts are needed as well. The same data element can express many different things, so just focusing on the implementation level is insufficient. Using the quadruple introduces a first mathematical concept to capture conceptualization as well as implementation of data elements.

So far, only data elements can be described. The information exchange between two independently developed simulation solutions is normally richer than just being a set of data elements. To cope with structures as expressed in data models, object models, or complex XML structures, the idea first applied for IRDS structures was used in support of complex mediation by Tolk (2004): propertied concepts that can be associated with each other. The data element becomes a property of a concept. A concept can have several properties that define it and its unique instantiation. Two concepts with identical properties cannot be distinguished. Two concepts with properties of the same type belong to the same concept type. In order to express higher concepts, this propertied concepts can be associated to build composites. For example, for a car, it makes sense to distinguish between carriage, the chassis, interior equipment, and engine. Each of these concepts can be broken into greater detail, e.g., the motor comprises pistons, cylinders, the block, gear and shaft assemblies, etc. Using this idea of starting with the atomic expressions representing the data elements – as defined in the ISO/IEC 11179 metamodel – and extending the definition to allow for propertied concepts and associated concept, an information exchange model can now be expressed as a graph with leaves representing the properties and inner nodes representing the propertied concepts and associated concepts. The associations can furthermore model the existential dependencies introduced by Tolk and Diallo (2010).

These structures allow now to address the three challenges engineers are faced with when they have to map the information exchange models to each other: the models can differ in resolution, in scope, and in structure as explained below.

- If one model uses the inner node of the other model as a property – e.g., one model describes the parts of a car as being the carriage, the chassis, interior equipment, and engine while the other model breaks the engine further apart into pistons, cylinders, valves, etc. – they differ in resolution.
- If the node representing a concept in one model has more associated nodes or leaves then in the other model – e.g., one model describes the interior equipment of a car using seats, radio, and dashboard while the other model uses seats, radio, dashboard, loudspeakers, hat shelf, and dashboard doll – they differ in scope.
- If the graphs representing two models have the same nodes but they differ regarding the edges – e.g., one model associated the gearshift stick with the engine of a car, the other model associates the gearshift stick with the interior – they differ in structure.

These challenges can all happen at the same time, resulting in graphs that – although being derived in support of similar tasks – are very different. Agents must model these structures in order to look for solutions when information is exchanged and the information models must be mediated into each other. Model-based data engineering support this task and was presented in more detail by Tolk and Diallo (2005, 2010).

Assuming that model-based engineering can and has been applied successfully, that means that all data required to be communicated between two models in support of the research question can be produced by the source system and understood by the receiving system, the ambassador agents still have to model if the required information can also be produced in time. The transformational dependencies introduced by Tolk and Diallo (2010) follow into this category as well. Besides ensuring that information is exchanged, ambassador agents have to support the orchestration of information exchange as well. For example, consider two models that may have been selected to support a research task. The information to be exchanged between them, labeled `data_item_1` and `data_item_2`, can be produced and read by the participating systems, however, while the source produces these items in sequence the receiver needs them to be provided at the same time. These kinds of orchestration are supported by Petri nets, so we can use Petri nets to extend the graphs to include orchestration information. These Petri net structures connect the graph representing the sending system and the graph representing the receiving system.

Up to this point, the focus for what the ambassador agent needs to know about the simulation solution has been on the interface. As pointed out in Tolk et al. (2011), composability requires transparency of the models as well. The example using the bridge and its support for three different models given earlier in this sectionshowed that not only is the mathematical expression of the simulation as a production system needed, but each data item within this production system needs to be connected to the represented concept as part of the overall theory that is applied to answer the research question.

As all the introduced ideas are based on finite mathematics, the ambassador agents can gain machine-based understanding of these tasks. Zeigler (1986) identified three requirements that are applicable in the context of understanding the conceptualization and implementation:

- Perception – The observing system has a perception of the system that needs to be understood.
- Meta-Models – The observing system has an appropriate meta-model of the observed system.
- Mapping – Mappings between the perception and meta-models explaining it.

To summarize the tasks and how they are supported by mathematical structures, we start with the research question formulated in the agent language. This describes what has to be modeled in the terms of the supported domain. Each ambassador agent knows what the simulation solution he is representing can do. He therefore can check for each part of the research task if he can execute it within his solution. As he is aware of the internal existential and transformational dependencies, he also knows how much effort is needed to provide this part of the solution. For example, if a simulation model can simulate a traffic light, but in order to simulate the traffic light the whole underlying electrical urban infrastructure needs to be modeled as well, it may not make sense to integrate this model to provide the functionality of the traffic light into a simulation. All ambassador agents know what they can do and can communicate with other what they can do, which provides the basis for selecting who should do what. The complexity of these tasks has already been elaborated by Page and Opper (1999), and some heuristics introduced in Tolk and Diallo (2010). However, all steps are supportable by a formal representation of model and implementation.

As all structures can be computed, so they fall under the realm of formal languages. As such, they fall also under the realm of model theory, which for a language for intelligent software agents is not too surprising, but with the steps of the last section is clearly motivated. The next section will introduce model theory as applicable in the context of supporting agents to compose simulation solutions to answer research questions based on a common agent language.

3 MODEL THEORY AND INTEROPERABILITY

Model theory is a subset of mathematics that focuses on the study of formal languages and their interpretations. It applies logic to the evaluation of truth represented using mathematical structures. In other words, the way we model truth using mathematical structures can lead to different interpretations: what is evaluated to be true in one representation can be false in another one. Ultimately, model theory deals with answering the questions what interpretations of formal languages are consistent, i.e., result in the same truth statements for the same questions.

For the M&S expert, these questions are very familiar when he develops federations: the same scenario initialized and executed in two simulation is likely to result in different outcomes. The main purpose of the former sections was to establish the understanding that M&S can be formally represented as mathematical standards, hence a model can be understood as a formal language. If we can find a way to use formal languages to express our M&S systems, we can apply model theory to find out if simulations can be federated producing consistent results. Let us introduce some key concepts of Model Theory and relate them to M&S and more specifically to M&S interoperability. The following definition are taken from Weiss and D'Mello (1997):

Definition 1 *A language L is a set consisting of all the logical symbols with perhaps some constant, function and/or relational symbols included*

Definition 2 *A model (or structure) U for a language L is an ordered pair <A, I> where A is a nonempty set and I is an interpretation function with domain the set of all constant, function and relation symbols of L such that a constant symbol is mapped to a constant, a function symbol is mapped to a symbol and a relation is mapped to a relation*

Definition 3 *A sentence is an assertion that can be assigned the Boolean value of true or false*

Definition 4 *If U is a model of L, the theory of U, denoted ThU, is defined to be the set of all sentences of L which are true in U*

While these definitions originate from model theory, they are useful in M&S. We define these terms in M&S as follows:

- *Universe:* In Model theory, The universe *A* is a set of constant, functions and relations symbols. In terms of M&S the universe is equivalent to *data*. It is very important to note that the traditional view of data as a collection of constants that are separated from functions and relations is generalized. In the context of interoperability, it means that agents can exchange not only constants but functions and relations since they are all symbols. The current state of the art deals only with constants but in order for the ambassador agents to represent a model, they must be able to deal with functions and relations as well. When associated with an interpretation function (a function with domain the set *A*) each constant, function and relation is given its interpretation. It is important to note that the interpretation function maps a constant symbol to a constant, a function symbol to a function and a relation symbol to a relation. In terms of interoperability, this provides us an apparatus to automatically detect mismatches in the way symbols are interpreted from one model to another. A model or structure in Model Theory is the ordered pair consisting of a universe and its interpretation function.

- *Language and satisfiability:* A language is a set of sentences that can are constructed from logical symbols with a combination of elements from the universe. There are formal rules for constructing *s* sentences and we recommend that the interested reader consult Weiss and D'Mello (1997) for more details. For our purpose, we simply define a language as a set of sentences as described above. From a positivist standpoint each sentence of a language has an inherent truth value; in contrast, M&S languages are neither true nor false and can only be evaluated with respect to a model. That is to say that a language is said to be satisfiable under a model if and only if every sentence of the language can be evaluated as true under that model. Recall that a model is the ordered pair of symbols (constant, function and relation) and their interpretation, in that sense each sentence of a language is *information* and the meaning of that information depends on its interpretation. Some sentences are not satisfiable under a model meaning it is impossible to evaluate whether they are true . Finally, a language can exist without its theory being first specified or a language can be directly specified from a theory. The points made here are valid either way. Note that each sentence of a language has to be evaluated separately which leads us to the introduction of a theory.
- *Theory and Interoperability:* A theory is a set of sentences. A theory of a model is the set of sentences of a language that are true under that model. Simply stated, given a set of information, a theory is the collection of information that is true for that model. In other words we can equate a theory with a set of *valid information* about a model . This is useful in M&S interoperability because it provides us a apparatus for automatically detecting mismatches in the interpretation of information between ambassador agents. Axioms are the minimum set of sentences necessary to generate all the sentences of a theory.

These basic concepts allow us to define how ambassador agents can interoperate and generate a common understanding of the information they are sharing in order to generate a common solution to a given problem. The additional constraint that we put on ambassador agents is that they interoperate in a consistent manner and that the result of their interoperation is consistent. That is to say that agents must be able to interpret a language from another agent and be capable to satisfy or reject each sentence of that language. This information exchange between agents should not result in any contradictions overall (a sentence being both true and false at the same time). We propose the following high level algorithm for the interoperability of agent languages

```
1. Formulate the problem of interest as a model
   (the interoperability model)
2. For each agent
      a. Identify the theory of its language under the
         interoperability model
      b. If two or more theories intersect apply Robinson
         Consistency Theorem
      c. If there are contradictions or theories do not
         intersect apply Łoś' Theorem
3. End
```

This high level algorithm is a simplification of a complex formal, and in most cases engineering, process and relies on two theorems that we describe and discuss rather than define in its entirety. The interested reader is again encouraged to consult Weiss and D'Mello (1997). Robinson Consistency Theorem is one of the most fundamental findings in mathematical logic and simply states that the union of two theories is satisfiable under a model if and only their intersection is consistent. Looking at step 2.b, there are several ways to ensure that this is done such that the theorem is not violated. This discussion is out of scope for this article but the point remains that this step can be done formally. Łoś Theorem generalizes the idea of expanding a universe through the Cartesian product of other universes and filters. The Cartesian product provides a larger set of data and information that must be filtered out consistently in order to obtain the set of information that is the theory of the interoperability model. The selection of filters is also outside of the scope of this paper but this can also be done formally. Whenever these two theorems are not violated we can guarantee consistency in the solution provided and a consistent execution – or orchestration – of the solution.

We can now map each of the steps of the proposed algorithm to the steps of identifying, selecting, composing and orchestrating a set of solutions represented by a set of ambassador agents:

- *Formulate the problem of interest as a model and identify the theory of its language under that model:* Each agent is a ambassador for a model or a simulation both of which are theories of some modeling question. The interoperability model represents the universe under which the agent language has to be satisfied. It means that identifying a solution is equivalent to identifying any part of an agent language that is a theory of the interoperability model. The composition of these solutions represents the theory of the interoperability model. The composition of these solutions might mean to take the union of all languages that contain applicable theories-it is important to note we cannot take the union of all theories since each theory is only true with respect to its model. It also means that Robinson Consistency Theorem applies, which leads us to the selection part.
- *Apply Robinson Consistency Theorem:* If more than one agent generates a solution their languages intersect and they also intersect with the interoperability model's language. According to Robinson Consistency Theorem, the solution generated by the union of these solutions is only valid if and only if it is consistent with respect to every agent. Consequently, the selected solution is the one that guarantees that Robinson Consistency Theorem is not violated. Practically it means selecting only those solutions that do not intersect with any other agent as a naïve approach and avoiding statements in the intersection in the case where solution do intersect with one another. More intelligent approaches can be used but are out of the scope of this paper. In general, Łoś Theorem can be used to compose solutions. The problem is that filters have to carefully specified (Łoś Theorem tells us how to do this) to remain consistent and new axioms have to be added to avoid emergence. Emergence is often observed when there are sentences that are not satisfiable under the interoperability model. These sentences cannot simply be ignored as they violate the consistency under interoperability constraint mentioned earlier.
- *Use Łoś Theorem to orchestrate composition:* Once the filters are designed and the solution is composed from different theories in different agents, orchestration is the sequence in which the filters are activated and deactivated. Orchestration should result in a consistent execution such that each sentence generated is satisfiable under the interoperability model and there is no other sentence that contradicts it under the interoperability model. This latter is extremely important especially when the sentences relate to a function because of the cascading effects that are involved in functions (they change things that might change other things) and the cost of tracking the effects. A contradiction in this case might be fatal in the sense that the theory might no longer hold under the model and it would be virtually impossible to tell in non-trivial cases.

The most powerful idea of model theory is that truth is relative. Each ambassador agent represents its internal truth. When that truth is exposed to another model such as the interoperability model it becomes information that has to be evaluated again which makes it difficult in practice to design models with such formality. However, with Model Theory, it is clear that absent a model of the language and most importantly absent an axiomatic structure, formal interoperability is not possible. Engineered interoperability as we practice it today will work only for trivial cases even if to us they appear to be non-trivial. Results from Model Theory also show that models can be interoperable if they are designed to be interoperable. Łoś theorem helps us generate interoperability models from existing models but this should be the exception and not the norm as it is today. In the next section we present the implications of Model Theory, Robinson Consistency Theory and Łoś Theorem for agent languages. These implications are expressed in the context of interoperability i.e. what features should the agent language support in order to support interoperability. We will use the Levels of Conceptual Interoperability Model (LCIM) (Tolk et al. 2008) as a foundational basis for the discussion in the next section.

4 IMPLICATIONS FOR AGENT LANGUAGES

In terms of interoperability, Model Theory provides an insight that somehow M&S is about theory generation and interoperability is the subset of M&S that studies how to generate theories from models. In the language of M&S, Verification and Validity as well as composability are subsets of M&S that are dealing with aspects of theory generation. Interoperability models such as the LCIM are ways to ensure that the theory can be generated and issues like satisfiability and consistency are addressed. We propose the following requirements for agent languages to be interoperable not as an outcome of putting models together but by design. The LCIM has seven levels:

- *Level 0:* Stand-alone systems have *no interoperability*. At this level the agent language is a closed theory (a theory that has all of its consequences). Please recall that a theory is a subset of the language that is true under a given model.
- *Level 1:* On the level of technical interoperability, a *communication protocol* exists for exchanging data between participating systems. On this level, a communication infrastructure is established allowing systems to exchange bits and bytes, and the underlying networks and protocols are unambiguously defined. At this level the agent language must have a Finite State Machine (FSM) representation. Technical interoperability would simply mean that every agent language represents a theory under some computer model. In practice, this model could be a protocol such as TCP/IP or HTTP. The key observation is that all theories would be valid under one model or another. For instance one could say that models are interoperable under HTTP but not FTP, i.e., the technical language is a theory under HTTP but not under FTP. As a reminder, it means that every sentence of the language is a sentence of HTTP (true in HTTP) and every sentence is simply a sentence in FTP (not satisfiable under FTP).
- *Level 2:* The syntactic interoperability level introduces a common structure to exchange information; i.e., a *common data format* is applied. On this level, a common protocol to structure the data is used; the format of the information exchange is unambiguously defined. This layer defines structure. In order to achieve this level agent language must be able to provide an interpretation function that transforms constants into constants, functions into functions and relations into relations. Syntactic interoperability is achieved if the agents share a common interpretation function on constants, functions and relations. In terms of the LCIM, syntactic interoperability deals with data formats and by data it means constants. This is a very limiting view of data, one that we extend for agent languages to functions and relations. It is also important to note that syntactic interoperability only requires the alignment of languages or an equivalence class of languages and does not require a model under which the languages can become theories. This is done at the next level.
- *Level 3:* If a common information exchange reference model is used, the level of semantic interoperability is reached. On this level, the *meaning of the data* is shared; the content of the information exchange requests are unambiguously defined. This layer defines (word) meaning. In

terms of agent languages, semantic interoperability can be supported if the axiomatic structure of the language is provided. Semantic interoperability means that the language of the common reference model can be generated for the axiomatic structure. It is important to note that meaning here is interpreted as the ability to generate the language and not the ability to generate a theory of the language. The theory part is covered in the next levels.

- *Level 4:* Pragmatic interoperability is reached when the interoperating systems are aware of the *methods and procedures* that each system is employing. In other words, the use of the data – or the context of its application – is understood by the participating systems; the context in which the information is exchanged is unambiguously defined. This layer puts the (word) meaning into context. This is accomplished by relating terms describing input and output data with terms describing the methods and procedures. In terms of agent languages, the axiomatic structure must be able to generate a theory of the interoperability model (the common reference model), i.e., every sentence of the agent language must be a sentence of the interoperability model.
- *Level 5:* As a system operates on data over time, the state of that system will change, and this includes the *assumptions and constraints* that affect its data interchange. If systems have attained dynamic interoperability, they are able to comprehend the state changes that occur in the assumptions and constraints that each is making over time, and they are able to take advantage of those changes. When interested specifically in the effects of operations, this becomes increasingly important; the effect of the information exchange within the participating systems is unambiguously defined. In this level the order in which the sentences of the theory generated from the axiomatic structure is important. Dynamic interoperability implies that the sentences of the respective theories either appear in the same order or deliver the same results at defined synchronization points.
- *Level 6:* Finally, if the conceptual model – i.e., the assumptions and constraints of the meaningful abstraction of reality – are aligned, the highest level of interoperability is reached: conceptual interoperability. This requires that conceptual models are documented based on engineering methods enabling their interpretation and evaluation by other engineers. The conceptual alignment implies that agent languages have to specify the model for which they are a theory. Conceptual alignment is the establishment of equivalence between the models of the languages.

The requirements specified here are preliminary and will be examined in depth in future publications but in order to support interoperability, the bottom up approach is easier from an engineering standpoint but extremely hard to track from a consistency standpoint. We do not foresee that all models in the future have to be completely specified as formalisms but we do expect that some large parts of the model will be. It is important to note that simulations are already formally specified as computer languages, it is time for models to follow suit.

5 SUMMARY

This paper established the necessity to express not only implementation details but also the underlying conceptualization in order to be able to decide composability of models and interoperability of simulations in the light of a research question. The paper continued by introducing quadruples to describe data elements by specifying and communicating definition, identification, representation and permissible values. Extending this data elements properties that establish propertied concepts that can be associated with other concepts allowed to express information exchange models as trees that allowed to unambiguously express multi-resolution, multi-scope, and multi-structure challenges. Using Petri nets, constraints regarding the timely availability of information exchange elements can be modeled and expressed mathematically as well. To make the internal structures transparent, models can be treated as production systems, which can be expressed as a formal language. Having reached this understanding, that a model can be transparently represented by a formal language, allows the application of model theory, and with it the application of many already proven principles that will allow better decisions regarding the composition of model and coupling of simulations in support of answering a given research question. Many current standardization efforts fall short in comparison and will likely result in suboptimal or even wrong results.

This paper presents the case for a formal approach on model descriptions based on formal languages to allow the rigorous application of model theoretic insights. Using formal languages allows to leverage different approaches, methods, and paradigms and generate a coherent view of insights already available in the body of knowledge. Furthermore, this allows to derive new knowledge for application of M&S to provide solutions in heterogeneous application domains. In other words, it provides the means for common communication of insights: a language for M&S science, which is no longer limited to M&S engineering providing master-tailored solutions for application domains, but that allows knowledge extraction and transfer to the benefit of M&S professionals.

REFERENCES

Ahn, J. H., M. G. Seok, C. H. Sung, and T. G. Kim. 2010. "Hierarchical Federation Composition for Information Hiding in HLA-Based Distributed Simulation." In *Proceedings of the 2010 IEEE/ACM 14th International Symposium on Distributed Simulation and Real Time Applications (DS-RT '10)*, 223-226. Washington, DC: IEEE Computer Society.

ISO. 2004. *ISO/IEC 11179: Information Technology – Metadata registries (MDR): Part 1: Framework; Part 3: Registry metamodel and basic attributes; Part 4: Formulation of data definition.* International Standards Organization. Accessed May 1. http://metadata-stds.org/11179.

ISO. 2005. *ISO/IEC 11179: Information Technology – Metadata registries (MDR): Part 2: Classification; Part 5: Naming and identification principles; Part 6: Registration.* International Standards Organization. Accessed May 1. http://metadata-stds.org/11179.

ISO. 2010. *ISO/IEC 19763: Information Technology – Metamodel framework for interoperability (MFI): Part 3: Metamodel for ontology registration.* International Standards Organization. Accessed May 1. http://metadata-stds.org/19763/index.html.

Logan, B., and G. Theodoropoulos. 2001. "The Distributed Simulation of Multi-Agent Systems." In *Proceedings of the IEEE*, Special Issue on Agent-Oriented Software Approaches in Distributed Modeling and Simulation, 89 (2):174–185.

Overstreet, C. M., and R. E. Nance. 1985. "A Specification Language to Assist in Analysis of Discrete Event Simulation Models." *Communications of the ACM* 28(2):190-201.

Page, E. H., and J. M. Opper. 1999. "Observations on the Complexity of Composable Simulation." In *Proceedings of the 1999 Winter Simulation Conference*, edited by P.A. Farrington, H.B. Nembhard, D.T. Sturrock, and G.W. Evans. Vol. I, 553-560. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Tolk, A. 2004. "XML Mediation Services Utilizing Model Based Data Management." In *Proceedings of the 2004 Winter Simulation Conference*, edited by R.G. Ingalls, M.D. Rossetti, J.S. Smith, and B.A. Peters, 1476-1484. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Tolk, A., and S. Y. Diallo. 2005. "Model-based Data Engineering for Web Services." *IEEE Internet Computing* 9(4):65-70.

Tolk, A., and S. Y. Diallo. 2010. "Using a Formal Approach to Simulation Interoperability to Specify Languages for Ambassador Agents." In *Proceedings of 2010 Winter Simulation Conference*, edited by B. Johansson, S. Jain, J. Montoya-Torres, J. Hugan, and E. Yücesan, 359-370. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Tolk, A., C. D. Turnitsa, and S. Y. Diallo. 2008. "Implied Ontological Representation within the Levels of Conceptual Interoperability Model," *Intelligent Decision Technologies* 2(1):3-19.

Tolk, A., S. Y. Diallo, J. J. Padilla, and C. D. Turnitsa. 2011. "How is M&S Interoperability Different from other Interoperability Domains?" In *Proceedings of Spring Simulation Interoperability Workshop*, 11S-SIW-008. Orlando, FL: Simulation Interoperability Standard Organization.

Weiss, W., and C. D'Mello .1997. *Fundamentals of Model Theory.* Department of Mathematics, University of Toronto, Canada. Accessed May 1. http://www.math.toronto.edu/weiss/model_theory.pdf.

Yilmaz, L., and S. Paspulleti. 2005. "Toward a Meta-Level Framework for Agent-Supported Interoperation of Defense Simulations." *Journal of Defense Modeling and Simulation* 2(3):161–175.

Yilmaz, L. and T. I. Ören. 2004. "Exploring Agent-Supported Simulation Brokering on the Semantic Web: Foundations for a Dynamic Composability Approach." In *Proceedings of the 2004 Winter Simulation Conference*, edited by R.G. Ingalls, M.D. Rosetti, J.S. Smith, and B.A. Peters, 766-773. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zeigler, B. P. 1986. "Toward a Simulation Methodology for Variable Structure Modeling." In *Modeling and Simulation Methodology in the Artificial Intelligence Era,* edited by M. S. Elzas, T. I. Ören, and B. P. Zeigler, Amsterdam: Elsevier North Holland Press.

AUTHOR BIOGRAPHIES

ANDREAS TOLK is Professor of Engineering Management and Systems Engineering at Old Dominion University. He is also affiliated with the Virginia Modeling Analysis and Simulation Center. He holds a M.S. and Ph.D. in Computer Science from the University of the Federal Armed Forces in Munich, Germany. His email address is atolk@odu.edu.

SAIKOU Y. DIALLO is Research Assistant Professor at the Virginia Modeling, Analysis and Simulation Center at Old Dominion University. He received his M.S and Ph.D. in Modeling and Simulation from Old Dominion University. His email address is sdiallo@odu.edu.

HEBER HERENCIA-ZAPANA is a Research Associate at the National Institute for Aerospace where his research focuses on applying formal methods to M&S Theory. He received his Ph.D. in Electrical and Computer Engineering from Old Dominion University. His email is Heber.Herencia-Zapana@nianet.org.

JOSE J. PADILLA is Research Assistant Professor at the Virginia Modeling, Analysis and Simulation Center at Old Dominion University. He received his Ph.D. in Engineering Management from Old Dominion University. His email is jpadilla@odu.edu.