Advancing a Joint Federation Object Model

Dannie Cutts
AEgis Technologies Group Inc.
USJFCOM J7, JTS Division
Suffolk, VA 23435
256.922.0802 x2209
dcutts@aegistg.com

Andy Bowers
The MITRE Corporation
USJFCOM J7, JTS Division
Suffolk, VA 23435
757.673.5718
bowersa@mitre.org

Kevin Brandt
The MITRE Corporation
USJFCOM J7, JTS Division
Suffolk, VA 23435
757.673.5714
brandtk@mitre.org

Keywords:

Federation Object Model, FOM, Base Object Model, BOM, High Level Architecture, HLA, M&S

ABSTRACT: *Over the past ten years, the U.S. military has developed an array of Joint and multi-Service federations to meet specific purposes under slightly different conditions. Each development team independently established a Federation Object Model (FOM) aligned to the specific focus of their federation. The issue under study will identify options and explore the feasibility of reconciling these separate federation object models. Moreover, this paper recommends Modeling and Simulation (M&S) community participation in continuing efforts to develop common approaches, best practices, reusable components, and standards to promote FOM extensibility and reuse.*

1 Introduction

1.1 Motivation

Across M&S communities, substantive interoperability equates to the Argonaut's quest for the Golden Fleece. The High Level Architecture (HLA) for Modeling and Simulation (M&S) promised to foster three fundamental tenets: interoperability, composability and reusability of M&S assets. Interoperability enables composability. Both interoperability and composability facilitate reuse.

The Federation Object Model (FOM) [1] sits at the core of HLA interoperability and reuse. Interoperability and reuse of federates across an array of federations requires alignment or realignment to multiple federation object models. This paper discusses fundamental approaches to the development and alignment of the FOM for federations that inherently share federates and routinely propagate new functionality.

This paper recommends M&S community participation in continuing efforts to develop common approaches, best practices, reusable components, and standards.

1.2 Background

The U.S. Department of Defense modeling and simulation glossary [2] defines *M&S interoperability* as "The ability of a model or simulation to provide services to and accept services from other models and simulations, and to use the services so exchanged to enable them to operate effectively together."

Davis [3] describes *composability* as "The ability to select and assemble components in various combinations to satisfy specific user requirements meaningfully. A defining characteristic of composability is the ability to combine and recombine components into different systems for different purposes." Thus, reuse is feasible only when the assets are interoperable and composable.

Certainly, the HLA promotes interoperability, composability, and reuse of M&S assets. The FOM forms the foundation for HLA interoperability. However, over the past decade, multiple Joint federations have developed divergent object models to satisfy unique needs without due consideration of component reuse. As a result, federate developers must expend limited resources to adapt their simulations to each federation's object model and data structures in order to participate in a federation execution and to be interoperable with the other members of that federation.

This dilemma intensifies when user requirements drive the propagation of new capabilities across federations that share federates, and when needs drive the inclusion of new federates. To meet user needs, evolution of fundamental modeling capability within at least one simulation – either an existing federate or a new federate – is a baseline cost. This baseline development is beyond the scope of this discussion, but reuse of these federates is critical.

Interoperability and reuse of components across Joint federations is the prime focus and immediate beneficiary of this effort. However, the approaches explored are important for a wider community.

2 FOM Interoperability

There are (at least) three approaches to achieving object model interoperability across federations built for similar purposes. First, build FOM gateways between each federate and federation. Second, construct a unitary FOM that encompasses the needs of these "allied" federations: a super-set FOM. Third, develop a reusable library of composable object models that, in aggregate, cover community-wide needs of all federations while supporting use of specific subsets of only those elements needed to satisfy each federation's unique requirements.

2.1 Multiple FOM Connections

Prior to the development of HLA, simulations attempted to achieve interoperability with others by aligning to a common data exchange model. The most notable of these being the object model embodied in the Distributed Interactive Simulation (DIS) protocol. Some simulations and simulators achieved interoperability by realigning internal data structures to the DIS data protocol. Others developed external gateways to translate between internal data and DIS protocol data units. With the advent of the HLA, federate developers have continued those two fundamental approaches to interoperability.

A few simulations have developed internal object models to align with "their federation" – effective for inclusion in a single federation but inflexible in the more general sense. Other simulations developed agile FOM [4, 5] approaches allowing individual federates to adapt to new and different federations by mapping their internal object model to each target federation's FOM. Either approach puts the entire onus and cost of federation alignment on the individual simulation. But while it is necessary, data alignment isn't sufficient for federation interoperability since interface control documents and federation agreements may specify other aspects of interoperability. Hence, depending upon the target federations and the implementation techniques used by a specific federate, data alignment alone may not work or provide the requisite interoperability with emerging functions.

In contrast, many federates developed and extended their own, simulation-unique gateways to their objective federation(s). This second approach provides external gateways or bridges that provide translation of data and functionality external to the simulations. The HLA to DIS gateway is the most widespread example of an external gateway. Note that some gateways employ agile FOM techniques for their mappings, but other functions of the gateway are not amenable to direct data mapping techniques (e.g., aggregation, disaggregation, and dispersal of units, elements, or objects).

Multiple gateways resulted from the successful reuse of critical simulations in multiple federations as specific gateways evolved for each target federation. Given time and resources, this approach works; but, it devours both time and resources. In short, neither of these approaches is scalable to large numbers of FOMs or federations and typically do not provide necessary flexibility.

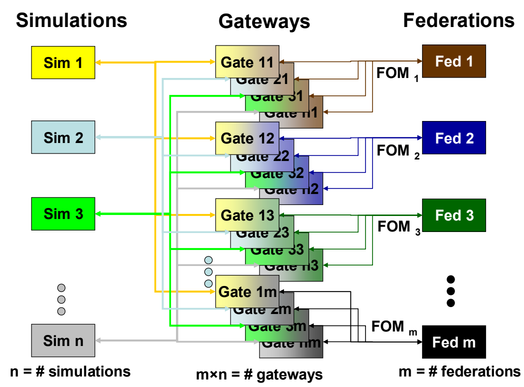

Figure 1: Multiple FOM Gateways

As shown in Figure 1, the upper bound on the "cost" of the multiple gateways approach is a function of several factors:

- Number (n) of simulations,
- Number (m) of federations,
- Number (m×n) of gateways, and
- Number (p) of errors

An expansion in the number of gateways compounds coordination and alignment between multiple federations and simulations. However, gateways provide real benefit. Specified development costs (federate or federation) include potential faults identified and corrected in phase checks. These faults have minimal external impacts. However, when faults persist beyond a phase these errors can propagate to other systems. These public errors have impacts on the costs of external systems. By using gateways, the impact of public errors can be constrained at a limited number of specific gateway(s).

Equation [1] shows this cost relationship as the sum of simulation costs [$\Sigma S_i(\$)$], federation costs [$\Sigma F_j(\$)$], gateway costs [$\Sigma G_{ij}(\$)$], coordination costs [$2\Sigma C_{ij}(\$)$], and error correction costs (in the specific gateways impacted by the error(s)) [$\Sigma p_{ij}G_{ij}(\$)$] and coordination/validation of the correction(s) [$\Sigma p_{ij}C_{ij}(\$)$].

[1] Multiple FOMs with Gateways $\$ = \Sigma S_i(\$) + \Sigma F_j(\$) + \Sigma G_{ij}(\$) + 2\Sigma C_{ij}(\$) + \Sigma p_{ij}C_{ij}(\$) + \Sigma p_{ij}G_{ij}(\$)$, *where* $i = (1 \ldots n)$, *and* $j = (1 \ldots m)$.

2.2 Unitary FOM

In the past decade of HLA development, there has been at least two notable efforts to develop a FOM having broad applicability by a large community across an array of potential federations: a community-wide "Unitary FOM". Undoubtedly, the most successful of these efforts has been the development of the Real-time Platform Reference (RPR) Federation Object Model (FOM). In contrast, the Joint Simulation System (JSIMS) program attempted to establish a unitary Joint FOM intended to encompass all Joint modeling requirements of the U.S. Department of Defense. The commendable JSIMS effort may be a technical success; but the program terminated in the face of exploding costs and programmatic delays.

In our first example of unitary FOM development, the RPR FOM, a broad-based community achieved FOM interoperability between federations by leveraging the existing community-developed DIS protocol as the basis for the underlying federation object model. While unquestionably successful, continued evolution of the DIS protocol and the parallel RPR FOM requires community-wide involvement and consensus: at times a slow, cumbersome process. Moreover, by design, the RPR FOM has limited scope that currently does not extend into modeling many aspects of multi-level resolution domains or non-military parameters. Hence, this unitary FOM will require extensions to expand its scope. Consequently, the effort might be extensive and slow. [1]

In the second example, the JSIMS FOM, the Joint program office expended a significant amount of time, funds, and leadership to construct the JSIMS FOM. By edict, future standardization on this Joint FOM was to occur. When the JSIMS program terminated, the program office archived some of the work but, much of it was lost; and the push for a single Joint FOM dissipated. The point of this example is that this effort was an "all or nothing" endeavor without significant and continuous consideration of the needs or abilities of other Joint simulations or federations. As a result, no other Joint federation has reused the JSIMS FOM in whole or in any significant portions. This JSIMS vignette illustrates the potential expense, fragility, and difficulty anticipated in attempting reuse of a unitary FOM.

These examples support a conjecture that efforts to build a single "universal" Joint FOM that will meet the needs of a community of Joint federations may not be practicable. Objective analysis reveals many of the pitfalls.

First, a unitary Joint FOM must have a broad scope that supports modeling many aspects of multi-level resolution domains, non-military operations, effects-based operations, and non-kinetic actions (e.g. information operations) in addition to the traditional simulation object models (SOM). Unfortunately, the underlying operational concepts for Effects Based Approach to Operations (EBAO) and M&S needs to support those concepts are still evolving … leading to the second issue.

Second, a unitary Joint FOM must be adaptable to meet emerging and evolving needs. Therefore, the FOM will always be a moving target. The demand for frequent changes implies that the FOM will never "settle" and that associated production costs, production errors, and production inertia will continue indefinitely without slack. Hence, unitary FOM approach combines the potential for increased costs with the constraint of its being not sufficiently responsive to the most dynamic federations.

Third, frequent changes, version control, and design trade-offs will challenge coordination and management efforts across the entire community. When a functional need drives a change in any one federate, that change may affect all simulations and all federations. This certainly drives simple additive costs for the first order impacts. However, the changes and costs could "explode" if a change in one federate precipitates changes in others (a potential since there is no intermediary gateways). This potential compounds coordination and alignment between multiple federations and simulations. If coordination errors occur, an explosive shock wave can blast yet another round of changes.

Fourth, federations using large FOMs pay a performance penalty during federation initialization as federates (i.e., data loggers, stealth viewers, etc.) that parse the Object Model Template file (.omt) and/or Federation Execution Data file (.fed) require more time to initialize as FOM size increases. For example, one federation required more than ten minutes to initialize before removing unnecessary classes and interactions. After pruning, the federation initialized in less than two minutes.

Finally, an upper-bound cost assessment comparison reveals that any potential benefits erode if any simulation or federation makes an error in design or if the community fails to include any potential member in the initial rounds of coordination. In either case, these actions necessitate more than one round of coordination efforts that add expense and impede progress.

[1] For more extensive information on the RPR FOM effort and products consult the Simulation Interoperability Standards Organization web at http://www.sisostds.org and activate the link to Product Development Groups and then the subsequent link to the RPR FOM.

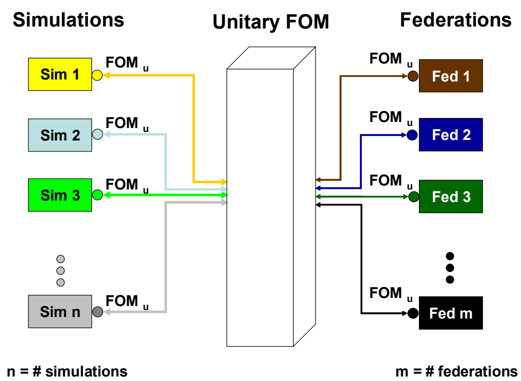

Figure 2: Unitary FOM

As shown in Figure 2, the upper bound on the "cost" of the unitary FOM approach is a function of several factors:

- Number (n) of simulations,
- Number (m) of federations, and
- Number (p) of errors

Equation [2] shows this compound relationship as the sum of the simulation costs [$\Sigma S_i(\$)$], federation costs [$\Sigma F_j(\$)$], coordination costs [$2\Sigma C_{ij}(\$)$], and error coordination costs [$p\Sigma C_{ij}(\$)$], and error correction costs [$p\Sigma S_i(\$)$]+[$p\Sigma F_j(\$)$].

As in the gateway option, the unitary coordination costs are $2\Sigma C_{ij}(\$)$ as the first round designs and implements the new functions needed across federates and the second round validates, assesses, or adjusts to the impact of the "first order" changes. However, in the unitary case, if coordination, design, or implementation errors occur, additional rounds of coordination engage all simulations and federations (i.e., not limited to gateway pairs).

[2] $\text{Unitary FOM } \$ = \Sigma S_i(\$) + p\Sigma S_i(\$) + \Sigma F_j(\$) + p\Sigma F_j(\$) + 2\Sigma C_{ij}(\$) + p\Sigma C_{ij}(\$)$, *where* $i = (1 \ldots n)$, *and* $j = (1 \ldots m)$.

In comparison to Gateway costs, the unitary costs might have a lower bound if the development process has no errors. But, the unitary approach carries more risk since errors of any type will push the costs into subsequent rounds of development. In addition, this approach extracts coordination time delays.

As the RPR FOM effort demonstrates, given time and resources, the unitary FOM approach works; but it devours coordination time and resources. In short, development of unitary Joint FOM is not scalable to a large numbers of federations and it does not provide requisite flexibility.

2.3 Composable FOM

In the face of the limitations inherent in the first two options, the quest for a better solution continues. A third approach exploits composable object models that promote standardization of common components but concurrently adapt to the unique requirements of each federation.

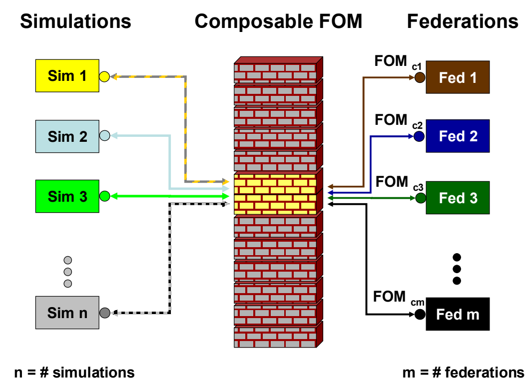

Figure 3: Composable FOM

As shown in Figure 3, the "cost" of the composite FOM approach is a function of several factors:

- The number (n) of simulations,
- The number (m) of federations,
- The number (t) of new object models (O_k) added to the object model library (O_l),
- The number of simulations already having a desired object model (s),
- The number of federations already supporting the object model (r), and
- The number of errors (p)

Equation [3] shows this compound relationship as the sum of the simulation costs [$\Sigma S_i(\$)$], federation costs [$\Sigma F_j(\$)$], coordination costs [$2\Sigma C_i(\$)+2\Sigma C_j(\$)$], object model costs [$\Sigma O_k(\$)$], and error costs [$\Sigma p_k C_i(\$)+\Sigma p_k C_j(\$)+\Sigma p_k O_k(\$)$].

[3] $\text{Composite FOM } \$ = \Sigma S_i(\$) + \Sigma F_j(\$) + 2\Sigma C_i(\$) + 2\Sigma C_j(\$) + \Sigma O_k(\$) + \Sigma p_k C_i(\$) + \Sigma p_k C_j(\$) + \Sigma p_k O_k(\$)$, *where* $i = (1 \ldots (n-s))$, $j = (1 \ldots (m-r))$, *and* $k = (1 \ldots t)$.

The single inheritance, monolithic nature of an HLA FOM has long been a hindrance to realizing a flexible, reusable object model. Recognizing this constraint, the Simulation Interoperability Standards Organization (SISO) commissioned the study and development of a modular FOM standard. [6, 7]

However, true composability requires more than a simple mechanism to pull object model components together to form larger object models. [8] A limited view of composability inhibits both HLA and Test and Training Enabling Architecture (TENA) object model composition approaches.

3 Object Model Composition

Davis [3] identifies the need for high-level conceptual representations that consolidate across three aspects:

- Characterization of components,
- Communications among or about components, and
- Possible compositions

Although Davis is referring to simulation components, the same concept applies to object model components.

Before reaching a conclusion on the advisability of object model composability, factors affecting object model composition warrant discussion.

3.1 Factors Impacting Object Model Composability

Davis [3] lists four factors that influence object model composability:

- Complexity
- Objective and Context
- Science and Technology
- Human Considerations

3.1.1 Complexity

As the complexity of the systems under study increases, the difficulty in building associated models increases. However, the hurdles to modular construction are mere "blips" in comparison to the daunting challenges faced in birthing a monolithic model. Traditionally, engineers have decomposed complex systems into simplifying component structures. This decomposition offers a viable approach to understanding the internal functions of the component modules. Notwithstanding this advantage, insights from complexity science highlight the necessity to explore and understand interrelationships between the modules and the inherent potential for unanticipated, emergent behaviors. If done poorly, modularity can lead to exponentially complex problems. If done well, modularity can lead to smaller, more understandable, workable object model components.

3.1.2 Objective and Context

Good models satisfy specific objectives in specified contexts. When applied outside their bounds, the validity or usability of a model may suffer. Decomposition can frustrate fundamental modeling objectives in the same way. For example, a conceptual model could segment systems along organizational lines, control tiers, application domains, and/or resolution levels. If the disassembly effort slices a model along organization lines (i.e., ground, air, or maritime forces; or friendly, enemy, or neutral forces), then reconstruction (if feasible) exacts extra effort to recombine along a different axis, e.g., level of resolution (i.e., platform or unit). Moreover, a mis-match in fidelity or accuracy of components can result in incompatible compositions. This is particularly acute when attempting to reuse current object models that were not designed modular from inception. Decomposition and reconstruction of these legacy object models presents a significant roadblock.

3.1.3 Science and Technology

This consideration confronts two elements. First, one must determine the validity of each component model or module. Second, one must be able to record and recover these components in formats that facilitate reuse.

Given finite resources and a drive to realize a positive return on investments, program managers generally develop simulations and their underlying object models to meet particular needs. Since most users (and hence most specifications) do not value "theory", few teams are resourced to capture conceptual models and other design documentation. During construction phases, members of the development team make numerous design decisions and select implementation options. Given resource constraints, few ever capture and document all of the "semantic" considerations involved. Fewer still record and convey that information to other potential developers.

Lacking definitive requirements, architectures, designs, and conceptual models, independent teams rarely discover, validate, and reuse "implied" concept models that they extract with effort from the software implementations. Lacking definitive model(s) or design documentation, verification and validation suffers. Lacking objective Verification and Validation (V&V), reuse of simulation components by others remains problematic.

In contrast, if teams anchor validation efforts on a well-defined object model, then their assessments will hold objective merit and their efforts might actually facilitate reuse of the software. Moreover, federation developers might then use object model metadata to assess usefulness of model components for their needs.

To be useful to dynamic development teams, the models and metadata must exist in formats and tools that facilitate their reuse. Tolk [8] identifies an approach to define "*white boxes*" with sufficient transparency to promote reuse and sufficient opacity to protect proprietary implementation of designs that can deliver interfaces that adhere to open-standards. The use of common tools, methodologies, and standards will improve the ability to compose object models to promote interoperability and reuse of simulation components. [8, 9]

3.1.4 Human Considerations

In addition to the previous three factors, people (and the knowledge and understanding they bring to the task) are essential for the appropriate composition of model representations. Several aspects of human involvement in object model composition are worth noting. First we need a common understanding or semantics of the composite system. This includes a mutually understood context and terminology. Today, different teams build individual components. Each may have a clear understanding of the role of "their" component. Unfortunately, they may have only a vague understanding of other components. This shortfall reaches critical mass if they fail to realize how to interact to reach a common goal or how to manage unintended interactions that might frustrate their drive toward common objectives.

3.2 Goals for Object Model Composability

A literature review and in-house "wargaming" sessions identified a significant list of desirable characteristics for any approach to object model composability. Those characteristics include:

3.2.1 Standards

Development teams must use accepted standards, tools, and methodologies to represent object model components in order to achieve long-term benefits. Moreover, teams should employ tools and methodologies common across the M&S community or across the commercial marketplace. Common tools and methodologies allow teams to develop and share object model components. [6, 7, 8]

3.2.2 Conceptual Modeling

A conceptual model provides a description of "what" the object model component represents. Moreover, it can capture assumptions or limitations of those abstractions. It also provides other information to assist users in understanding the model in an implementation-independent manner. Conceptual models assist in object model composition. [3, 6, 7, 8, 9]

3.2.3 M&S Ontology / Lexicon

An established lexicon and ontology for M&S object models supports a shared understanding of the object models. These efforts facilitate both interoperability and composability of object models. [8, 11, 12]

3.2.4 Object Model Repositories / Directories

As object model components emerge into common usage, the community needs to develop a standard approach to storing components so federation developers can easily locate and access them. In addition to the object model, the repository will also hold and/or the directory must reference the metadata, conceptual model, and use cases for the object model. This data should include V&V findings for the object model and use histories. [8]

3.2.5 Reduced Maintenance Costs

Although there will be additional start-up costs to define and document object model components, in the longer term cost savings, cost avoidance, and increased responsiveness must be able to offset the initialization expenses. Initialization costs can be contained in the near term if initial efforts implement short-cycle, spiral development approaches with in-phase validation of modular segments focused on critical, mission-oriented capabilities. Feasible savings accrue because the long-term costs of maintaining the object models should pale in comparison to the expense of maintaining multiple versions of federates and/or gateways for multiple federations.

Cost Matrix

Approach → $ Elements ↓	Multi-FOM & Gateways	Unitary FOM	Composable FOM
Federates (S)	$\Sigma\check{S}_i$, i=1...n	$\Sigma\text{Ş}_i + p\Sigma\text{Ş}_i$, i=1...n	$\Sigma\acute{S}_i$, i=1...(n-s)
Federations (R)	$\Sigma\check{R}_j$, j=1...m	$\Sigma\text{Ŗ}_j + p\Sigma\text{Ŗ}_j$, j=1...m	$\Sigma\acute{R}_j$, j=1...(m-r)
Gateways (G)	$\Sigma G_{ij} + \Sigma p_{ij}G_{ij}$, i=1...n, j=1...m	none	none
Coordination Points (C)	$2\Sigma C_{ij} + \Sigma p_{ij}C_{ij}$, i=1...n, j=1...m	$2\Sigma C_{ij} + p\Sigma C_{ij}$, i=1...n, j=1...m	$2\Sigma C_i + \Sigma p_k C_i + 2\Sigma C_j + \Sigma p_k C_j$, k=0...t i=0...(n-s), j=0...(m-r)
Object Model (O) //OM Library (O_l)	$\epsilon\ S_i$ & $\check{R}_j$ // none	$\epsilon\ \text{Ş}_i$ & Ŗ_j // none	$\Sigma O_k + \Sigma p_k O_k$, k=0...t If $O_k \epsilon O_l \mid O_k(\$)=0$
"Soft" Upper Bound	$= \Sigma\check{S}_i + \Sigma\check{R}_j + \Sigma G_{ij} + 2\Sigma C_{ij} + \Sigma p_{ij}G_{ij} + \Sigma p_{ij}C_{ij}$, i=1...n, j=1...m	$= \Sigma\text{Ş}_i + \Sigma\text{Ŗ}_j + 2\Sigma C_{ij} + p\Sigma\text{Ş}_i + p\Sigma\text{Ŗ}_j + p\Sigma C_{ij}$, i=1...n, j=1...m	$= \Sigma\acute{S}_i + \Sigma\acute{R}_j + 2\Sigma C_i + 2\Sigma C_j + \Sigma O_k + \Sigma p_k C_i + \Sigma p_k C_j + \Sigma p_k O_k$, k=0...t, i=0...(n-s), j=0...(m-r)

Note: additive factors

m = number of Federations (F)
n = number of Federates (S)
p = number of public errors*

r = number of Federations w/ OM prior
s = number of Federates w/ OM prior
t = number of new OM not in OM library

Figure 4: Cost Comparison Matrix

In making a direct comparison between the three options, first note that the highlighted boxes illuminate the points impacted by public errors (those exposed to others). The gateway option limits the impact of public errors to the gateways. The unitary option explodes these costs across all federates and federations. The composable option limits exposure to those federates / federations that must implement the change and to the development of each object model.

At the macro level, note that the cost functions for federates and federations are similar, but not identical.

The potential for error costs make the unitary approach the most expensive over the long term. Likewise, the composable option limits changes to a (potentially) smaller number of federates (n-s) and federations (m-r) to identify it as the least costly. This differential also highlights the value of STANDARDS: if all of the object models within a multi-FOM implementation were standard (i.e. like the RPR), then one could validate once against the standard and not have to re-look (certify) each model for each "new" function even when it has been substantially completed and included previously.

Gateway costs only occur in the multi-FOM option.

Coordination points are most expensive in the unitary option (in the presence of errors) because error coordination occurs across all federations / federates. By using gateways, the multi-FOM option limits error coordination to those federates / federations impacted by the error, but still requires (like the unitary option) initial coordination across all federate / federation mixes (a multiplicative factor). In the composable FOM case, the initial coordination and error correction cases are simply additive (with each federate / federation independently working toward an open or emerging standard).

Object model development costs only incur in the composable option. While this cost is real, the development of standardized object models by community of interest (COI) at large mitigates the costs for each individual program (federate / federation) and enhances interoperability. The scope of these costs and the development of tools / techniques / procedures are the subject of our continuing research efforts.

3.2.6 Fewer Duplications

Reduce duplicate modeling efforts. Leverage standards. Encourage reuse of component object models. Leverage community wide repositories to promote reuse. This would allow federations to focus on those object model components unique to their needs. [13]

3.2.7 Higher Level Interoperability

Support higher level of interoperability by using common object models with well-understood semantics. As sets of common object models emerge, the community must posture to capture their semantics. This meta-data facilitates a common understanding of the semantics of the object model [8, 14, 15]

3.2.8 Support SOA

Emerging open standards and common industry approaches to service oriented architectures (SOA) and web services provide new opportunities to enable reusable interfaces to command and control systems and to share (and/or provide) common services. The business model for SOA parallels the rationale for composite object models for M&S. SOA "provides benefits in four basic categories: reducing integration expense, increasing asset reuse, increasing business agility, and reduction of business risk." [16] We seek these precise benefits in object model composability. Moreover, a web-based community-wide approach to object model composability would enjoy flexibility and efficiency while benefiting from the improved information visibility and resultant consistency, accuracy, and completeness. [8, 15]

3.2.9 Object Models Convergence

Over the long term, the community should strive toward convergence of object modeling across architectures, protocols, and standards. In the Joint arena, this array includes HLA, TENA, Command and Control Information Exchange Data Model (C2IEDM), Battle Management Language (BML) …. While initial efforts focus on developing mission-critical capabilities, the intent is to reconcile Joint federation object models as a step toward a longer-term goal: integration of models across the Live, Virtual and Constructive (LVC) boundaries. This objective moves us toward convergence of HLA, TENA, C2IEDM, and BML. [13, 17]

3.2.10 Facilitate Simulation Composition

The focus of this paper is object model composability. However, that effort is not the end goal. The rationale for this effort is ultimately to enable and promote simulation software composition.

4 Future Requirements

The initial work to "prototype" this composite FOM concept will focus on the development of a "Joint FOM" (composite structure) to solve Joint M&S interoperability problems. We welcome community involvement and seek others across the Joint community to engage and collaborate in the effort. Over the long term, we expect the effort to address interoperability with the TENA Standard Object Models, the C2IEDM ontology, and the BML data model. This endeavor will feed into other efforts: the DoD Live, Virtual, Constructive Architecture Roadmap (LVCAR), and the SISO Study Group on LVC Interoperability.

5 Recommendations and Advances

This paper focuses exclusively on the composability of object models. Other community-wide discussions have considered the wider issues of simulation composability [3, 8, 17]. We conclude that object model composability, as discussed in this paper, advances M&S composability.

Within United States Joint Forces Command (USJFCOM), there are multiple federations with multiple FOMs currently executing. Our alignment efforts begin the process of reconciling and aligning a subset of the Joint object models. This reconciliation process is constrained by continuing requirements to support ongoing federation-supported exercises.

USJFCOM in partnership with Allied Command Transformation (ACT) is working to determine what type of M&S capability is needed for the North Atlantic Treaty Organization (NATO). One possibility is the Joint Multi-Resolution Model (JMRM) Federation using the JMRM FOM. If the JMRM was selected, USJFCOM would move to develop and align the JMRM FOM with open standards to promote Alliance interoperability. ACT anticipates that this standardization process would involve the NATO M&S community. Whatever the decision, we expect to exploit the opportunity to work cooperatively towards satisfying an expressed need for OM standardization across a multinational forum with a wide range of system interoperability requirements.

In the near term, we begin with the Joint Live, Virtual, Constructive (JLVC) and Joint Multi-Resolution Model (JMRM) FOMs as these represent primary Joint federations. We intend to decompose those FOMs using the BomWorks™ tool [14], which supports the SISO Base Object Model (BOM) standard. After decomposition, the effort will identify areas of commonality. The defined end-state, a set of common object model components, will provide the basis for a composite Joint FOM that will support a set of object model components "extensions" unique to the particular needs of each Joint federation. If successful in these endeavors, we envision that over time, the Joint M&S community will iteratively develop common object model components for use within USJFCOM and beyond.

6 References

[1] IEEE 1516.2-2000 – IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) Object Model Template (OMT) Specification, 21 September 2000

[2] U.S. Department of Defense, Modeling & Simulation Glossary, Online M&S Glossary (DoD 5000.59-M), http://www.dmso.mil

[3] Paul K. Davis and Robert H. Anderson, "Improving the Composability of Department of Defense Models and Simulations", Rand Monograph MG101, 2003, http://www.rand.org/pubs/monographs/MG101/index.html

[4] Len Granowetter, "Solving the FOM-Independence Problem", www.siaa.asn.au/get/2395379573.pdf

[5] Ke-Thia Yao, Information Sciences Institute, "Agile Data Logging and Analysis", I/ITSEC 2006, Paper # 2580

[6] SISO Standard SISO-STD-003-2006, Base Object Model (BOM) Template Specification, http://www.sisostds.org

[7] SISO Standard SISO-STD-003.1-2006, Guide for BOM Use and Implementation, http://www.sisostds.org

[8] Andreas Tolk, "Composable Mission Spaces and M&S Repositories – Applicability of Open Standards", 04S-SIW-009, SISO Simulation Interoperability Workshop, April 2004.

[9] Katherine Morse, "Simulation and Component Interoperability", Interoperability Brief to AMG, 13 September 2005

[10] Paul Gustavson and Tram Chase, "Using XML and BOMs to Rapidly Compose Simulations and Simulation Environments", Proceedings of the 2004 Winter Simulation Conference

[11] Mikel D. Petty, "Two Aspects of Composability: Lexicon and Theory", DMSO Workshop on Composable M&S, 9 July 2002

[12] Mikel D. Petty, Eric W. Weisel, "A Composability Lexicon", Spring 2003 Simulation Interoperability Workshop", Paper 03S-SIW-023

[13] Dannie Cutts, Paul Gustavson, & John Ashe, "LVC Interoperability via Application of the Base Object Model (BOM), 2006 I/ITSEC conference

[14] Tram Chase; Paul Gustavson, & Larry Root, "From FOMs to BOMs and Back Again", 06S-SIW-116, April 2006

[15] Mikel Petty, "Semantic Composability and XMSF", Moves Institute Workshop on XMSF, https://www.movesinstitute.org/xmsf/workshop/petty/SemanticComposabilityAndXMSFv1.pdf

[16] Ronald Schmelzer, "The ROI of SOA", Zapflash Document 20050127, http://www.zapthink.com/report.html?id=ZAPFLASH-20050127

[17] Ulrich Schade and Michael Hieb, "Development of Formal Grammars to Support Coalition Command and Control: A Battle Management Language for Orders, Requests and Reports (I-069)", 11th International Command and Control Research and Technology Symposium, Command and Control in the Networked Era

Author Biographies

DANNIE CUTTS is a Senior Computer Scientist with the AEgis Technologies Group Inc. supporting the U.S. Joint Forces Command (USJFCOM) Joint Technology and Simulation Division (JTSD) in Suffolk, VA. He has over 20 years experience in M&S for NASA and the DoD and has been involved with the High Level Architecture (HLA) since 1995 serving on the Interface Specification and Time Management Working Groups. He has provided HLA Training, Cadre support for DMSO, and currently serves on the IEEE Drafting Group for the HLA IEEE 1516 standard. Mr. Cutts has supported numerous federation development efforts as well as projects bringing legacy and new simulations to HLA Compliance. At USJFCOM he is involved in efforts to improve interoperability between Live, Virtual, and Constructive assets for Joint Training.

ANDY BOWERS is a Modeling and Simulation Engineer with MITRE supporting the Joint Forces Command (JFCOM) Joint Technology and Simulation Division (JTSD) in Suffolk, VA. A retired U.S. Army officer, Mr. Bowers has been involved in simulation-supported training for over fifteen years. Mr. Bowers has supported several federation development efforts and is currently the FOM manager for the JMRM and MTWS-JCATS federations. At USJFCOM he supports the JTSD Systems Engineering Branch.

KEVIN BRANDT is a lead systems engineer with MITRE supporting the USJFCOM, Joint Warfighting Center, Joint Technology and Simulation Division. He has over thirteen years of active experience in applied operations research, over fifteen years of experience in development of modeling and simulation systems, and concurrent experience in constructing M&S enterprise architectures for training, analysis, and experimentation. More recently, he has developed and tested integrating concepts for command and control that exploit M&S capabilities to advance issues in complexity science.