

USING A FORMAL APPROACH TO SIMULATION INTEROPERABILITY TO SPECIFY LANGUAGES FOR AMBASSADOR AGENTS

Andreas Tolk

Saikou Y. Diallo

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

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

ABSTRACT

Ambassador agents represent simulation services that are candidates to contribute to the solution of a problem. They need to know and express enough about the simulations to negotiate with other ambassador agents if the represented simulation systems can be composed to contribute to the solution. A formal approach to simulation interoperability based on set theory and data modeling theory was developed. The formal model of data in M&S capturing possible representations of real or imagined things in the world including definitions for existential and transformational dependencies is presented. Existential dependencies capture the relationships within a model while transformational dependencies capture the relationships between interactions with a model. These definitions are used to formally specify interoperation, the ability to exchange information, as a necessary condition for interoperability. The elements needed for a language needed for ambassador agents are derived using the formal approach to interoperability.

1 INTRODUCTION

This paper is composing three research domains supported by the authors and merges them into a common application, namely a theory, an engineering method, and the use of agents.

- The topic of composability of models and interoperability of simulations resulted in the development of the *Levels of Conceptual Interoperability Model* (LCIM), which was first introduced in (Tolk and Muguira 2003) and has been continuously evolved using feedback from other experts in various application domains to its current form, as described in (Tolk et al. 2010).
- Using the theoretic insights from this model, the engineering method *Model-based Data Engineering* (MBDE) was developed to support heuristics that can be applied by systems engineers and federation developers to achieve the goal of composing model-based solutions (Tolk and Diallo 2005).
- By developing a formal approach to simulation interoperability and capturing the formalism using mathematically precise terms, the development of profiles is supported that now allows the application of *Agent-Directed Simulation* (ADS) to support the composition. Using the meta-data of the profile, agents become ambassadors for the model-based solution and can negotiate their composition with other solutions in support of a research question.

This paper uses many of the groundbreaking ideas on agent-based mediated solutions that so far were missing a common component of a formal approach to interoperability. This component should allow the profiling of a model-based solution, the measurement of the semantic distance between alternatives as well as the measurement of the distance between required capability and provided functionality, and a

measurement of the feasibility of a proposed composition. The recent doctoral research conducted at the Old Dominion University (Diallo 2010) provides the means to close these gaps and enable the use of MBDE based on the LCIM conducted by ADS.

To introduce this composition of ideas, the paper is structured as follows: Section 2 introduces the general challenges that have to be solved when composing model-based solutions and that ambassador agents have to be able to cope with when they are applied. Section 3 synthesizes several recommended solutions on how to use agents in support of interoperability and composability tasks and how to ensure the necessary introspective. Section 4 describes the main findings of our formal approach to simulation interoperability and how this can be used to improve current solutions and close several of the identified gaps. Section 5 finally makes some recommendations regarding a specifying a language for ambassador agents.

2 CHALLENGES TO ADDRESS WHEN COMPOSING MODEL-BASED SOLUTIONS

The objective of this paper is synthesizing eminent ideas in the agent-directed community with latest research results of the author enabling agents to support the composition of model-based solution in support of a research question. A model-based solution is an executable implementation of a model that represents the formal specification of a conceptualization. Simulations qualify as model-based solutions and several standards support the information exchange between simulations, the challenge remains composing the model. When several model-based solutions are composed, several activities have to be aligned to ensure that the result is a meaningful composition. While it seems logical that the question of meaningful composition requires knowing what the composition shall be used for, many practical applications do not answer the question sufficiently, at least not formally. What is needed to decide whether a composition is meaningful is a formal representation of the model-based solution capabilities as well as of the research question. A formalism expressed in mathematical terms allows defining profiles representing the important characteristics for the required capability (research question) as well as the available functionality (model-based solutions).

Once the research question as well as the model-based solution capabilities is formulated, the following four tasks can be supported (Tolk et al. 2010):

1. *Identify* applicable solutions that can contribute to answering the research question;
2. *Select* the best solution or mix of solution if alternatives are available;
3. *Compose* the selected model-based solutions;
4. *Orchestrate* the execution of the composition of the model-based solutions.

Most current simulation interoperability standards are insufficient regarding support of these four tasks. As Tolk, Diallo and Turnitsa (2008) point out many of the current approaches focus exclusively on information exchange to support the federation of solutions without providing the necessary introspective. The simulation interoperability standards support this idea as well: IEEE 1278 on Distributed Interactive Simulation (DIS) defines Protocol Data Units (PDUs) that are exchanged between participating simulators and simulation to communicate events and their effects, such as collisions, duels, or the exchange of operationally relevant data. IEEE 1516 on the High Level Architecture (HLA) is a little bit more flexible, as the information to be exchanged itself is not standardized but only how to structure the data, namely in form of the Object Model Template (OMT) and provided via the interfaces between federate and the Runtime Infrastructure (RTI), but the focus remains on information that can be exchanged with a system.

The underlying assumption for the success of such efforts is that all models are embedded into and derived from one common reality that can be expressed by some kind of common *Übermodel* representing the truth completely and free of contradictions. Consequently, if such a model existed all models could be derived from this common model by pruning and aggregating, or simplification and aggregation. The DIS PDUs represent messages that are sufficient to be exchanged between entities of

this common model, if they are on platform level. The HLA OMT describes the information more generally, as the entities may be aggregates, but the idea is that we can define the transitive hull of each model as a subset of the common model of reality and define the information exchange needs to allow for communication between the models as well as for synchronization and alignments of entities and events that take place in more than one model. Modeling is therefore understood as identifying the necessary subset of a common truth that underlies all resulting model as a common denominator.

The authors are convinced that the idea that all simulations share a common reality is neither achievable nor desirable. The idea may have been driven by the physical models mimicking system behavior of physical systems. Many validation papers also support related worldviews by using the system of the real world as the reference systems that needs to be the yardstick for fidelity and accuracy of a modeled solution of this system. Even the Discrete Event System Specification (DEVS) support such a viewpoint. So what is the problem? As long as we are in the physical world in which measures of well understood physical effects are the basis for the model, there is no problem, but that is only a small fraction of what simulation can be used for. In most other domain, modeling is not just the process of simplification and abstraction to support mimicking effects, it is a creative process that results in a consistent and precise formulation of a particular world view in a model.

This view is supported by other research as well. As stated by Reynolds (2009), there are two application categories for modeling and simulation (M&S): *problem solving* and *gaining insight*. If a problem as well understood and solving algorithms to deal with challenges are available, the model captures these algorithms and the simulation system is used for computational reproduction of the solution. Training and analysis are typical application domains. In this application category, validation and verification are applied to ensure that the known solution is reproduced accordingly by the model and implemented correctly by the simulation. When M&S is used to gain insight, another paradigm is applied. In this category, models are used to understand what the main parameters are, how they interact, if emerging behavior can be observed, etc. M&S is applied to explore possible connections and relations. The main role of the model is to help to think about a problem. They do not implement a solution, but they are used to evaluate several solutions one them may be applicable, or several of them may help to shed light on parts of the solution. In particular in new domains that are not dominated by physical and technical effects, such as human, social, cultural, and behavioral (HSCB) modeling challenges (Tolk 2009, Numrich and Tolk 2010) are characterized by not necessarily having one correct solution. Human and social sciences are not 'exact' sciences like physics. Furthermore, today even these exact sciences agree with the post-positivistic worldview that it is not possible to agree on one truth, and not even language or logic are perceived to be means of unambiguous representations of truth.

Yilmaz and Oren (2004) introduced the ideas of multi-models and multi-simulation to cope with this challenge. They define a multi-model as a modular model that subsumes multiple sub-models that together constitute the behavior of a complex multi-phased process and as such encapsulating several aspects of reality. Multi-simulation includes simulation based on multi-models, multi-aspect models, and multistage models. The common factor is that it does not need to be assumed that reality is or can be captured by just one model. This general understanding is also formulated in Bankes (1993) essay on exploratory modeling and needs to be rediscovered to address our current challenges.

Nonetheless, while it is not generally possible to capture truth, the truth captured in each model must be represented consistently. Furthermore, if several models are composed, no contradictions shall be introduced by doing so. For the challenges to be addressed for the composition of models, these leads to the following general observations:

1. Each model represents its own view of reality. This must be a consistent logical representation of research questions that can be dealt with in the model (questions that can be answered with TRUE or FALSE). The view of a model is the sum of all answers that can be generated. The simulation is an execution of this model, which may add additional constraints derived from the nature of the chosen implementation.

2. Composability of models and interoperability of simulation is not a characteristic of the model or simulation. It can only be discussed in the light of a research question to be answered representing the sum of answers needed to be provided by the target system. Two models are composable if they answer questions in the scope of the research question consistently. Two simulations are interoperable if their execution in support of experiments to answer the research question does not lead to contradictions in their results.
3. As no common unambiguously expressible reality or viewpoint can be assumed, the research questions assumes a pivotal role and shall be expressed using the same formal approach as used to express the capabilities of the model-based solution. We assume that two models can be composed under a research questions if their relevant capabilities can be mapped to an aggregated and pruned version of the research questions.

In other words: consistency of models means consistency of answers provided to the research question. If only one model provides an answer, this is consistent. If more than one model provides an answer, the answers provided must be the same. It has been observed that this consistency constraint is valid for all kinds of compositions: multi-models, multi-aspect models, and multi-stage models. Yilmaz and Tolk (2008) provide first insights on how to apply these insights in the domain of ADS.

The formal approach to interoperability must take all ideas presented in this section into account. Using the formal approach to define profiles for model-based solutions and the research questions to support a language for agents to negotiate contributions and evaluate alternative solutions then becomes feasible.

3 USING AGENTS AS AMBASSADORS FOR MODEL-BASED SOLUTIONS

The idea to use agents to construct a meta-simulation layer framework is not new. Yilmaz (2007) summarizes the state of the art of how agents can be used to support interoperability and composability. He introduces a family of agents that are needed and constitute an agent organization, namely mediators, facilitators, brokers, and matchmakers that provide functionality needed to perform data and service management, alignment, and transformation. One of the main contributions of this approach is decoupling the simulation concepts from the intricate details, or – in other words – separation of conceptual model and simulation detail. The service infrastructure provides the standards necessary for the simulations of the society of simulation to exchange information; it provides the means for interoperability. The agent-based interoperation framework supports selection and composition of models; it provides the means for composability. Both layers are needed to improve dynamic interoperability. Figure 1 shows the meta-simulation layer, as it was first documented by Yilmaz and Paspuletti (2005).

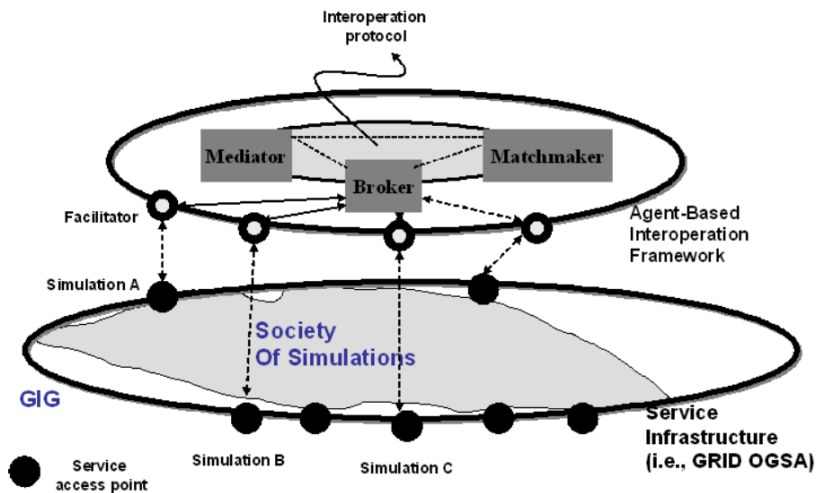


Figure 1: A Meta-simulation Layer (Yilmaz and Paspuletti 2005)

The roles of the four agent types can be described as follows: The facilitator agent acts as a gateway between the simulation infrastructure and the agent organization that orchestrates the simulation interoperation. The mediator agent is responsible for converting simulation content to/from a common reference model used as the common interpretation of data exchange on the conceptual level. Broker and matchmaker agents select and update collaboratively the producers and consumers of data in a common workflow as cooperative partners. The idea of introspection, the seamless and transparent update of behavior of the simulation system – who is doing what, when, where, why, and how – by representing agents, plays a pivotal role and was first published by Yilmaz (2004) and updated by Yilmaz and Tolk (2006). It was recently extended in support of autonomic introspective simulation Systems by Yilmaz and Mitchell (2008).

In order for these agents to support challenges when composing model-based solutions, they need to know about assumptions, constraints, and implementation details of the simulations that make up the society of simulations. However, what is also needed for them to know are the assumptions, constraints, and implementation requirements of the desired solution, or in academic terms: the research or modeling question is needed as a necessary requirement to decide about the composability of two or more model-based solutions. Composability is not a characteristic of a model-based solution; it can only be decided in the context of a modeling question! Yilmaz and Oren (2004) identify the following areas to facilitate formal composability:

- A *formalism* providing the basis of annotating model-based solutions to unambiguously capture assumptions, constraints, and implementation details;
- An *ontology* based on the formalism capturing the necessary facets of the model-based solutions. In this context, the ontology can be understood as the formal specification of the conceptualization underlying the model-based solution.
- *Profiles* for each model-based solution describing the individual characteristics and forming the basis for the comparison of provided functionality by the solution, alternatives provided by other solutions, and required functionality by the modeling question.

Yilmaz and Oren also include tools in their list. The paper by Yilmaz and Oren (2004) contributed significantly to the discussion on semantic composability challenges, although it only mentioned the concepts without going into details of the formalism and the mathematical means. The formalism provided in section 4 shall close this gap. How the ontology should look like is specified in more detail by Yilmaz (2007a). The main contribution of this paper (Yilmaz 2007a) with regard to the topic of the current work, however, is not the general ontology, but the idea to apply graph-theory to questions of conceptual interoperability. Yilmaz (2007a) introduces meta-level graphs to represent the model-based solutions, their dependencies, and their derivation. He is using this approach to align the required and the available content, as depicted by Tolk and Diallo (2005). He also introduces graph similarity metrics to measure the ‘distance’ between two comparable solutions or between a solution and the desired functionality. He introduces a three-level abstraction hierarchy in which simulations participating in a federation on the lowest level are specified by meta-models on the middle level. These meta-models are building the nodes of the meta-level graph on the highest level that represents the federation and how the meta-models exchange information with each other.

The research of the authors of this paper shows that, as mentioned before in this paper, composability requires a representation of the modeling or research question to decide if two model-based solutions can be composed to provide the necessary functionality in consistent form. Therefore, the meta-level graph suggested by Yilmaz (2007a) should be replaced by such a model of the modeling questions. The formalism presented in the following section serves then not only as a guide on how to represent the individual model-based solutions, but is also the frame for the distance metrics shown to be necessary. This will support the definition of a language for ambassador agents.

4 A FORMAL APPROACH TO SIMULATION INTEROPERABILITY

The formalism presented in this section was developed within the research described in (Diallo 2010). Interoperability is defined as “*the ability of two or more systems or components to exchange information and to use the information that has been exchanged*” (IEEE 1990). This definition reflects the common understanding that interoperability necessarily involves information being exchanged, which implies a technical framework to support the exchange and information being used, which means that a common understanding of the information is shared by the systems. A closer inspection of this definition leads to the following questions:

- What is information for a system?
- What is useful data for a system?
- What happens when there is no state change or output being produced?

While humans have a cognitive process and a language to grasp, formulate and exchange information, computers can only deal with bits and bytes in a context free environment. In general, computer agents that are implemented on a digital computer deal with data and use relationships between data to infer information. Ackoff (1989) provides a framework to distinguish between data (factual information), information (data in context) and knowledge (procedural application of information). This framework can be applied for computer agents as well: data are simply a set of symbols, facts or figures, while information is data that are processed to be useful and provide answers to questions such as “who, what, where, and when” (Ackoff 1989). As each computer implementation of an agent is using logic and computational functions, we define data as a set of symbols and information as a combination of symbols to form logical statements that are true or false, including those that are always true or always false. Ackoff (1989) notes that information must be processed to be useful which renders our definition somehow incomplete so far as it does not provide a way to determine what is useful.

Information as defined above requires that data are processed and evaluated as being useful once it is provided. As such, knowledge introduces the procedural component and allows to describe the “how and why” of tasks as well. In order for a computer agent to process data and determine whether it is useful, it is necessary to first establish the state it is in when it receives data and whether the data is complete and can be processed. Intuitively, data are useful if their use result in a change in state in the receiving system or an output being produced.

If no state change happens in the receiving system, there are two possible reasons for this occurrence, namely incomplete data and useless data. Data can be useful but not usable in general or in the state at which it is received. Tolk et al. (2007) introduce the notion of weak and strong composites to describe data that have to be provided as a whole in order to be interpretable. In this case, the computer agent needs to either provide additional data internally or seek data from systems. Useless data are data that are not interpretable by the computer agent either because a match does not exist or there is no state/input pairing for it. If a match does not exist, it can be said that the computer agent is unaware of the data or simply that it cannot use the data. If on the other hand there is no state input pairing, it means that there is a worldview mismatch between the sending and receiving system. While this problem is important it is outside the scope of this article. The interested reader is referred to (Padilla 2010) for additional details of this new application paradigm of M&S to process and produce knowledge.

The three questions above and the possible explanations while intuitive point to the need for a formal way to describe data that enables us to determine whether these questions can be answered in general and if not propose heuristics that approximate a solution.

In data modeling theory, the three terms entity, property, and value domain constitute the core of a data model. For M&S, we need to add the notion of a domain which is similar to the traditional view of the value domain but is generalized to encompass the domain of discourse. Finally, the term element is

used to mean anything real or imagined that can be described and/or observed. The five terms are defined as follows:

- *Elements* are real or imaginary things.
- *Entities* are abstractions of elements. It is worth noting that by this definition any abstraction including processes and relationships are considered entities.
- *Properties* are the characteristics of an element.
- *Symbols* are the representations of elements. Symbols can be numbers, strings, images, text or a combination of symbols.
- A *domain* is a collection of unique symbols. The domain is the set of elements that belong to a context. Every element is uniquely identifiable within a given domain.

It is important to note that elements are used to separate the notions of entities and properties and how they are related. Elements can be modeled as entities or properties and then related to form the entity-property relationship. Given these definitions, let us first formally capture a conceptualization of the referent as an element.

Definition 1 Let S be the set of elements, Ω the set of entities, Π the set of properties, V the set of symbols. A conceptualization S is formally captured by the categorization of elements into entities, properties or symbols. Formally a conceptualization is a partial function F such that:

$$F(S) = \begin{cases} \Omega & \text{if } S \text{ is an entity} \\ \Pi & \text{if } S \text{ is a property} \\ V & \text{if } S \text{ is a symbol} \end{cases}$$

An element within S is an entity, a property or a symbol, otherwise stated the three sets are mutually disjoint. Given a set S of elements and a non-empty set of domains Δ , every element in S is associated with a domain. Mathematically, we define the tuples:

- α is a subset of $\Omega \times \Delta$, the Cartesian product of entities and domains
- β is a subset of $\Pi \times \Delta$, the Cartesian product of properties and domains
- γ is a subset of $V \times \Delta$, the Cartesian product of symbols and domains

Given Δ the set of domains, we define the relation ρ as the subset of $\Delta \times \Delta$ the Cartesian product of domains. A model M of a conceptualization S denoted M_S is the relation $(\alpha, \beta, \gamma, \rho)$.

It is worth mentioning that the relation ρ captures the relationship between entities and entities, entities and properties, entities and symbols, properties, properties and symbols and symbols and symbols. In addition ρ captures relationships between relationships if one considers that all the relationships in definition 1 above are elements that have as domain a subset of $\Delta \times \Delta$ and therefore abide by the previous definitions. The relation ρ is a graph with vertices Δ . This allows capturing all current model paradigms, from system dynamics in which only properties are used to capture system changes, to complex model federations, where relations over related components of groups of parameters may need to be defined. Using definition 1, we can now capture the model of conceptualizations that are comparable and fulfill the requirements posit by Yilmaz and Oren (2004). The quadruple $(\alpha, \beta, \gamma, \rho)$ defines these models and supports similarity comparisons and metrics for alignments as requested in a generalized form that serves as a triple required in support of formal composability. To show the power of this formalism, we define two different categories of well known dependencies.

Definition 2 Let X, Y be sets of entities, properties or symbols with respective domains Δ_X and Δ_Y , Y is existential dependant on X denoted $X\Phi Y$ or $\Phi(X, Y)$ if the existence of Y implies the existence of X .

Definition 3 Let X, Y be sets of entities, properties or symbols with respective domains Δ_X and Δ_Y , Y is transformational dependant on X denoted $X\Theta Y$ or $\Theta(X, Y)$ if a change to Y implies a change to X .

Existential dependencies occur when two elements are related in such a way that one cannot exist without the other. While they may seem independent, they may be existentially connected in the model of the conceptualization. An example is an electric power line that uses a bridge. Both entities are separate entities, but when the bridge gets destroyed, the power line will be affected as well. A simple model may therefore connect both; another may model them in more detail as separated things. Transformational dependency generalizes this idea: if a change in one entity triggers also changes in the dependent one, we observe transformational dependency. Whenever two models are brought together, these dependencies need to be aligned, or strange, unintended and surprising effects will be observed when additional dependencies exists, or expected effects may not occur if expected dependencies do not exist.

A computer agent can support interoperability only if it can answer the three questions given at the beginning of this section. Further, the computer agent must be able to determine what is meaningful which means it has to operate at least at the semantic level of interoperability. The fundamental question that the computer agent must answer is therefore: *Given data, can I determine whether it is useful in answering the research question?* Diallo (2010) shows that in general a “yes/no” answer cannot be provided for this question. He furthermore shows that in general:

- A data model in M&S can be captured as relation between entities within a domain, properties within a domain, symbols within a domain and domains between themselves. A domain is simply a namespace in which entities, properties and symbols are uniquely identified.
- Data do not always exist independently from one another. The existence of one datum might be predicated on the existence of another datum (Existential dependency). Also, a change in one datum might necessitate a change in another datum (Transformational dependency). A computer has to be aware of both in order to answer the fundamental question posed earlier.
- There are two fundamental ways to capture the meaning of data. The first one is the use of terms or labels that intrinsically have meaning and the second is to relate terms to generate additional meaning through the relationships. Both approaches are used simultaneously in most cases (ontology for example) and when they are it is important to recognize that both are equally important. For a computer agent, the first approach reduces to a set of strings or symbols within a domain. The domain guarantees that each string is unique within its domain. The second approach is effectively a graph relating strings to one another. One can ignore the set of strings because a computer agent cannot derive the meaning of a string. Therefore, the fundamental question posed earlier reduces to the following: Given a graph, Can I determine whether it is useful in answering the research question? In order to answer this question, a computer agent has to compare the graph it receives with the research question. If the question is captured as a model, it is also a graph, which means that the computer agent must compare a graph with a least some part of another graph. This problem is known as the sub-graph isomorphism problem and it is NP-Complete.
- Two systems are interoperable if and only if their existential and transformational dependencies intersect and they are interoperable only over the intersection. Otherwise stated in terms of graphs, two systems are interoperable if and only if there is an isomorphism between the two.

Ambassador agents can now be used to implement a heuristic that approximates an answer to the research question, as the general answer is in NP-Complete. The agent must know the intersection of isomorphic classes between all candidate systems. In practice, it means that the agent must insure that only the selected system executes the part of the modeling question it is assigned. It also means that due to dependencies, a system might have to execute some part of the research question it was not assigned in order to answer the part of the question it was assigned. If this happen, the potential for conflict exist and

therefore the agent must either harmonize, de-conflict (ignore one or the other for example), synchronize or de-select (find another alternative). The next section presents an algorithm that takes into account the issues and questions raised in this section. The algorithm is computable which means that a computer agent tasked with identifying candidate systems can use it to determine which ones can answer parts or the whole of the research question.

5 TOWARDS A LANGUAGE FOR AMBASSADOR AGENTS

In general, the model of a conceptualization as resulting from definition 1 can be captured as a graph and the decision as to whether two graphs can be combined is NP-Complete. Therefore, as a result, to determine how models should be compared, we proposed the following heuristic. The heuristic requires that the modeling question be captured in the form of a model which is called the interoperability model. The Common Reference Model (CRM) is the part of the modeling question that can be answered by composing existing models. The interoperability model that captures the modeling question is mostly a human endeavor. The heuristic takes as input the interoperability model and a candidate model that is being evaluated for merging into a federation. The output is an approximate CRM or nothing if there is no CRM. If more than one model is required to answer a given part of the modeling question, the heuristic can be applied by replacing the interoperability model with the CRM previously defined and another candidate model. If more than one model is needed to answer the modeling question as a whole, the heuristic can be applied by replacing the interoperability model with the subset for which the candidate model is being evaluated.

The main approach in this heuristic is to first create an interoperability model representing the research question. Based on this model, a number of candidate models are selected to form a federation. The idea is that the federation of candidate models should replicate the interoperability model. There are two main possibilities to explore once these models are obtained. First, can a single model replicate the interoperability model? If the answer is yes, that model is selected and there is no need to federate. The next question is how many models are required to replicate the interoperability model. In order to answer that question, let's assume that the interoperability model can be partitioned into distinct but related parts. The question becomes then how many models are needed to replicate a given partition. The answer to that question in general is NP-Complete as shown in (Diallo 2010). However, if it is possible to approximate the answer for every partition provided they are small and manageable, it is possible to take the union of those models capable of replicating those partitions as the ones most capable of approximating the interoperability model.

Once the interoperability model is specified, the heuristic assumes a small partition is selected. The first goal is to determine whether the candidate model has some common domains with the interoperability model, if that is the case the domain along with its existential dependency set is selected to be a domain in the CRM. The reason for selecting the existential dependency set is to make sure that all possible relationships with elements are captured and to maintain the structural integrity of the candidate model. The domain might be independent in which case it is a trivial dependency. If no domains are common the candidate model is eliminated as it is impossible to ascertain in its current form whether it is capable of replicating the interoperability model. As a caveat, it is always possible to reengineer the candidate model or apply some series of transformation to arrive at the interoperability model, but the reader should think the reengineered model as the candidate model not the original one. Once the domains identified, for each domain, the heuristic looks for elements defined similarly across equivalent domains. The heuristic eliminates elements that do not belong to the same groupings. Elements within the same groupings and same domain along with their existential dependencies sets constitute the initial elements of the CRM. After groupings are identified, elements that differ in scope and resolution are eliminated as a modeling decision is required to resolve the issue. The heuristic purposefully does not provide a resolution as this should be done on a case by case basis. With this step, interoperation is now possible. In order to have interoperability, the transformational dependencies for each element have to be verified. If an element is independent or depends on an independent element in the CRM, it stays in the CRM. If the

element depends on a dependant element or it depends on an element outside the CRM it is removed. That is not to say that this element cannot be exchanged, it just means that an engineering process that ensures that the exchange does not lead to inconsistencies is required. As the candidate model needs additional information in order to replicate the parts or whole of the interoperability model those elements cannot be part of the CRM as-is. Figures 2 – 5 capture the four steps of this approach.

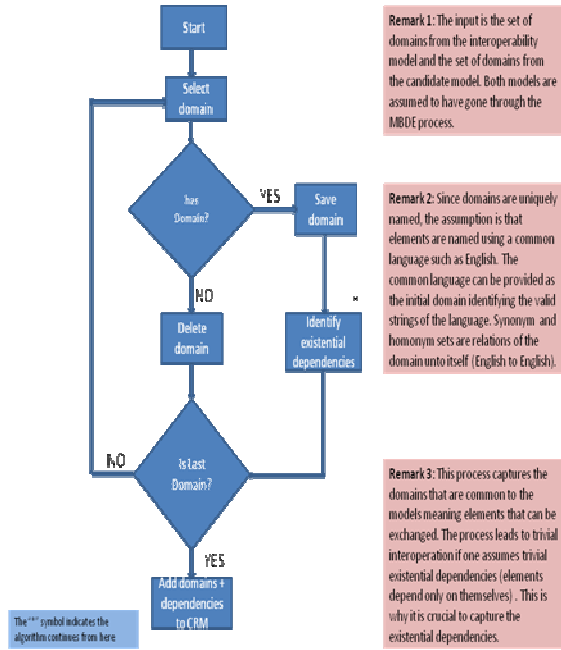


Figure 2: Heuristic Part 1

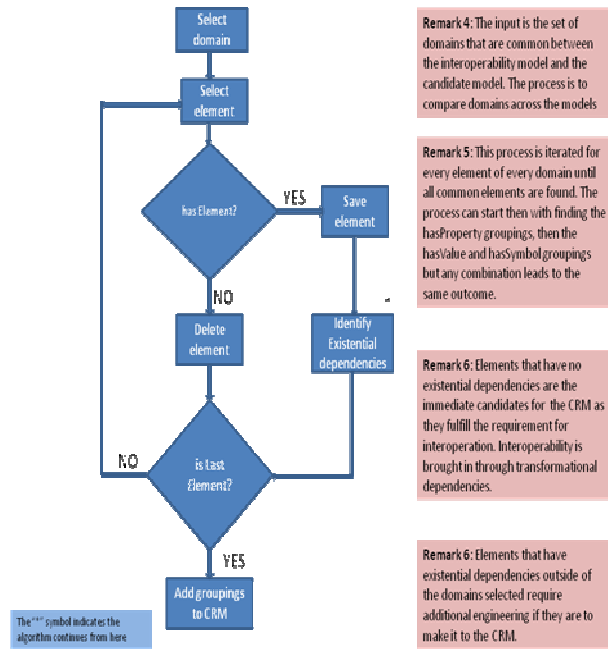


Figure 3: Heuristic Part 2

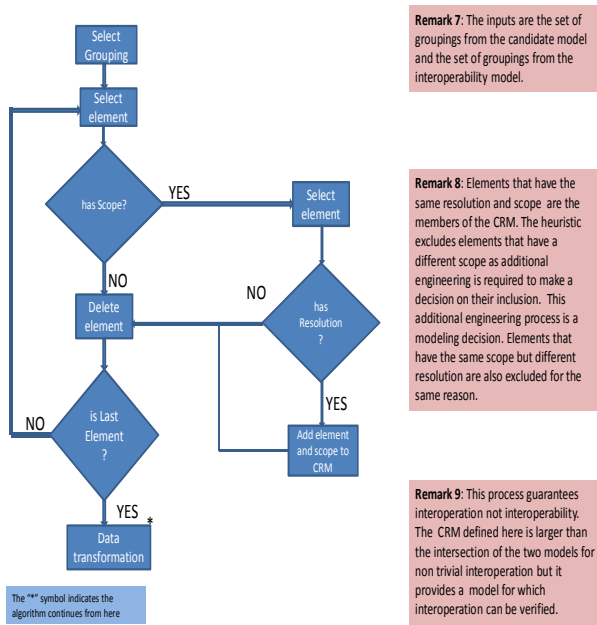


Figure 4: Heuristic Part 3

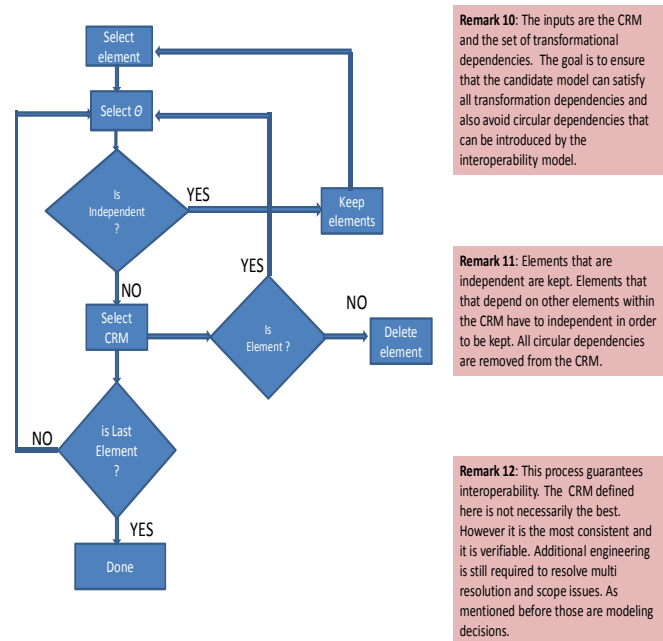


Figure 5: Heuristic Part 4

In order for ambassador agents to support this approach, they need to be able to express the models of the conceptualizations of the candidate systems, which are members of the society of simulations as defined by Yilmaz and Paspuetti (2005), as well as the research question – or interoperability model. As such, the language can be reduced to an algebraic expression over the quadruple $(\alpha, \beta, \gamma, \rho)$ and evaluate the alignment of existential and transformational dependencies.

6 SUMMARY

Within this paper, we merged former independent solution to provide a framework for agent-mediated as envisioned in visions provided nearly a decade ago (Yilmaz and Oren 2004). The formalism provided by Diallo (2010) allows creating profiles for the model-based solutions that can be used by ambassador agents to identify applicable solutions that can contribute to answering the research question, to select the best solution or mix of solution if alternatives are available, to compose the selected model-based solutions, and to orchestrate the execution thereof.

One side result of this research is that it allows extending the LCIM into an interoperability maturity model. The LCIM defines technical, syntactical, semantic, pragmatic, dynamic, and conceptual levels. The symbols defined in the formalism define the syntax, the relations of symbols with the domains and further specifications results in the semantic. The information exchange requirements for each provided model-based solution define the context of its use, hence define the pragmatics. The modes of the model-based solutions capture the dynamics. The assumptions and constraints for the whole system as well as on each lower level can now be expressed using the elements of the formalism. As they can be expressed in sets and graphs, measures as recommended in (Yilmaz 2007a) can now be applied to measure alignment of required capability and provided functionality as well as similarity of alternative solutions. This is topic of ongoing research.

REFERENCES

- Ackoff, R. 1989. From Data to Wisdom. *Journal of Applied Systems Analysis*, 16, 3-9
- Bankes, S. 1993. Exploratory Modeling for Policy Analysis. *Operations Research*, 41(3):435-449
- Diallo, S.Y. 2010. *Towards a Formal Theory of Interoperability*. Doctoral thesis at Old Dominion University, Frank Batten College of Engineering and Technology, Norfolk, VA
- IEEE. 1990. *A Compilation of IEEE Standard Computer Glossaries*. New York: IEEE Press
- IEEE. 2000. *IEEE Standard 1516 on High Level Architecture*. IEEE CS Press
- IEEE. 2002. *IEEE Standard 1278 on Distributive Interactive Simulation*. IEEE Press
- Numrich, S.K, and A. Tolk. 2010. Challenges for Human, Social, Cultural, and Behavioral Modeling. *SCS M&S Magazine* 1(1), January
- Padilla, J. J. 2010. *Towards a Theory of Understanding Within Problem Situations*. Doctoral thesis at Old Dominion University, Frank Batten College of Engineering and Technology, Norfolk, VA
- Tolk, A. 2009. New M&S Challenges derived from the NATO Research & Technology Organization (RTO) Systems Analysis Studies (SAS-071) Task Group on Analytical Tools for Irregular Warfare In *Proceedings of the 2009 Winter Simulation Conference*, eds. M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin and R. G. Ingalls, 2844-2851. 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., S.Y. Diallo, and C.D. Turnitsa. 2008. Mathematical Models towards Self-Organizing Formal Federation Languages based on Conceptual Models of Information Exchange Capabilities. In *Proceedings of the 2008 Winter Simulation Conference*, eds. S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler, 966-974. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

- Tolk, A., S.Y. Diallo, R.D. King, C.D. Turnitsa, and J. Padilla. 2010. Conceptual Modeling for Composition of Model-based Complex Systems. In S. Robinson, R. Brooks, K. Kotiadis, and D.-J. van der Zee (Eds.) *Conceptual Modelling for Discrete-Event Simulation*, CRC Press, pp. 355-381
- Tolk, A. and J. A. Muguira. 2003. The Levels of Conceptual Interoperability Model. *Proceedings of the Fall Simulation Interoperability Workshop*, September 14-19, Orlando, FL, IEEE CS Press
- Weisel, E.Y. 2005. *Models, Composability, and Validity*. Doctoral thesis at Old Dominion University, Frank Batten College of Engineering and Technology, Norfolk, VA
- Yilmaz, L. 2004. On the Need for Contextualized Introspective Simulation Models to Improve Reuse and Composability of Defense Simulations. *Journal of Defense Modeling and Simulation* 1(3):135-145
- Yilmaz, L. 2007. A Strategy for Improving Dynamic Composability: Ontology-driven Introspective Agent Architectures. *Journal of Systemics, Cybernetics and Informatics* 5(5):1-9, IIS
- Yilmaz, L. 2007a. Using Meta-Level Ontology Relations to measure Conceptual Alignment and Interoperability in Simulation Models. In *Proceedings of the 2007 Winter Simulation Conference*, eds. S. G. Henderson, B. Biller, M.-H Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 1090-1099. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Yilmaz, L. and B. Mitchell. 2008. Autonomic Introspective Simulation Systems. In L. Yilmaz and T. Oren (eds.) *Agent-directed Simulation and Systems Engineering*, Wiley-Berlin, pp. 37-72
- Yilmaz, L. and T. Oren. 2004. Exploring Agent-Supported Simulation Brokering on the Semantic Web: Foundations for a Dynamic Composability Approach. In *Proceedings of the 2004 Winter Simulation Conference*, eds. R. G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, 766-773. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- 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 A. Tolk. 2006. Engineering ab initio dynamic interoperability and composability via agent-mediated introspective simulation. In *Proceedings of the 2006 Winter Simulation Conference*, eds. L. R. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, 1075-1182. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.
- Yilmaz, L. and A. Tolk. 2008. A Unifying Multimodel Taxonomy and agent-supported Multisimulation Strategy for Decision Support. In G. Phillips-Wren, N. Ichalkaranje, and L. Jain (Eds.) *Intelligent Decision Making: An AI-Based Approach*, Studies in Computational Intelligence 97:193-226, Springer-Verlag Berlin Heidelberg

AUTHORS BIOGRAPHIES

ANDREAS TOLK is an Associate 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 a Research Scientist with the Virginia Modeling, Analysis and Simulation Center – VMASC – 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.