Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

THE ELUSIVENESS OF SIMULATION INTEROPERABILITY – WHAT IS DIFFERENT FROM OTHER INTEROPERABILITY DOMAINS?

Andreas Tolk

The MITRE Corporation
903 Enterprise Parkway, Ste200
Hampton, VA 23666, USA

ABSTRACT

Simulation interoperability is a recurring theme in simulation conferences and workshops for more than 20 years. With the IEEE Standards 1278 and 1516, two simulation interoperability standards were introduced, and both were adapted and implemented by the community. Nonetheless, the simulation community is still struggling with interoperability challenges that are not solved. Why is this the case? This paper gives an overview of the current approaches to simulation interoperability, including the standardized approaches as well as contributions of simulation formalism. It then addresses the mathematical foundations of simulation interoperability, including model theory. As a result, the need for the consistency in the representation of truth in all participating simulation systems emerges as the concept that needs to be addressed by interoperability solutions.

1 INTRODUCTION

Over the last decade, we witnessed a significant increase in interoperability in the computational world, in particular in domains of cyber-physical systems (CPS), the Internet-of-things (IoT), and initiatives like ”Industry 4.0”. We expect our new smart TV to inter-operate with our home wireless network. When getting a rental car at the airport to drive to the conference, we expect to be able to pair our cellphone with the car’s radio systems for phone calls and listening to our own music mixes. We simply got used to things that play well with each other without requiring integration efforts. But how come that this is not the case for computer simulation interoperability? Why do we still struggle to bring independently developed simulation systems together and create a useful composition that provides more and better functionality to the user than the individual simulation systems can do?

For computer engineers, the task of creating a simulation composition is relatively simple, as the task is mainly to bring objects and processes together that are implemented in multiple systems. Objects have defining attributes and relations. Processes are used for creating, updating, changing, or deleting objects. If objects exist in more than one system, they have to be aligned. If processes exist in more than one system, they have to be synchronized. The whole composition may need to be orchestrated, but all of these are relative straightforward tasks: aligning, synchronizing, and orchestrating are conducted in many other domains successfully, so it should be easily doable for simulation systems as well.

Within this paper, I will describe and discuss the current simulation interoperability approaches with focus on the two standards IEEE 1278 Distributed Interactive Simulation Protocol (DIS) and IEEE 1516 High Level Architecture (HLA) and demonstrate how they address the challenges of aligning, synchronizing, and orchestrating. I will then describe the mathematics behind interoperability challenges, starting with revisiting the Levels of Conceptual Interoperability Model (LCIM), followed by a section on model theory.

This description of the mathematical foundation will show that the main challenge of simulation interoperability is not a software or computer engineering problem, but the models implemented by the simulation systems must be composable, which motivates the distinction between integratability of infrastructures, interoperability of simulation systems, and composability of models. It is the last point of this enumeration, the composability of models, that makes simulation interoperability different from other domains. This is a conceptual domain that cannot be solved by technical solutions. Nonetheless, the task can be significantly facilitated by several methods from the systems, software and computer engineering domains, as will be discussed in the last section.

2 SIMULATION INTEROPERABILITY STANDARD APPROACHES

As discussed in (Hill and Tolk 2017), the requirement to inter-operate independently developed simulation systems was mainly driven by the Armed Forces in the last two decades of the 20^{th} century. Within the NATO alliance, the Armed Forces had made significant investments into simulation technology to train their soldiers, support the acquisition process for new systems, and conduct analysis on tactical, operational, and strategic levels. The simulators and simulation systems developed by different nations, services, and branches represented the special tasks they were designed for well, but did not provide the same fidelity for other branches, services, or nations with which they had to collaborate. Instead of developing a new simulation system that was comprised of components with the same high fidelity, the idea was to compose the existing simulation systems to provide a new set of functionality.

At the end of the 20^{th} century, the medical and health care simulation industry gained comparable status in the search for simulation interoperability to support composable and reusable solutions. Comparable to the examples given for the defense domain, many high-fidelity models have been developed that look at the details of healthcare. These include computer models of injuries of particular organs and other body parts, as well as dynamic models from soft tissue body part models to mannequin based patient simulators. Dynamic models are replacing cadavers and animal tissue as the first choice for medical education. This increase in simulation lead to the wish of combining such simulators and simulation systems into a more realistic digital patient, in which high-fidelity organ simulators fulfill their special role in an orchestrated set of simulators and simulation systems. Together this would provide better education, analysis, and medical insight than could be created by the stand-alone systems alone (Dunn et al. 2013).

Other application domains, such as transportation, architecture, and energy, are starting to get interested in simulation interoperability solutions as well, and new application fields, like smart cities and houses, are using model-based solutions that benefit from these standards. In the following sections, we will look at the current standards in the context of their approach to simulation interoperability.

2.1 SIMNET and the IEEE1278 Distributed Interactive Simulation Protocol

The year 1983 is often seen as the start of the modern story of simulation interoperability within distributed simulation, as the Simulator Networking (SIMNET) program was initiated by the Defense Advanced Research Projects Agency (DARPA) (Cosby 1995). SIMNET objectives were to bring armor, mechanized infantry, helicopters, artillery, communications, and logistics components together into a common, situated, virtual battlefield. Simulator crews were supposed to observe each other, communicate via radio channels, and observe each others effects. The simulators representing the various systems remained autonomous and were responsible for local perception and modeling the effects of events on its objects. They sent events to other simulators to communicate events, such as changing positions, fire fights, damage reports, etc. The receiving simulations determined if that information was relevant and what the effects would be.

This activity was successful and caught the interest of the Armed Forces as well as Industry and lead to the development of the Distributed Interactive Simulation (DIS) protocol (Office of Technology Assessment (OTA) 1995). The idea was to keep the standard easy to understand, easy to implement, and open for future developments. In a series of workshops, the focus of DIS was the development of the Protocol Data Unit(s) (PDU) that capture the information to be shared between the autonomous simulators. These PDUs had clear interpretations in the real referenced world: the firing at a target, hitting the target, moving through the terrain, colliding of two objects, and so on. The ideas were captured in the standard family IEEE 1278.

As simulators continue to operate on their own, using their visual representation of their perception and compute their own effects, several additional activities were needed to ensure fair fight conditions between them. It is assumed that the DIS world shares a common understanding of terrain and other cultural features, but it is not enforced. As a result, the representation of the common environment and its objects may differ in the various simulators, leading to systemic advances of one simulator over another. Imagine two tank simulators fighting against each other, one of them representing vegetation, the other one is only representing the profile of the terrain. While the crew of the first second tank can already see the opposing tank and open fire, the first tank cannot see his opponent due to the vegetation. He is fired at, gets killed, and had no fair chance due to the setup of the experiment. *Consistent representation of the common simulation sphere could not be enforced.* Within DIS, this problem was addressed by Best Practice guides making developers aware of the problem and recommending means outside of the core standard to support solutions. The following figure demonstrates the challenge.

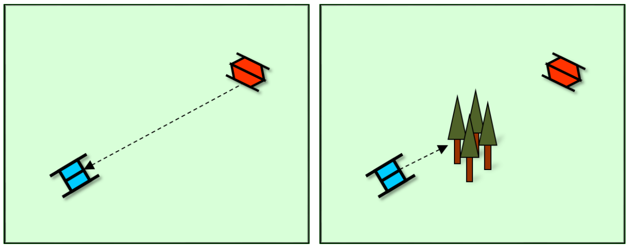

Figure 1: Fair Fight Challenge.

DIS also ensured unambiguous situations by clearly distributing responsibilities for the distributed computation of effects. If an effect happens between two simulators, the initiating simulator is responsible to calculate the location of the effect, and the receiving simulator computes the effect. Again, using a duel is the best example: the shooting tank computes if he hits or misses the target and, in case of a hit, sends a PDU with details of the hit to the target. The target receives these data and computes the damage based on its own data on armor, terrain fortifications, etc. While their is no ambiguity related to each individual events, there can still be inconsistencies based on which simulator is the target: as the simulators are autonomous, the way they compute damage is simulator-specific and can result in counterintuitive results, as the exact same duel with the exact same system can lead to different results depending on which simulator represents the target systems. *Consistent representation of common effects could not be enforced.* This challenge is much harder to address, as these effects are often deeply embedded in the code of the simulator software.

Despite these shortcomings, DIS is still very successful and still a dominant standard. The idea is an easy to communicate, to understand, and to implement standard that allows independent representations of systems in a common simulation sphere to exchange status updates and represent interactions by data exchange.

2.2 The IEEE1516 High Level Architecture

While the DIS protocol focused on simulators and simulation systems supporting the representation of individual weapon systems, another simulation interoperability effort targeted the aggregated levels of system representation, i.e., higher levels units, such as brigades or divisions, to support command post exercises (CPX) or computer assisted exercises (CAX). In these exercises, several weapon systems were aggregated into a group of weapon systems, homogeneously or heterogeneously, and were used to represent the objects of interest for the training community. The DARPA activity initializing supporting research was the Aggregated Level Simulation Protocol (ALSP) (Weatherly et al. 1991).

ALSP had to deal with a huge variety of possible aggregated units and therefore also with a even higher variety of interactions. It was therefore no longer possible to focus exclusively on the exchanged information, but the type of objects and interactions needed to be agreed upon as well. This was done in an interface control document (ICD) that enumerated objects and possible messages in a well-formed agreed form. Further, the problem of consistency was addressed by introducing the concept of *ghosted* objects: an object under control of another simulation system was not replicated in the other systems, but only ghosted, so that no inconsistencies could occur. The third big idea introduced by ALSP was a simulation interoperability infrastructure that provided additional services beside the mere transport of information, such as time management. ALSP was successfully used to enable the worldwide Joint Training Confederation (JTC). JTC has been used to support exercises comprising exercise and simulation sites in the United States, Germany, Korea, and Japan. In 1997, twelve simulation systems from varied armed services participated in a JTC worldwide exercise (Prochnow et al. 1997). Although it was never standardized, it developed many ideas that influenced later standardization efforts, in particular the IEEE 1516 High Level Architecture.

These success stories of DIS and ALSP were noticed by the US Congress. They recognized that distributed simulation technology could provide significant benefits to the Department of Defense (DoD). The development of common simulation interoperability solutions and related standards became a high priority. After a review of existing solutions resulted in the insight that the available standards were insufficient for a DoD wide standardization effort, the architecture management group (AMG) was established and funded in 1995 to develop the High Level Architecture (HLA), resulting in the widely used HLA 1.3 NG specification. This was generalized in the standardization process for the IEEE 1516-2000 HLA standard family, which was further developed into the IEEE 1516-2010 versions of these simulation interoperability standard. This standard has been academically supported by textbooks (Kuhl et al. 1999; Topçu and Oğuztüzün 2017) and many tutorials.

The objective for HLA was to merge the lessons learned from DIS and ALSP resulting a general purpose simulation interoperability standard. The result was a set of 10 rules governing the interplay between the composition of participating simulation systems, the so called federation, and each individual simulation system, the so called federates. These rules define how to exchange information, who is responsible for the object and interactions invoking the information exchange, and other high level rules. A second part of the standard defines the details of the interface between the so called Runtime Infrastructure (RTI) and federates. The RTI provides a set of services supporting the execution of federations, namely

- Federation management: Creating, joining, and managing federations, saving and restoring federations, and synchronizing federates for saving and restoring operation.
- Declaration management: Defining publication and subscription of information exchange element types for the federate, i.e., declaring what kind of information the federate is interested in publishing itself or receiving from other federates.
- Object management: Defining the use of instantiated objects and interactions information exchange element objects for the federate, i.e., this is the real information exchange regarding instantiated objects and interactions of the type declared before.
- Ownership management:Defining ownership of objects, including how to transfer it, as only at most one federate is allowed to have ownership of the attributes of an object.
- Time management: Defining the time paradigms and their synchronization, which addressed all possible time paradigms utilized in parallel and distributed simulation systems (Fujimoto 1998).
- Data distribution management: Defining constraints allowing for the optimization of data traffic between federates by defining dimensions of areas to which federates can subscribe. These can be geographic areas, such as the region around an object of interest, as well as logical regions.

The information exchange is captured in a third volume of the standard, which defines the Object Model Template (OMT). The template provides means to define persistent objects with their attributes as well as transient interactions with their parameters. The OMT provides a set of tables that allow to unambiguously define all the data necessary for the definition of data types, transportation constraints, and even a lexicon for all terms utilized in the OMT (Lutz et al. 1998). The standard differentiates between the definition of the information exchange that can be supported by a single federate (the simulation object model (SOM)), support of the information exchange within the composition (the federation object model (FOM)), and the standardized management object model (MOM) that provides information objects needed for all the services provided by the RTI.

In order to support an easier migration from DIS to HLA, the community worked on a common representation of DIS PDUs in the HLA format, the Realtime Platform Reference Federation Object Model (RPR-FOM) that was used as the foundation for many follow on activities, as it combined the flexibility of HLA with the well-known information exchange objects of DIS (Möller et al. 2014).

What is often not covered is the fact that HLA was designed one of three pillars of a *Common Technical Framework* that comprised of a set of data standards to unambiguously define data to describe the battle sphere and its objects – which was one of the reasons the Synthetic Environment Data Representation and Interchange Specification (SEDRIS) development project was funded by the DoD (Foley et al. 1998) – as well as a conceptual description of the mission space that needed to be supported (Pace 2000) – which resulted in the two projects Conceptual Model of the Mission Space (CMMS), a data model driven modeling approach, and Functional Description of the Mission Space (FDMS), an object-oriented approach. The idea migrated into the Knowledge Integration Resource Center (KIRC), but never was really embraced by the community. An overview of these development is captured in (Pace et al. 2011).

2.3 Non-standardized Contributions

While the HLA objective was to be a general simulation interoperability standard, resulting in a relatively broad and open standard that left a lot of room for interpretation in the application in the defense domain, the *Test and Training Enabling Architecture* (TENA) chose a different path (Powell and Noseworthy 2012). TENA purposefully focused exclusively on the support of test ranges for military applications, allowing to use highly efficient but domain specific solutions.

To avoid ambiguities, the approach included an object-oriented Logical Range Object Model. Furthermore, TENA provided not only a common architecture, but also a common language and a common communication mechanism. In addition, TENA ensures the common context in form of a common understanding of the environment, a common understanding of time, and a set of common technical processes. As a result, the TENA infrastructure provided fully integrated solutions, and even a repository for reusable components. However, this exclusive focus on military ranges prohibits the easy migration to other application domains.

In parallel to this mostly DoD driven specific simulation interoperability efforts, which were actively supported by the newly established *NATO Modelling and Simulation Group* (MSG), additional ideas were derived from other standardization activities in the wider distributed systems community, such as the model driven engineering approaches - as recently compiled in (Topçu et al. 2016), and the semantic web, as promoted by the Extensible Modeling & Simulation Framework (XMSF) (Brutzman et al. 2002; Blais et al. 2005) and other academic research endeavors (Page et al. 2000). Some approaches, like (Tolk and Mittal 2014), asked for a radical change of thinking about the interoperability problem, recommending to separate the status information of objects from the process definitions of their change, and by providing unique definitions for both, allowing the composition of simulation from reusable objects and processes as required for the simulation experiment of interest. Currently, the NATO MSG–136 activity is evaluating the possibility of providing M&S as a service (MSaaS) as a means of delivering M&S applications, capabilities and associated data on demand by providers to qualified consumers, bringing national, NATO, and mission clouds together under a common framework (Siegfried and van den Berg 2015). The taxonomy used for the alignment of data and harmonization of processes was purposefully build on top of the Consultation, Command, and Control (C3) taxonomy used within the NATO C3 community.

3 FOUNDATIONS FOR SIMULATION INTEROPERABILITY

In this section, I will mainly focus on the contributions of my research team, without claiming completeness or exclusiveness. Over the last years, many valuable contributions regarding the development of interoperable and reusable simulation solutions have been published, including the important topic of conceptual modeling as well as the development of simulation components.

3.1 The Levels of Conceptual Interoperability Model

Since its first publication in (Tolk and Muguira 2003), the *Levels of Conceptual Interoperability Model* (LCIM) was extended and improved using recommendations and criticisms of our peers and colleagues into the seven layers depicted in the following figure.

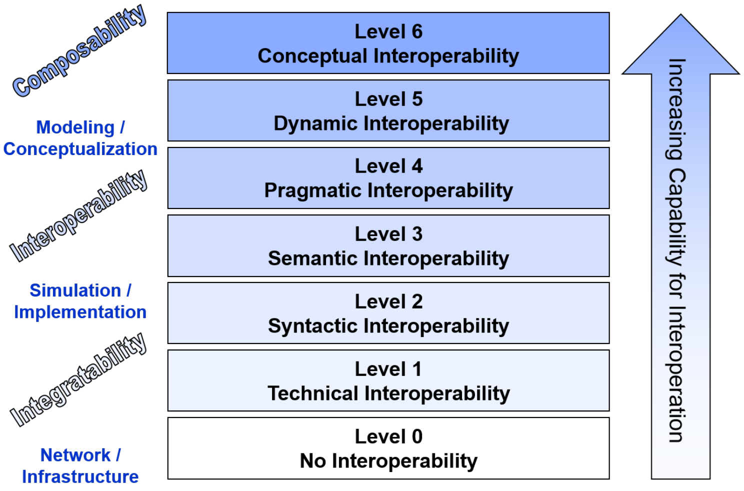

Figure 2: Levels of Conceptual Interoperability Model.

The definition of these layers, which are used to cluster problem classes of interoperability, can be described as follows:

- Stand-alone systems are not connected and have no interoperability.
- The technical layer deals with infrastructure and network challenges, enabling systems to exchange signals, the carriers of information. In particular in tactical networks or infrastructure solutions supporting life elements, this alone can be a difficult task that by itself may require several sub-layers.
- The syntactic layer deals with challenges to interpret and structure the information to form symbols within protocols. Many challenges have standard solutions today, but new, high-efficient encoding standards developed to support new concepts, like mobile edge-computing or the next generation of the Internet-of-things, may create new challenges.
- The semantic layer provides a common understanding of the information exchange. On this level, the pieces of information that can be composed as objects, messages, and other higher structures are identified. Common taxonomies, such as name spaces, are supporting the unambiguous definition of terms, ensuring the alignment of data.
- The pragmatic layer recognizes the patterns in which data are organized for the information exchange, such as the inputs and outputs of procedures and methods to be called. This is the context in which data are exchanged as applicable information. These groups are often referred to as business objects, as they are identified in business process models or comparable architecture artifacts.

- The dynamic layer recognizes various system states, including the possibility for agile and adaptive systems. The same business object exchanged with different systems can trigger very different state changes. It is also possible that the same information sent to the same system at different times can trigger different responses. This layer provides transparency of the system, but also timing concepts needed to harmonize the processes.
- Finally, assumptions, constraints, and simplifications need to be captured. This happens in the conceptual layer. This layer addresses the harmonization of conceptualizations. The conceptual layer is the formal layer of the conceptual model that allows to identify any concepts that are important for the simulation experiment that may be hidden for the user, but may lead to inconsistencies within a composition.

These levels are well aligned with the three governing concepts of interoperation proposed in (Page et al. 2004). The concept of integratability contends with the physical/technical realms of connections between systems, which include hardware and firmware, protocols, networks, etc. Interoperability contends with the software and implementation details of simulation implementations; this includes exchange of data elements via interfaces, the use of middleware, mapping to common information exchange models, etc. Finally, composability contends with the alignment of issues on the modeling level, the conceptualizations underlying the implemented solution. *Successful interoperation of solutions requires integratability of infrastructures, interoperability of systems, and composability of models.* In (Tolk et al. 2013), we extended the LCIM towards an interoperability maturity model and also showed the relation to a variety of other interoperability efforts within NATO.

The sixth level introduces means for synchronizing and harmonizing processes. In practice, two simulation systems do not have to be in sync all the time, but only when they have to exchange data and/or are ask to provide an answer to a certain task, which should be consistent. To ensure these assertions, the two simulation systems can be interpreted as coupled state transition systems, allowing to apply the concepts of bi-simulation to ensure that both systems are consistent in these points, as discussed in (Szabo et al. 2009).

The seventh level of the LCIM was always hard to grab intuitively. We described it as capturing the underlying common theory from a philosophy of science viewpoint, or the collection of common assumptions and constraints that define the validity of all supporting components. The explanation accepted by scholars was a collection of artifacts ensuring the consistent representation of truth, i.e., if the same question is ask two different components, the answer should either by consistent, or there should be no answer at all. In other words: if an object and/or action was within the realm of several component, the representation should be consistent in all of them. For practitioners, this could best be guaranteed by a ”blueprint” representing the simulation experiment – such as a mission of a military operations, showing who is doing what, where, and when, or a detailed business plan, well defined medical procedures, etc. The conceptual elements of this blue print would than be implemented by available simulation components. The ideas of using such a blueprint are explained in some more detail in (Tolk et al. 2007).

At the time of writing this paper, the various papers on the LCIM are referenced in more than 800 scholastic contributions. The LCIM has been successfully applied in several diverse application domains, such as medical simulations (Weininger et al. 2016), the Internet-of-things (Kolbe et al. 2017), manufacturing systems (Panetto and Molina 2008), and even the digital library ecosystem (Kostelic 2017). During our work on formalizing the different levels better, we discovered model theory as a branch of mathematics, that allows to elegantly address various challenges.

3.2 Model Theory and Interoperability

Several experts published their conviction that it is possible to engineer interoperability between two simulation systems; examples can be found in (Taylor et al. 2015). My research team did not agree, but without a formal representation, we could not prove it. After looking at set theory for a formal way to express the LCIM, it was Saikou Diallo who discovered model theory and spearheaded the effort to apply it to better understand simulation interoperability (Diallo et al. 2014).

Model theory is a subset of mathematics that applies logic to formal structures, such as defined by sets, enumerations, or formal languages (Weiss and DMello 1997). A model collects all the information needed to decide if a statement is true in its context, i.e., if the statement is a member of the applicable sets or the enumeration, or of it can be generated by the formal language. If the statement is true, it is satisfied in the model. This allows model theory to treat mathematical truth as relative: the same statement may be true or false, depending on how and where it is interpreted. As computer languages are formal languages, and as simulation systems are programed in computer languages, the results regarding truth representation in formal languages can be applied to consistent representation of truth within computer simulations. As truth regarding the same facts and interpretations need to be consistent within M&S applications, the research findings are significant for understanding interoperability and composability challenges in order to address them when selecting simulations to be federated in support of an exercise, operation, or any other simulation application domain.

Model theory works as follows: A language $\mathscr{L}$ is a set consisting of all the logical symbols with perhaps some constant, function and/or relational symbols included. A model, sometimes also called a structure, $\mathscr{U}$ for a language $\mathscr{L}$ is an ordered pair of the universe $\mathscr{A}$, which is a nonempty set, and an interpretation function $\mathscr{I}$ with its domain being the set of all constant, function and relation symbols of $\mathscr{L}$. The interpretation function maps each constant symbol to a constant, each function symbol to a function, and each relation symbol to a relation. A sentence is an assertion that can be assigned the Boolean value of true or false. And, finally, if $\mathscr{U}$ is a model of $\mathscr{L}$, the theory of $\mathscr{U}$ is defined to be the set of all sentences of $\mathscr{L}$ which are true in $\mathscr{U}$. Simulations can be expressed by such a language $\mathscr{L}$ and the interpretation function $\mathscr{I}$. If they are not consistent under this model theory constructs, they are also not consistent in a composition, as they will compute different results. Therefore, two results of model theory, captured as theorems, are of particular interest for interoperability challenges.

Robinson Consistency Theorem simply states that the union of two theories is satisfiable under a model if and only if their intersections are consistent, in other words: there is only one interpretation of truth valid in both models where they overlap. If this is not the case, there will be inconsistencies!

As it is possible that two theories are using different languages and the resulting sentences are not directly comparable, *Łoś Theorem* generalizes the idea of expanding a universe through the Cartesian product and defines filters that allow the comparison in a common equivalent representation: if two simulation systems using different data to represent the simulated entities and their actions can be mediated into a common language to make them comparable, and if they are inconsistent in their overlap, the federation will show inconsistencies as well.

An interesting aspect is that concepts that are captured in two components but that are not shared via the simulation interoperability protocol – such as objects and interactions in case of HLA, or a PDU in case of DIS – have to be aligned nonetheless, or the model becomes inconsistent. This is a significant insight showing a weakness in all addressed simulation interoperability standards addressed in this paper.

As shown in (Diallo et al. 2014), these model theoretic insights provide the mathematical foundations needed to unambiguously capture the artifacts needed for the seventh layer of the LCIM: conceptual interoperability. This furthermore motivated the definition proposed in my contribution to (Taylor et al. 2015): *Interoperability is the ability to exchange information and to use the data exchanged in the receiving system. Composability is the consistent representation of truth in all participating systems.* It is the reliance on the model underlying the simulation implementation that is responsible for the elusiveness of simulation interoperability: it makes little sense without model composability. What was intuitively motivated first could now be proven.

4 CONCLUSION AND DISCUSSION

Interoperability and composability are not values in themselves, but they enhance reusability, comprehensibility, shareability, and reproducibility of solutions and results. Simulation interoperability standards should therefore be rooted in model theoretic insights. That this is a reasonable requirement is implied by the use of model theory to define the web ontology language OWL consistently (Horrocks et al. 2003). The same rigor needs to be applied to simulation standards. That also explains why current solutions are following short, as they do not capture the conceptual domain of the model sufficiently well, and instead trying to solve the problem technically by focusing exclusively on the simulation level.

4.1 The way to Standardizable Composability Solutions

In practice, the need to compose or federate solutions out of pre-existent and independently developed components that do not fulfill the rigorous requirements of composability. As many of these legacy solutions represent a significant financial assets resulting from several man-years of development, a migration path will be required. Based on the research presented in this paper, the following steps should be part of migration path.

Metadata: The lack of transparency is one of the main hurdles of composable solutions. The conceptualization underlaying the simulation solution needs to be documented, preferably in machine understandable form (Robinson et al. 2015). While the object-oriented approach has many advantages for programming, the concepts of encapsulation is counterproductive for transparent solution, as common concepts may be implemented inconsistently in participating solutions without ever showing up in the interface of the components.

Common Initialization: A first step towards a common understanding of the simulation experiments is the use of a common initialization language, such as the military scenario definition language (Wittman Jr. 2012). While the various participating solutions may still have special interpretations of attributed objects and their activities within the situated virtual environment, the common language at least provides a common starting point.

Common Services: The exchange of attributes describing the objects of interest is well covered in current solutions. However, the processes that change these attributes are often hidden within the components and result in unfair fight situations (Turnitsa 2012). The use of common services to compute processes of common interest, such as weapon effect servers, weather servers, and terrain services in the military domain (Neugebauer et al. 2009) ensures that no systemic bias results in unfair fight computations.

Object and Process Servers Truly composable solutions require that attributed objects and processes are both provided by servers that ensure the unambiguous implementation of the conceptualization that drives the simulation experiments, e.g., as envisioned in (Tolk and Mittal 2014) in the form of mobile propertied agents representing the objects that are manipulated by common services.

There are many additional steps needed, but these four may guide the way for simulation standardization efforts that allow to preserve legacy investments into the reuse and migration of simulation solutions while promoting fully composable component developments in parallel.

4.2 Competitive and Complementary Solutions

Another discussion is necessary in this context as well. Inconsistencies in models is often interpreted as something that is undesired. This is not always the case. Modeling is the task-driven purposeful simplification and abstraction of a perception of reality (Tolk 2013), so it is likely that different models may have different viewpoints or illuminate different aspects of the problem. In this case, variety of models is not inconsistency, but diversity of solutions. As computer simulations have to be consistent, we do need several simulation solutions implementing the different views to allow for competition of these diverse ideas. Complementary solutions should be consistent in their overlapping domain, but add additional information to the composed solution. They differ in scope, so that the composition adds additional angles and facets to look at the problem.

As discussed in (Tolk et al. 2013), ontologies can be used to capture competitive and complementary ideas in one common reference model. This reference model comprises all aspects of a simulation experiment, in particular those representing mutual exclusive viewpoints, resulting in a complete, but not consistent representation of the knoweldge about the experiment. Using model-transformation, it is then possible to derive consistent conceptual models that can be implemented in simulation solutions, each based on a consistent set of the aforementioned viewpoints. Like a meteorologist uses various weather models to forecast different possible paths of a hurricane, this approach allows utilizing a diverse set of models to evaluate different possibilities under uncertainty of which model is actually the best in the current situation.

In conclusion, simulation standards – as shown using the example of simulation interoperability standards – should be rooted in a clear mathematical understanding of the needed of alignment of data and harmonization of processes on the conceptual modeling level as well as the simulation implementation level. Applying software and computer engineering standards is a good practice, but is not sufficient, as they do not address the need for conceptual alignment which is unique to M&S interoperability and applications.

Disclaimer

The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions, or viewpoints expressed by the author. This paper has been approved for Public Release; Distribution Unlimited; Case Number 17-3081-19.

REFERENCES

Blais, C., D. Brutzman, D. Drake, D. Moen, K. Morse, M. Pullen, and A. Tolk. 2005. "Extensible Modeling and Simulation Framework (XMSF) 2004 Project Summary Report". Technical report, Monterey, California: Naval Postgraduate School.

Brutzman, D., M. Zyda, J. M. Pullen, K. L. Morse, S. Fouskarinis, D. Drake, D. Moen, C. Blais, A. Kapolka, and D. McGregor. 2002. "Extensible Modeling and Simulation Framework (XMSF) Challenges for Web-based Modeling and Simulation". Technical report, Monterey, California. Naval Postgraduate School.

Cosby, L. 1995. "SIMNET: An iInsider's Perspective, IDA Document D-1661". *Institute for Defense Analyses, Defense Technical Information Center, Washington, DC*.

Diallo, S. Y., J. J. Padilla, R. Gore, H. Herencia-Zapana, and A. Tolk. 2014. "Toward a Formalism of Modeling and Simulation using Model Theory". *Complexity* 19(3):56–63.

Dunn, W., E. Deutsch, J. Maxworthy, K. Gallo, Y. Dong, J. Manos, T. Pendergrass, and V. Brazil. 2013. "Systems Integration". In *The Comprehensive Textbook of Healthcare Simulation*, edited by A. I. Levine, S. DeMaria Jr, A. D. Schwartz, and A. J. Sim. Berlin, Germany: Springer Science & Business Media.

Foley, P., F. Mamaghani, and P. Birkel. 1998. "The Synthetic Environment Data Representation and Interchange Specification (SEDRIS) Development Project". In *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*. Orlando, FL.

Fujimoto, R. M. 1998. "Time Management in the High Level Architecture". *Simulation* 71(6):388–400.

Hill, R. R., and A. Tolk. 2017. "A History of Military Computer Simulation". In *Advances in Modeling and Simulation*, edited by A. Tolk, J. Fowler, G. Shao, and E. Yücesan. Cham, Switzerland: Springer International Publishing AG.

Horrocks, I., P. F. Patel-Schneider, and F. Van Harmelen. 2003. "From SHIQ and RDF to OWL: The Making of a Web Ontology Language". *Web Semantics: Science, Services and Agents on the World Wide Web* 1(1):7–26.

Kolbe, N., J. Robert, S. Kubler, and Y. Le Traon. 2017. "PROFICIENT: Productivity Tool for Semantic Interoperability in an Open IoT Ecosystem". In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. Melbourne, Australia. November 7 - 10.

Kostelic, C. S. 2017. "Applying the Levels of Conceptual Interoperability Model to a Digital Library Ecosystem–A Case Study". In *International Conference on Dublin Core and Metadata Applications*, 62–72. Washington, DC.

Kuhl, F., R. Weatherly, and J. Dahmann. 1999. *Creating Computer Simulation Systems: An Introduction to the High Level Architecture*. Upper Saddle River, NJ: Prentice Hall.

Lutz, R., R. Scrudder, and J. Graffagnini. 1998. "High Level Architecture Object Model Development and Supporting Tools". *Simulation* 71(6):401–409.

Möller, B., A. Dubois, P. Le Leydour, and R. Verhage. 2014. "RPR FOM 2.0: A Federation Object Model for Defense Simulations". In *Proceedings of the Fall Simulation Interoperability Workshop*. Orlando, FL: SISO Inc.

Neugebauer, E., D. Nitsch, and O. Henne. 2009. "Architecture for a Distributed Integrated Testbed". In *NATO Technical Report RTO MP-MSG-069*.

Office of Technology Assessment (OTA) 1995. *Distributed Interaction Simulation of Combat, Report OTA-BP-ISS-151*. Washington D.C.: U.S. Congress, U.S. Government Printing Office.

Pace, D. K. 2000. "Ideas about Simulation Conceptual Model Development". *Johns Hopkins APL Technical Digest* 21(3):327–336.

Pace, D. K., S. Robinson, R. Brooks, K. Kotiadis, and D. van der Zee. 2011. "Conceptual Modeling Evolution within US Defense Communities: The View from the Simulation Interoperability Workshop". In *Conceptual Modeling for Discrete-Event Simulation*, edited by S. Robinson, R. Brooks, K. Kotiadis, and D.-J. van der Zee, 423–449. Boca Raton, FL: CRC Press.

Page, E. H., R. Briggs, and J. A. Tufarolo. 2004. "Toward a Family of Maturity Models for the Simulation Interconnection Problem". In *Proceedings of the Spring Simulation Interoperability Workshop*. Orlando, FL: SISO Inc.

Page, E. H., A. Buss, P. A. Fishwick, K. J. Healy, R. E. Nance, and R. J. Paul. 2000. "Web-based Simulation: Revolution or Evolution?". *ACM Transactions on Modeling and Computer Simulation* 10(1):3–17.

Panetto, H., and A. Molina. 2008. "Enterprise Integration and Interoperability in Manufacturing Systems: Trends and Issues". *Computers in Industry* 59(7):641–646.

Powell, E. T., and J. R. Noseworthy. 2012. "The Test and Training Enabling Architecture (TENA)". In *Engineering Principles of Combat Modeling and Distributed Simulation*, edited by A. Tolk, 449–477. Hoboken, NJ: John Wiley & Sons Inc.

Prochnow, D., E. Page, and M. C. Fischer. 1997. "Management of the Joint Training Confederation Family of Specifications". In *Proceedings of the Simulation Interoperability Workshop*. Orlando, FL: SISO Inc.

Robinson, S., G. Arbez, L. G. Birta, A. Tolk, and G. Wagner. 2015. "Conceptual Modeling: Definition, Purpose and Benefits". In *Proceedings of the 2015 Winter Simulation Conference*, edited by L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, 2812–2826. Piscataway, NJ: IEEE.

Siegfried, R., and T. van den Berg. 2015. "M&S as a Service: Paradigm for Future Simulation Environments". In *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)*. Orlando, FL.

Szabo, C., Y. M. Teo, and S. See. 2009. "A Time-based Formalism for the Validation of Semantic Composability". In *Proceedings of the 2009 Winter Simulation Conference*, edited by M. D. Rossetti, R. R. Hill, B. Johansson, A. Dunkin, and R. G. Ingalls, 1411–1422. Piscataway, NJ: IEEE.

Taylor, S. J., A. Khan, K. L. Morse, A. Tolk, L. Yilmaz, J. Zander, and P. J. Mosterman. 2015. “Grand Challenges for Modeling and Simulation: Simulation Everywhere: From Cyberinfrastructure to Clouds to Citizens”. *Simulation* 91(7):648–665.

Tolk, A. 2013. “Interoperability, Composability, and their Implications for Distributed Simulation: Towards Mathematical Foundations of Simulation Interoperability”. In *Proceedings of the 2013 IEEE/ACM 17th International Symposium on Distributed Simulation and Real Time Applications*, 3–9. Delft, The Netherlands: IEEE Computer Society.

Tolk, A., L. J. Bair, and S. Y. Diallo. 2013. “Supporting Network Enabled Capability by Extending the Levels of Conceptual Interoperability Model to an Interoperability Maturity Model”. *The Journal of Defense Modeling and Simulation* 10(2):145–160.

Tolk, A., S. Y. Diallo, J. J. Padilla, and H. Herencia-Zapana. 2013. “Reference Modelling in Support of M&S–Foundations and Applications”. *Journal of Simulation* 7(2):69–82.

Tolk, A., and S. Mittal. 2014. “A Necessary Paradigm Change to enable Composable Cloud-based M&S Services”. In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S. Y. Diallo, I. O. Ryzhov, L. Yilmaz, S. Buckley, and J. A. Miller, 356–366. Piscataway, NJ: IEEE.

Tolk, A., and J. A. Muguira. 2003. “The Levels of Conceptual Interoperability Model (LCIM)”. In *Proceedings of the Fall Simulation Interoperability Workshop*. Orlando, FL: SISO Inc.

Tolk, A., C. D. Turnitsa, and S. Y. Diallo. 2007. “Model-Based Alignment and Orchestration of Heterogeneous Homeland Security Applications Enabling Composition of System of Systems”. In *Proceedings of the 2007 Winter Simulation Conference*, edited by S. G. Henderson, B. Biller, M.-H. Hsieh, J. Shortle, J. D. Tew, and R. R. Barton, 842–850. Piscataway, NJ: IEEE.

Topçu, O., U. Durak, H. Oğuztüzün, and L. Yilmaz. 2016. *Distributed Simulation: A Model Driven Engineering Approach*. Cham, Switzerland: Springer International Publishing AG.

Topçu, O., and H. Oğuztüzün. 2017. *Guide to Distributed Simulation with HLA*. Cham, Switzerland: Springer International Publishing AG.

Turnitsa, C. D. 2012. *Exploring the Components of Dynamic Modeling Techniques*. PhD thesis, Modeling, Simulation, and Visualization Engineering, Old Dominion University, Norfolk, VA.

Weatherly, R., D. Seidel, and J. Weissman. 1991. “Aggregate Level Simulation Protocol”. In *Proceedings of the Summer Computer Simulation Conference*. Baltimore, MD: Society for Computer Simulation.

Weininger, S., M. B. Jaffe, M. Robkin, T. Rausch, D. Arney, and J. M. Goldman. 2016. “The Importance of State and Context in Safe Interoperable Medical Systems”. *IEEE Journal of Translational Engineering in Health and Medicine* 4:1–10.

Weiss, W., and C. DMello. 1997. *Fundamentals of Model Theory*. Department of Mathematics, University of Toronto, Canada.

Wittman Jr., R. L. 2012. “Standards-based Combat Simulation Initializing using the Military Scenario Definition Language (MSDL)”. In *Engineering Principles of Combat Modeling and Distributed Simulation*, edited by A. Tolk, 579–605. Hoboken, NJ: John Wiley & Sons Inc.

AUTHOR’S BIOGRAPHY

ANDREAS TOLK (OrcId 0000-0002-4201-8757) is Technology Integrator for the Modeling, Simulation, Experimentation, and Analytics Division of the MITRE Corporation and Adjunct Professor at Old Dominion University. He holds a PhD and M.Sc. in Computer Science from the University of the Federal Armed Forces in Munich, Germany. He published more than 250 journal articles, book chapters, and conference papers and edited nine textbooks and compendia on Modeling and Simulation and Systems Engineering topics. He is a senior member of ACM and IEEE and a Fellow of SCS. His e-mail address is atolk@mitre.org.