Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

KNOWLEDGE REPRESENTATION AND THE DIMENSIONS OF A MULTI-MODEL RELATIONSHIP

Charles Turnitsa

Virginia Modeling, Analysis and Simulation Center
1000 University Blvd
Suffolk, VA 23435

Andreas Tolk

Old Dominion University
College of Engineering
Norfolk, VA 23529

ABSTRACT

The exchange of data between different models in a multi-simulation environment is about the exchange of information from within the context of two separate worldviews. This amounts to knowledge (information in context). To represent this knowledge, in any of a number of different ways (meta-data, ontological reference, frames, etc.) there must be a method to bridge the gap between what the worldview of origination can represent and what the worldview of destination can receive. This requires some understanding of the dimensions of difference between the two worldviews. As each separate model is the product of a different perspective by the model's developer, each individual model will have a different worldview. Understanding what these differences are, and viewing them in light of the requirements for conceptual interoperability will contribute a great teal to the knowledge that must be captured for meaningful exchange.

1 INTRODUCTION

The field of distributed simulation is based on the notion of combining a number of different simulation systems (each with their own models) together, to form a larger or more complex synthesized simulation space than any one of the systems could provide on their own. Because this space is synthesized, each simulation system may need to have access to data about the overall space, that it is not itself the author of. In order to get access to that data, it may have to rely on retrieval of information from either other individual simulation systems, or from some distributed framework that represents the entire synthesized space. An example of the former is any situation where two (or more) simulation systems have a data exchange agreement with each other. Examples of the latter are DIS and HLA federations.

Each simulation system deals with a different portion of the synthesized space, which is of course what makes it valuable to include in a multi-simulation combination.

However, the fact that each simulation system has a different focus also means that varying data and processes within that system will differ from data and processes in all of the other systems. This is true even when the data or processes within the different simulation systems are modeled after the same real world referent.

The difference in simulation approach (examples might include a system dynamics model, or a discrete event simulation) are immediately identified as the culprit for these differences, as in the case of a hybridized model. There is a necessary reason why some differences would exist, however, between any two distinct simulation systems, regardless of the approach being heterogeneous or homogenous. A simulation system, at its heart, is a software package that can execute one or more models over synthetic time or space. These models consist of a number of objects (that they represent), each a referent of the idealized world that the model is seeking to mimic; and a number of processes (that provide transformation of and influence on the objects). Together these objects and processes represent the artificial world that the model describes. Given some start state for that worldview, and given the means to order the processes based on one or more identified goals, the objects will have their data attributes change during the execution of the simulation.

Interoperability between simulation systems can be described using the levels of conceptual interoperability model - a model that shows the different levels of interoperability that may exist between systems, from technical interoperability through conceptual interoperability. The differences between the two simulation systems exist between systems interoperating at level 1 in the same strength as they exist between systems that are capable of interoperating at level 6. The increased value of each successive level in the LCIM over level one, however, is that accommodations for the differences between world-identity tuples are of increasing value. In plainer terms, the differences between two systems that are interoperating at level one would be the same as if those systems were interoperating at level two - however, if they were to interoperate at level two, either more of the differences, or an increasing depth of the same number of differences, would have to be dealt with in comparison to level one. And so on until level six is reached.

The problem of interoperability has been dealt with by several research groups. The idea of combining interoperability with composability was given an analytical treatment by Petty and Weisel in (Petty and Weisel 2003). The idea of, and requirements for composability to assist with models (especially military models) was presented by Davis and Anderson in (Davis and Anderson 2003). The possible dimensions of the differences between the involved simulation systems (and their world-identities) have not previously been explored, and are the focus of this paper. By exploring the various implications of the world-identity tuple, the various static and dynamic dimensions of difference between simulation systems are highlighted and defined below.

2 KNOWLEDGE REPRESENTATION, ONTOLOGY AND INTEROPERABILITY

Knowledge representation is the presentation of information, in the context of a world view. The information is data, identified to have some semantic meaning. The context is a definition of what that semantic meaning is valued at, within the framework of its world view. This increased layering of requirement, from data through information, knowledge, and beyond, is captured in the Knowledge Representation Requirements Model (KRRM). This is a model showing the increasing needs for greater ability to represent knowledge, and also gives the levels of conceptual interoperability that may be reached if the requirements are met.

The KRRM has five levels of requirement, showing describing what is needed for a system to either represent or accommodate knowledge. The depiction in (**Error! Reference source not found.** – Knowledge Representation Requirements Model) shows the KRRM levels on the left, along with the Levels of Conceptual Interoperability Model (described below) levels on the right that the KRRM levels correspond to. A brief description of the KRRM requirements follows.

- **Data** – the only requirement for the depiction of knowledge at the data level is just that – Data. This corresponds closely to level 2 of the LCIM, which is the Syntactic Level.
- **Information** – this requirement becomes more useful, as now the data is tagged with some semantic label, depicting what the data means to the system of origination. This corresponds to level 3 of the LCIM, which is the Semantic Level. At this level of requirement, some sort of reference model is referred to for the semantic tagging – which could be an agreed to meta-data schema, or a reference data model, to name a few examples.

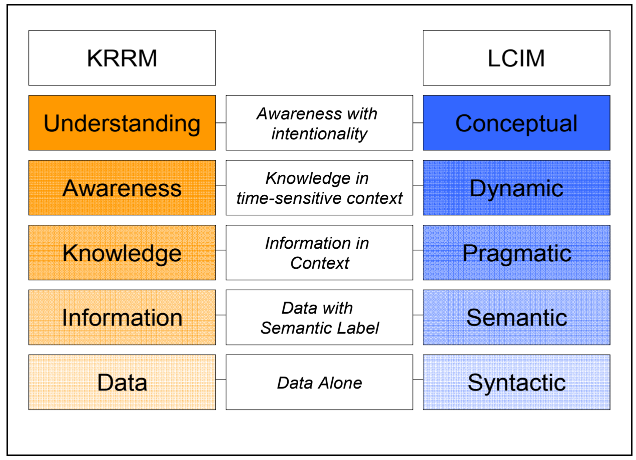

Figure 1:Knowledge Representation Requirements Model

- **Knowledge** – this requirement requires that the semantic labels that transform data into information are now described, meaningfully, in the context that they originated in the system of origination. This corresponds to the Pragmatic Level, or level 4 of the LCIM. At this level of requirement, typically some sort of ontological representation is relied upon in order to capture the context. For context, here, we refer to the ontological entities (described below), which includes all of the objects and processes that make up the system. The context, at any one time, is an enumeration of all of the objects and processes making up the system's world, and any attribution they might have. This follows after the work in (Devlin 2006).
- **Awareness** – this requirement leaves the realm of static knowledge and enters the levels of dynamic knowledge. The information-in-context of the Knowledge requirement level now must be able to be represented in a time-sensitive way, as the context may change dynamically with the life of the system. This corresponds to the Dynamic Level, or level 5 of the LCIM, and not surprisingly this level of requirement requires some method of ontologically representing the changing worldview of the system, as it develops over time.
- **Understanding** – Once the awareness level of knowledge requirement is reached, the next level of requirement is Understanding, which relies on not only having a method for representing knowledge in a time-sensitive context, but also anticipating the intentionality of the originating system, such that the dynamic worldview can be anticipated. This requirement matches up with level 6 of the LCIM, Conceptual Interoperability.

Each simulation system has its own worldview, attributable to its varied parentage (developed by different sources, for different purposes). In order to represent and exchange knowledge, therefore, a way for expressing the semantics of the information making up that knowledge –

from the framework of its origination worldview – must be possible. This is accomplished by using an ontological representation to define the semantic meaning of the information. A commonly used reference for defining what an ontological representation is for information system knowledge exchange is (Gruber 1993). This states that an ontological representation is a "formal specification of a conceptualization". The formal specification is to define the information being exchanged, and the conceptualization is the understanding of the worldview of origination for that information.

In order to access the conceptualization that an ontological representation is a formal specification of, it is necessary to break that specification up into accessible components. The first three types of components that are discussed are entities, relations and rules. Entities and relations are quite familiar to the data modeling community, and also appear within most modern ontological engineering theories. Rules, however, are an additional component that assists with the ontology model being useful to systems, and will be described here in more detail. A fourth component, primitives of meaning are essential to the other component types and will be addressed below.

In light of the fact that this paper is discussing knowledge representation and the dimensions of difference between simulation systems, and to define these it relies on the ontological representation of information systems, and more specifically, ontology for the purpose of assisting interoperability between information systems, entities become quite easy to define. As entities are revealed in the entity-relational model (Chen 1976), it can be seen that they are easy to recognize within a model. Entities are the exchangeable symbols (words, data elements, etc) that represent the *things* of which our systems can address. Things are further defined as being not only physical things, but also everything, which can be addressed by systems (things, both physical and otherwise; phenomena, including both processes and events; modifiers for both of these). It is typical to think of entities as the three dimensional objects of the worldview, but this is not the only practice. There are a number of systems that are described as being four dimensional that is their worldview is one based in time, and the temporal confluence of entities, but not so much that any entity is permanent. A good overview covering ontological representations of these two worldviews, and how the Basic Formal Ontology can describe them is found in (Grenon 2003).

Entities, in order to satisfy the specification presented here, need to be represented as both types and instances. Entity-types may be divided up further into subtypes, but each child of an entity-type (whether a true instance, or a subtype) retains all of the identity of the parent type. This idea of terms of understanding being less generally defined than their parents is known in the knowledge representation and artificial intelligence communities as subsumption and a treatment of the topic can be found in (Brachman and Schmolze 1985). The organization of all of an ontology model's entities into an interconnected graph is referred to as a taxonomical model.

Different entities, originating from different systems, may have the same "name", or symbol, representing them and have different characteristics. This leads to a situation making the enablement of interoperability very difficult. Additionally, difficulties in enablement would arise when differently named entities are meant to represent the same thing from our limited universe of discourse. In both situations, and as hinted at above, it can be seen that entities differ from each other based on their characteristics. These characteristics are defined by the primitives of meaning that the entities can exhibit. This is discussed further, below.

The type-subtype-instance relationship (of the taxonomical model) is not the only class of relations between entities that can exist. Relations can provide a semantic link between entities in any number of different ways. The enumeration of particular relation types is potentially unique for each universe of discourse (Smith, et. al. 2005).

System-to-system interoperability requires exchange of data, and that data must have a syntactic form. Further, to proceed to even higher levels of conceptual interoperability, semantics are required of the data interchange. In both cases, and for further extension, a rule set, or grammar, is required to control the syntax and semantics of the data exchanged. But the data within a system undergoes certain operations defined by that system. A set of rules defining the syntax and semantics of those operations is also required.

The existence of a taxonomical model that systems can reference allows for the specific identification of entities referred to during system-to-system communications (Zhang and Ziegler 1989). A set of rules can provide for a semantically meaningful method for combining those entities into communications that satisfy the system-to-system communications supporting interoperability up to the semantic level. Internal relations identified among the entities of a system's data model even allow, in effect, inference to be made within the interoperability supporting data exchanges between systems. What is still missing from our ontology, although it was mentioned several times above, is the specific characterization of our entities. This characterization provides for definition of our entities, and also allows for the application of the relations and rules defined above. Primitives of meaning, which are exhibited by entities, provide this characterization.

Primitives of meaning, or just "primitives", are the basis for giving entities definition and characterization. They are the most difficult component of the ontology to define. They are also often difficult to see within the entities that exhibit them. It is helpful to have a good definition of what is meant by concept in order to see how the ontology model requires them. One aspect of primitives to consider during the definition of the term is that primitives are the only component of our ontology that exists within actual items. They are the link between a data representation of an item, and the actual item itself. The concepts behind, for instance, a truck, and the data representation (within an information system) of a truck are the same (Sowa 2000). These concepts are what we are calling primitives.

Each ontological entity has a unique collection of primitives of meaning. Within the domain that the systems in question come from, the primitives of meaning must be universally recognized and accepted. However, each system's ontological representation may have a different collection of primitives that make up the various entities it entails. This gives the different morphology of similarly named entities, and is where the defined difference may be found between the different system's worldviews. As each system is a different abstraction of potentially the same reality, the difference is in which primitives of meaning each system assumes are involved in the make up of their ontological entities.

The terms to be exchanged between simulation systems – which are to capture the knowledge that the originating system wishes to convey to the receiving system – are representative of some collection of these primitives of meaning. In order to capture the meaning, as conceived of by the system developer via the semiotic triangle which connects referent to term (through conceptualization), an enumeration of the primitives of meaning will be relied upon.

If the primitives, which give identity to an entity, are known, and captured within the ontology, then regardless of any ambiguities with the entity's name (or symbol), it can still be clearly identified by using exactly these concepts (Sowa 2000). Similarly, proper definition of the primitives that give definition to the entities of two different systems interoperating with each other can show where there may be conceptual gaps or misalignment between those entities.

The primitives of meaning can ontologically show the collection of all of the aspects that collectively make up what a term is, but how did that term become understood in the first place is described by the semiotic triangle (described below). Although we are talking about symbols and terms as the knowledge that is being exchanged between systems, it also includes all of the processes that affect the entities making up that knowledge.

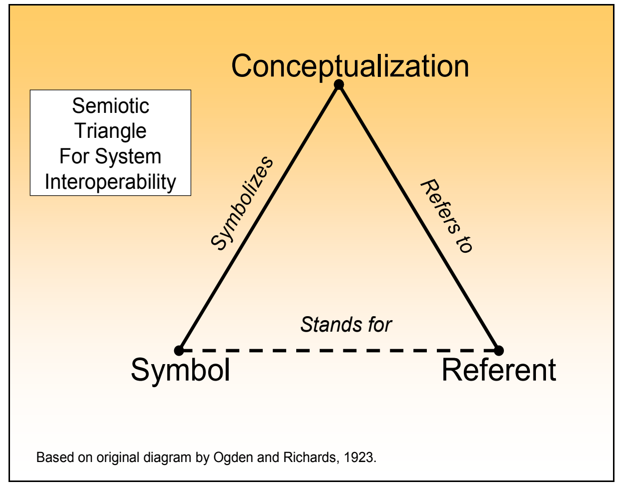

Figure 2: Semiotic Triangle for System Interoperability

Some definition of the difference between terms and meaning is appropriate. A useful source for this definition is the theory of semiotics. Semiotics is the theory of meaning of symbols. Some of the main contributors of ideas in this field were Charles S. Peirce and Ferdinand De Saussure. As a concise introduction, and to serve our purposes, we will rely on the semiotic triangle (Ogden 1923). As can be seen from (Figure 2: Semiotic Triangle for System Interoperability), a system will have some basis for referring to real world (or imagined world) referents, that the developers conceived of. The stronger the bond (meaning it is harmonized) between symbol and conceptualization, then the more "correct", or semantically accurate, the symbol is. Finally, if there is an adequate reference (relationship between referent and conceptualization), and if there is a correct symbolization (between conceptualization and symbol), then there is a relationship whereby the symbol actually stands for the referent.

For our purposes, the terms of exchange between systems are symbols (representing the semantic information in context, or knowledge, being represented for exchange). The meaning behind those symbols is represented by the relationships between the symbol and the conceptualization, and between the conceptualization and the referent.

3 MODEL DIFFERENCE AND THE LCIM

The research on composability conducted at the Virginia Modeling Analysis & Simulation Center resulted in the LCIM, which underwent several improvements since its first publication (Tolk and Muguira 2003). The current version of LCIM as depicted in (Figure 3: Levels of Conceptual Interoperability Model) is documented in (Turnitsa 2005). The different levels are characterized as follows:

- Level 0: Stand-alone systems have **No Interoperability**.
- 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 it to exchange bits and bytes, the underlying networks and protocols are unambiguously defined.

- 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.
- 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.
- Level 4: **Pragmatic Interoperability** is reached when the interoperating systems are aware of the methods and procedures that each other are 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.
- 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**, then they are able to comprehend the state changes that occur in the assumptions and constraints that each other is making over time, and are able to take advantage of those changes. In particular when interested in the effects of operations, this becomes increasingly important; the effect of the information exchange within the participating systems is unambiguously defined.
- 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 will be documented based on engineering methods enabling their interpretation and evaluation by other engineers. In other words, on this we need a "fully specified but implementation independent model" as requested in (Davis and Anderson 2003) and not just a text describing the conceptual idea.

It should be pointed out that these layers of operations are still driven by implementations of agile systems that should be described in order to enable intelligent software agents to evaluate their applicability to support a decision and their composability with other solutions. As such, it is a typical bottom-up approach To what degree the bottom-up approach can be merged with top-down approaches, such as the coherence/correspondence approach described by (Sousa-Poza 2005) is a topic of ongoing research.

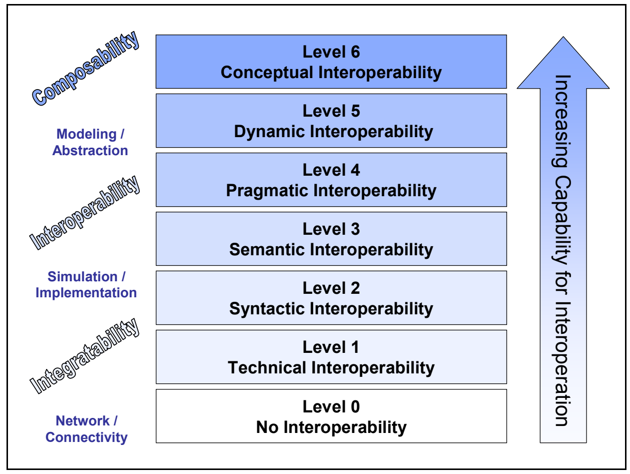

Figure 3: Levels of Conceptual Interoperability Model

The LCIM was applied in various domains successfully and featured as a reference model in various journal contributions and book chapters. The originally intended use is described in (Tolk et al. 2006): applying the ideas to support composable M&S service for net-centric command and control applications. The Interoperability Framework for future U.S. Department of Energy solutions for the Power-Grid described in (Gridwise 2007) uses a derivate of the model. How to apply the LCIM to align smart applications is the topic of (Dobrev et. al. 2007). Finally, the recent book on model and simulation-based data engineering uses the LCIM to show functionality and supported concepts of their solution (Zeigler and Hammonds 2007). The study of Carnegie Mellon University on System of Systems mentions the LCIM as one of the candidates for successful evaluation of approaches (Morris et. al. 2004).

Defining the ontology of the system is important, to derive and expose the primitives of meaning. A further aid in understanding the dimensions of difference between systems, and of breaking down the elements of knowledge into atomic parts, can be accomplished by specifically defining a simulation systems world-view. The proposed method here is to derive a tuple, which consists of a number of defining statements that would describe all of the pertinent components of a system's world-view. This tuple is the World-View Identity Tuple (WVIT).

4 IDENTIFYING A SIMULATION SYSTEM'S WORLD-VIEW

Ontologically speaking, we have defined ontological entities to be anything within the system that can be addressed. This includes not only objects (we will use the term objects, rather than simulation entities, to avoid confusing the term entity), but also processes. The other items to be captured as part of the ontology, according to situation theory (Devlin 2006) would include temporal and spatial references, situations, infons, and parameters. Additionally, following after (Sowa 2000), processes are equally as important as objects in representing the knowledge of a worldview. So to the list from situation theory, we add an equally rich list dealing with processes. As all of the temporal and spatial references, situations, infons, parameters that may apply to either processes or objects are themselves attributes of those two types of ontological entities, we identify the first two elements of the WVIT to be objects and processes. Together we refer to the elements (temporal reference, spatial reference, situation, infon, parameter) suggested by (Devlin 2006) as the attributes of the objects and processes. Whenever we speak of attributes, it is assumed that attributes have values, and the affects on and depictions of attributes often affect the values of that attribute more than the existence of the attribute itself, although this is certainly possible.

One of the things not captured in the ontological description described earlier in this paper concerns the behavior of both the objects within the world, and also the behavior of the world as a whole. We will differentiate these two from each other by referring to the first as behavior, and the second as intention.

Objects are the simulation entities (although we try to avoid that term, so as not to confuse it with ontological entity) that exist within the synthetic environment of the simulation system. They also include all of the other features of the synthetic environment that the simulation entities can interact with. Objects have (from their ontological depiction, described earlier) as attributes a series of primitives of meaning, each of which is responsible for defining one aspect of the object. Each primitive of meaning may be responsible for one or more attributes that the object has. In addition, each object has a number of different ontological relationships that determine its place in the taxonomical ordering of ontological entities, as well as other world-view appropriate relationships that define its capabilities, roles, and subjectivities. Taken together, an object then is itself a tuple of its ontological primitives of meaning, attributes that derive from the primitives of meaning, and all of the ontological relationships that it may be subjected to.

Processes are the functions that exist within the simulation system world-view. They are responsible for bringing about change to the attributes that the objects of the simulation system exhibit. Processes, therefore, resemble a function, and are defined by which primitive of meaning/attribute pairings they affect, and the definition of how they can affect each of those pairings. The process, which is itself an ontological entity, is described by its primitives of meaning, the resulting paired attributes (defining such things as duration, reliability, predictability, etc.), and also its ontological relations (defining which objects it may affect). Just as the objects, then, the process description is itself a tuple, consisting of its ontological primitives of meaning, attributes that derive from the primitives of meaning, and all of the ontological relationships that it can be part of.

Behaviors are the processes that an object can undergo, due to its own defined nature. As with everything else in the WVIT, this is subject to the conceptualization of the referent that the simulation system developers chose to observe, so that while the same object that may be represented in several different simulation systems, will always exhibit the same behavior. The ontological element described earlier known as rules are what are of interest here. As such, a behavior would include a pairing of an object, a process (both connected via a common ontological relationship). Note that the primitives of meaning and resulting attributes that are affected by the particular relationship are already described in the tuple defining the process. As with the earlier world-view identity items (the objects and processes), this leads behaviors to also be a tuple, consisting of an object/process pairing.

The final world-view identity item, intentions, is responsible for the overall behavior description of the simulation systems world view. This is where the goals of the individual objects, and groups of objects, can be defined. For the WVIT, this means that the intention is (not surprisingly) a tuple. It defines the desired condition, in terms of an object and its attributes, as well as (optionally) a series of (one or more) time depictions. In this way an intention could state, for example, that a desired goal (outcome) of the simulation-system would be that all Person objects be alive (attribute of Health be greater than 0) at the end of the simulation execution.

The WVIT is then a tuple, consisting of four items. Each of those items is itself a set of all possible component tuples of a particular type.

4.1 Static Dimensions of Difference

These first three dimensions of difference are static. That is, they apply to the structure of the simulation, or rather the models that the simulation system executes over time, but the nature of the difference is not based on the simulation system actually executing. The differences, here, can be seen just by examining the WVIT of the various systems statically.

4.1.1 Multi-Resolution

Different world-views concentrate some part of their identity on amalgams or aggregates of items that could exist at high levels of detail. For instance, an object could actually represent a grouping together of other objects that exist at a higher level of detail, but can all be combined into something addressable at a lower level of detail. The most common example would be a military simulation that concentrates on the actions and effects of combat at the level of a battalion, rather than of the companies, platoons, squads or individuals that make up that battalion. Another example would be a rail-line engineering simulation that showed the effects of a steep grade on the overall train, rather than showing the affects of the grade on the individual cargo carriages that make up the train. The same dimension of difference could apply to the other elements in the world-view identity as much as they apply to objects. They could apply to processes, goals and behaviors.

Usually, within a simulation system the resolutions of all the world-identity items are required to be in alignment with each other, but not always. Consider a military simulation system that handles units of battalion level strength and identity, but in its combat adjudication processes, temporarily converts (internal to the process) the battalion to companies for purpose of determining effects, but then converts back to battalion before applying the effects to the object.

4.1.2 Multi-Scope

In some ways, this difference is the easiest to spot, but may have the gravest consequences to simulation interoperability. In the case of a Multi-Scope difference between systems, it is quite likely that the traversal of the semiotic triangle by the various system designers resulted in systems that concern completely different objects, processes, behaviors, and/or intentions from each other. The idea of a developer conceptualizing a referent, and then developing the symbols (ontological entities, rules, relations, and their implementation in a system) for a particular purpose, and a separate developer developing a separate system (although based on the same referent) could quite easily lead to this difference.

Accommodating interoperability, and knowledge exchange, between systems with this level of difference is accomplished by some sort of translation method, where the entities that system A requires from system B are introduced in the interoperability connectivity between the two. The limitation here is, at most, level three in the LCIM, or the Semantic Level, and that is only if certain data for the interoperability can be derived from data in the system or origination, and the translating method introduces the semantic markup needed to increase the data to information (following the KRRM). If this is not possible, then the greatest level of interoperability, given this dimension of difference, is Syntactic, or LCIM level two.

4.1.3 Multi-Structure

As each of the ontological entities, specifically objects and processes, are defined ontologically (and later, in the WVIT) as containing primitives of meaning, it is possible that the same object or process, representing the same referent, from different simulation systems, might have different internal structure of primitives of meaning. Additionally, other defining elements of the object or process, such as the relationships or rules that they are subject to ontologically, might also be different between different simulation systems. This is why the WVIT is so important to be aware of when comparing simulation systems, and the knowledge that each espouses and is capable of representing.

In terms of the LCIM, systems may be up to level three, Semantic Interoperability, compatible and still have Multi-Structure variances. At level three, for instance, there is a semantic value to the data being produced, marked up to information (as per the KRRM), and exchanged, but the ontological definition of the primitives of meaning are not part of the system definition, nor the needs of the interoperability, yet. Once level four, Pragmatic Interoperability, is reached, the need for an ontological representation of the system requires that such primitives of meaning be exposed, and here Multi-Structure difference will prohibit interoperability.

Some accommodation for Multi-Structure difference could be made with a purpose-built translator from the world-view of one simulation system to another; however this could be potentially encompassing as the system it is seeking to accommodate. For instance, if a translator accommodates the passing of objects by making up for deficient primitives of meaning, lacking in their definition in a system they are being exchanged to, then the processes of the original system that may rely on the presence of those primitives of meaning (and their attendant attributes) to operate on the object once it is returned, and it may have expected that processes it underwent in the other simulation system should have made changes to those attributes. To build these processes into the translator system, is to mimic the functionality of the other simulation system, redefined with the world-view of the originating simulation system.

4.2 Dynamic Dimensions of Difference

The dynamic execution of the simulation-system over time can introduce new changes to the structure of the world-view of that simulation-system. This, of course, can introduce differences between it and another simulation-system that wasn't apparent upon static investigation of the world-view. This is the domain of the dynamic dimensions of difference. A note should be inserted here that two of these three are actually meta-dimensions. The first meta-dimension is in the area of Multi-Stage differences, where the differences are introduced by having a repeated iteration of a multi-simulation combination, and each new execution brings in some changes to the WVIT of one of the simulation systems (sometimes introducing new differences). The second meta-dimension is the dimension of multi-perspective, which is actually a catch-phrase to include one or more of the other dimensions. It, in itself, only refers to the differences that the initial designers of the various simulation systems might have had as they traversed the semiotic triangle, leading from referent to conceptualization to symbol. But the resulting differences from this disjoint perspective results in one or more of the other dimensions.

4.2.1 Multi-Phase

Given that a simulation system encapsulates a defined world (leading to the WVIT), it is possible that with permutations within the simulation-system that the world-view itself will change (either intentionally, or otherwise). The possibility of this being unintentional would be a rare occurrence in a software based automata system, as all of the definitions for the objects, processes, behaviors and intentions must be developed to change dynamically, and to adjust their parameters and operations with that dynamic change. It is, however, theoretically possible. In any event, whether intentional or not, it is possible that the overall world view of the simulation system would change. This leads to a dynamic WVIT.

If a separate simulation system that the original system is interoperating with is not aware of the dynamic nature of the WVIT, then it is likely that it will lose whatever level of interoperability it was capable of, once elements of the originating systems WVIT have modified. In order to accommodate such changes, there must be some sort of knowledge translator that is dynamically aware of the changes in WVIT, and must accommodate them using whatever methods described under Multi-Resolution, Multi-Scope, or Multi-Structural differences are then introduced. If this can be accomplished, then the level of Knowledge Understanding, or Conceptual Interoperability, exists between the simulation systems.

4.2.2 Multi-Stage

This situation comes from the introduction of the concept of multiple iterations of the execution of the simulation systems in question. Each individual paired iteration of execution is potentially subject to any of the other dimensions of difference, but with multiple iterations (with the purpose of altering the base conditions of attribute values, or even object or process presence, for the purposes of a simulation study or experiment) then the concept of ranges of acceptable WVIT values come into consideration.

The ideas of Multi-Stage Multi-Simulation comes from (Yilmaz et. al. 2007), and is actually a common occurrence, especially with simulation techniques that rely on multiple “runs”, or executions of the simulation system in order to produce a range of results.

With each exception that causes this difference to be present, one or more of the other dimensions could be present, depending on how much the WVIT has changed in the case where difference is introduced. Depending on which of the other dimensions are affected would determine the extent to which the LCIM or KRRM could be accommodated.

As this difference is really a meta-difference – after all, some of the individual iterations of the multi-simulation existed with a reduced level of difference, it is only after the concept of multiple iterations with different WVIT values that this dimension becomes apparent – it is only able to be accommodated with a situational fix, again depending on the nature of dimensions of difference that each new iteration brings with it.

4.2.3 Multi-Perspective

As the system world-view identity tuple can show, the worldview of each system is different. This difference stems from the perspective that the system designer had when developing the system. The objects, processes, goals, and behaviors that make up the worldview are based on that perspective, and all of them are based on some referent that the simulation system designer was seeking to emulate. That referent (or set of referents) might have been considered by another simulation system designer, but from a different perspective. For instance, if one simulation system designer is attempting to model a combat situation for infantrymen operating in an urban environment, and another simulation system designer is seeking to do the same thing, there might be some expectation of overlap between the two resulting systems. However, if the perspective of the first designer is to show the results of different courses of action by the infantry objects in the simulation, and the second designer's perspective is to show the effect that different types of urban buildings have on small unit infantry tactics, then there will be some identifiable difference between the systems. Perhaps the world-view entity tuples of each would show identical objects, maybe even identical behaviors and identical processes - but there would be (in this example) different goals. These differences derive from having different perspectives.

This dimension of difference is included largely based on the fact that it was identified early on in the works of (Davis and Anderson 2003). In light of the other dimensions it appears unlikely that it would appear as a sole difference. More likely it appears as if a difference in perspective would lead to differences in resolution, structure, phasing, staging, or scope.

REFERENCES

Brachman, R., and J. Schmolze. 1985. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science*, 9(2): 171–216.

Chen, P. P. 1976. The Entity-Relational Model – Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1(1): 9--36, March, 1976.

Davis, P. K. and R. H. Anderson. 2003. *Improving the Composability of Department of Defense Models and Simulations*. RAND, National Defense Research Institute Report.

Devlin, K. 2006. Situation Theory and Situation Sementics. *Logic and the Modalities in the Twentieth Century*, Elsevier Handbook of the History of Logic, Vol. 7, ed. by D. Gabbay and J. Woods, Elsevier.

Dobrev, P., O. Kalaydjiev, G. Angelova. 2007. From Conceptual Structures to Semantic Interoperability of Content. Bookchapter in `Conceptual Structures: Knowledge Architectures for Smart Applications`, pp. 192-205, Springer Verlag.

Grenon, P. 2003. Spatio-Temporality in Basic Formal Ontology, Institute for Formal Ontology and Medical Information Science, University of Leipzig, 2003. Available online at: `http://www.ifomis.org/bfo` (last visited March, 2008)

GridWise Architecture Council Interoperability Framework Team. 2007. Interoperability Context-Setting Framework. V1.0, July 2007.

Gruber, T. 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition*, 5(2), 199-220.

Morris, E., L. Levine, C. Meyers, P. Place, D. Plakosh. 2004. System of Systems Interoperability (SOSI), Final Report. Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 2004

Petty, M. D., E. W. Weisel. 2003. A Composability Lexicon, Proceedings IEEE Spring Simulation Interoperability Workshop, IEEE CS Press.

Smith, B., W. Ceusters, B. Klagges, J. Kohler, A. Kumar, J. Lomax, C. Mungall, F. Neuhaus, A. Rector, C. Rosse. 2005. Relations in biomedical ontologies. *Genome Biology*, April 2005

Sousa-Poza, A. A. 2005. Pragmatic Idealism as the Basis for Understanding, *Proceedings 2005 International Conference on Systems, Man and Cybernetics*, IEEE Press.

Sowa, J. 2000. Ontology, Metadata, and Semiotics, Published in B. Ganter & G. W. Mineau, eds., *Conceptual Structures: Logical, Linguistic, and Computational Issues, Lecture Notes in AI #1867*, Springer-Verlag, Berlin, pp. 55-81.

Tolk, A., J. A. Muguira. 2003. The Levels of Conceptual Interoperability Model (LCIM). *Proceedings IEEE 2003 Fall Simulation Interoperability Workshop*, IEEE CS Press.

Tolk, A., S. Y. Diallo, C. Turnitsa, L. S. Winters. 2006. Composable M&S Web Services for Net-centric Applications. *Journal Defense Modeling and Simulation* 3 (1): 27-44.

Turnitsa, C. 2005. Extending the Levels of Conceptual Interoperability Model, *Proceedings IEEE 2005 Summer Computer Simulation Conference*, IEEE CS Press.

Yilmaz, L., T. Ören, A. Lim, and S. Bowen. 2007. Requirements and Design Principles for Multisimulation with Multiresolution, Multistage Multimodels, 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. 823 – 832. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zeigler, B. P., H. Praehofer, and T. G. Kim. 2000. *Theory of Modeling and Simulation*. 2nd Edition, Academic Press.

Zeigler, B. P., and P. E. Hammonds. 2007. *Modeling & Simulation-Based Data Engineering: Introducing Pragmatics into Ontologies for Net-Centric Information Exchange*. Academic Press.

Zhang, G. and B. P. Zeigler. 1989. The system entity structure: Knowledge representation for simulation modeling and design. In *Artificial Intelligence, Simulation and Modeling*, L. E. Widman, K. A. Loparo, and N. R. Nielsen, Eds. New York: Wiley, pp. 47-73.

AUTHOR BIOGRAPHIES

ANDREAS TOLK is Associate Professor for Engineering Management and Systems Engineering of Old Dominion Universities Modeling, Simulation, and Visualization Faculty. He is also a Senior Research Scientist at the Virginia Modeling Analysis and Simulation Center (VMASC). He holds a M.S. in Computer Science (1988) and a Ph.D. in Computer Science and Applied Operations Research (1995), both from the University of the Federal Armed Forces of Germany in Munich. His e-mail address is <atolk@odu.edu>.

CHARLES D. TURNITSA is a Ph.D. candidate at the Virginia Modeling Analysis and Simulation Center (VMASC) of the Old Dominion University (ODU). He received his B.S. in Computer Science (1991) from Christopher Newport University (Newport News, Virginia), and his M.S. in Modeling & Simulation (2006) from ODU. His Ph.D. research under Tolk focuses on the domain of dynamic and fractal ontology models for M&S interoperability. His e-mail address is <cturnits@odu.edu>.