

LVC Interoperability via Application of the Base Object Model (BOM)

Dannie Cutts
JFCOM J7, JTS Division
Suffolk, VA 23435
dannie.cutts@att.jfcom.mil

Paul Gustavson
Sim Ventions, Incorporated
Fredricksburg, VA 22408
pgustavson@simventions.com

John Ashe
Aegis Technologies Group
Huntsville, AL 35806
jashe@aegistg.com

ABSTRACT

Substantive interoperability between Live, Virtual, and Constructive (LVC) assets is essential to providing the highest quality warfighter training. However, the current LVC architectures in common use are not interoperable. The High Level Architecture (HLA) is most often used in the M&S community for integrating virtual and constructive assets, while the Test and Training Enabling Architecture (TENA) is widely used to integrate live assets into training exercises. We will discuss both technical and substantive interoperability issues between the architectures.

This paper will propose a strategy for moving toward improved LVC interoperability, and will focus on one aspect of interoperability... namely Object Model interoperability. The paper will explore the feasibility of using the Base Object Model (BOM) as a foundation for bridging the significant deficiencies that exist in the integration of HLA Federation Object Models (FOMs) and TENA Logical Range Object Models (LROMs). Inconsistent object models are a major source of interoperability problems. This paper will cover some of the more common interoperability problems as well as proposing solutions supported by a well designed BOM.

ABOUT THE AUTHORS

Dannie Cutts is a Senior Computer Scientist supporting the Joint Forces Command (JFCOM) Joint Technology and Simulation Division (JTSD) in Suffolk, VA. He has over 20 years experience in Modeling and Simulation (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 1516.1 standard. Mr. Cutts has supported numerous federation development efforts as well as projects bringing legacy and new simulations to HLA Compliance. At JFCOM he is involved in efforts to improve interoperability between Live, Virtual and Constructive assets for Joint Training.

Paul Gustavson is the Chief Scientist for SimVentions, Inc. and long-time advocate of the Base Object Model (BOM) concept and simulation composability. He has over 17 years of computer engineering experience supporting a wide variety of modeling and simulation, software development, and web technology efforts within the DoD and commercial communities. He is also a principal author of "C++ Builder 6 Developers Guide," a contributing author of "C++ Builder 5 Developers Guide," and the technical editor for "SAMS Teach Yourself UML (2nd Edition)".

John Ashe is the Director of Simulation Integration for Aegis Technologies Group, Inc. His organization supports numerous Modeling and Simulation efforts for the U.S. Army and Joint Forces Command. Programs that include: Team OneSAF, the Joint Land Component Constructive Training Capability, Entity Resolution Federation (JLCTC ERF) and the JFCOM ATTG. He has more than 23 years of applied engineering experience in a broad range of disciplines from power generation to satellite design.

LVC Interoperability via Application of the Base Object Model (BOM)

Dannie Cutts
JFCOM J7, JTS Division
Suffolk, VA 23435
dannie.cutts@att.jfcom.mil

Paul Gustavson
Sim Ventions, Incorporated
Fredricksburg, VA 22408
pgustavson@simventions.com

John Ashe
Aegis Technologies Group
Huntsville, AL 35806
jashe@aegistg.com

INTRODUCTION

This paper identifies some of the obstacles to interoperability among existing LVC architectures, and focuses on object modeling as one of the major barriers to LVC interoperability. The paper will explore the feasibility of using the Base Object Model (BOM) as a foundation for composability and for bridging the significant deficiencies that exist in the integration of HLA Federation Object Models (FOMs) and TENA Logical Range Object Models (LROMs). Inconsistent object models are a significant source of interoperability problems and this paper will propose solutions supported by a well designed BOM.

Background

A primary operational objective of the Joint National Training Capability (JNTC) is to provide a Joint world-wide LVC training environment, including integration of joint and service virtual and constructive simulations, opposing force capabilities, and range instrumentation. (DuSD(P&R), 2006; DoD T2, 2006). That Joint LVC training environment will allow warfighters to “train like we fight and fight like we train.” Live, Virtual, Constructive (LVC) capability and integration development is the number four priority in the JNTC Program Goals and Objectives for FY07.

An integrated LVC training environment is critical to providing superior warfighter training. However, the current LVC architectures in common use are not interoperable. The High Level Architecture (HLA) is most often used in the Modeling and Simulation (M&S) community for integrating virtual and constructive assets while the Test and Training Enabling Architecture (TENA) is widely used to integrate live assets into training exercises. The current level of interoperability is achieved through the use of numerous bridges and gateways to translate between HLA, TENA, and other protocols.

These gateways and bridges add complexity, latency, and potential mis-translation of the data. In addition, the gateways and bridges themselves are largely non-interoperable and often inconsistent.

Common Characteristics

Although the two architectures are not natively interoperable, they do share a number of characteristics including:

Use of Object Models

Both TENA and HLA require object models for semantic interoperability. A TENA meta-model is a model of the elements used to construct a TENA LROM (TENA MetaModel 2006). In the same way, the HLA Object Model Template (OMT) describes the elements of an HLA Object Model (OMT, 2000). A TENA LROM represents an interface “contract” for a given logical range while an HLA Federation Object Model (FOM) represents a data “contract” for a given HLA Federation. TENA Object Models in general, offer a much richer schema, supporting a more Object Oriented representation, including composition, which is not currently supported by the HLA OMT. Thus, while both TENA and HLA require an object model, their object model representations are significantly different in structure and application.

Similar Organizational Oversight

The TENA Architecture Management Team (AMT) and the HLA Architecture Management Group (AMG) are each responsible for oversight of their respective architectures. In the early days of HLA, the HLA AMG played a more active role in HLA standards definition, but as the HLA has transitioned to an open international IEEE standard (Rules, 2000; Ifspec, 2000; OMT, 2000; FEDEP, 2003), the AMG’s primary responsibility has been to ensure that the HLA continues to

meet the needs of the DoD M&S community. AMG members represent M&S users across the DoD and play an active role in the IEEE process. Similarly, the TENA AMT plays a very active role in the evolution of the TENA standard. The AMT meets approximately every 3 months and is composed of representatives from the TENA user community.

There are, of course, many individuals involved in both communities who recognize the need for better interoperability between HLA and TENA.

Set of Core Rules

HLA and TENA both define rules which facilitate interoperability within their respective domains. HLA rules are specified in an IEEE Document (Rules, 2000). HLA rules outline the responsibilities of HLA federates and HLA federations to ensure a consistent implementation of the architecture. TENA's technical architecture view specifies rules for using TENA and affiliated standards that assist applications in achieving TENA's technical requirements and broader DoD goals. TENA rules specify three levels of compliance that applications may attain.

Use of Middleware Implementations

A common misconception of HLA and TENA is in respect to implementation. TENA and HLA are both architectures... not implementations. However, to use either TENA or HLA requires an implementation of the architecture. TENA Middleware is the implementation of the communication and delivery infrastructure of the architecture. It is roughly analogous to the HLA Runtime Infrastructure (RTI). Both HLA RTI and TENA Middleware offer a standard Application Programmers Interface (API) through which applications address the infrastructure software.

Although TENA and HLA are similar in some aspects, their native incompatibility is a major inhibitor to seamless LVC interoperability.

INTEROPERABILITY INHIBITORS

There are several key inhibitors to seamless LVC interoperability that should be addressed:

Lack of understanding of the interoperability issues between Live (TENA) and Virtual/Constructive (HLA)

If seamless LVC interoperability is the desired end state, the differences between Live, Virtual and Constructive environments must be thoroughly investigated and documented. Domain specific strategies such as dead reckoning (commonly used in the M&S community to reduce data transmission) will need to be addressed for the live domain. Understanding the issues will allow them to be more readily addressed.

Differences in Intended Use

TENA and HLA were developed for different domains and uses. Zimmerman and Rumford (Zimmerman, 2001) point out that both TENA and HLA were developed with complementary objectives... HLA was intended to provide interoperability among and reuse of Modeling and Simulation (M&S) assets while TENA was intended to provide interoperability among and reuse of test resources. TENA is being widely used to integrate live range assets into training environments. Each was developed to meet the particular needs of its community.

Incompatibilities in Object Modeling

Object modeling has always been a significant obstacle to interoperability and composability, even within a single architecture.

The Distributed Interactive Simulation (DIS) protocol attempted to solve the interoperability problem by developing a single data model to be used by all DIS participants. However, this approach didn't offer the flexibility needed to represent complex, changing and diverse systems. HLA shifted to the other end of the spectrum by specifying a format for recording the object model, but leaving the definition and content of the object model open to the individual developer. This approach offered greater flexibility but introduced a substantial interoperability problem due to many diverse FOMs being developed across the M&S Community. The complexity of integrating an HLA Federation increases significantly when participating simulations have been developed to different HLA Object Models. Within the HLA Community, some effort has been made to develop standard "Reference" FOMs such as the Realtime Platform Reference (RPR) FOM.

TENA specifies its object model format in the TENA Meta-Model (TENA MetaModel, 2005) and also specifies a suite of “Standard Object Models” from which more complex object models can be composed. This approach of Standard Object Model subsets, in our opinion, offers a more acceptable tradeoff between flexibility and standardization. However, both the HLA and TENA object modeling approaches are unique to their specific architecture or protocol.

Lack of Composability

Composability is defined within the Department of Defense (DoD) Modeling and Simulation (M&S) Master Plan (M&S Master Plan, 1995) as “the ability to rapidly select and assemble components to construct meaningful simulation systems to satisfy specific user requirements.”. Such composability is intended to “enable effective integration, interoperability, and reuse.” The dilemma, however, is that we have “not adequately achieved composability across the M&S community” for HLA or TENA and this deficiency is a limiting factor to our ability to achieve interoperability between the architectures (Chase, 2006). For instance, the lack of composability support offered by HLA object modeling has made assembling HLA FOMs from piece parts much more difficult. A FOM serves as the binding contract which allows systems and simulators to exchange meaningful information. If there is difficulty and delay in being able to produce a FOM then the ability for such systems and simulators to interoperate may be at risk. We strongly argue that a better approach to develop usable object models that meet the needs of HLA and TENA... as well as future architectures, would be to seek a single object modeling methodology that is centered upon achieving composability.

Systems Engineering Process

Early experiences (both successes and struggles)

with HLA led to the development of a well defined systems engineering process for building and executing HLA Federations. The resulting HLA Federation Development and Execution Process (FEDEP) (FEDEP, 2003) is a systems engineering process adapted to the development of HLA Federations. Figure 1 illustrates the seven steps of the FEDEP process. Object Modeling is a critical component of every step of the FEDEP.

The FEDEP does not preclude live participation, and in fact, alludes to the use of live assets in an HLA Federation. However, the FEDEP process may need to be re-examined to address any specific requirements (tools, VV&A, Products) related to TENA. TENA has a similar systems engineering process outlined in the TENA Architecture Reference Document (TENA Architecture 2002). In addition, the Joint Mission Environment Test Capability (JMETC) program has adopted the TENA Systems Engineering Process renamed the “JMETC Integration and Customer Support Process”. A single systems engineering approach is desirable and would be a significant enabler for LVC interoperability.

Business Process Incompatibilities

TENA and HLA have adopted very different business strategies. HLA embraced an open international standard with commercial-off-the-shelf (COTS) and government implementations of the architecture. TENA has adopted a single Government-off-the-shelf (GOTS) solution developed by the Government and distributed free of charge. The advantages and disadvantages of each of these approaches should be weighed, and a single DoD wide strategy should be considered... particularly if the two architectures are eventually combined into a single LVC integrating architecture.

Middleware / Infrastructure Incompatibility

Both TENA and HLA implementations provide a communications infrastructure layer with a well-defined user API and a set of services designed to

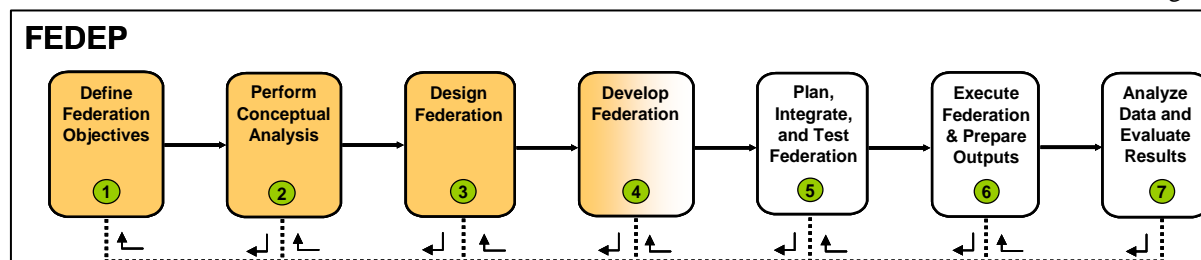


Figure 1 – Federation Development and Execution Process (FEDEP)

distribute data between producers and consumers based on a publish/subscribe paradigm. Although they provide similar message delivery services, they differ in their intended uses. HLA, for example, provides a number of services specifically designed to meet the unique requirements of the M&S community (e.g. Time Management). If the middleware implementations are merged in the future, functionality specific to either the Test or M&S requirements will need to be addressed.

OBJECT MODELING

The intent of this paper is to more fully address the Object Modeling interoperability inhibitor described above. The two major issues are the interoperability constraints, and the lack of seamless composability that HLA, TENA and other architectures exhibit. We argue that if the composability of object models could be achieved for HLA, TENA or other architectures using a single object modeling methodology, then the opportunity for providing interoperability within and across LVC systems and simulations that sit on top of these architectures could be more easily realized. Therefore, as an initial framework to address these needs, what is required is a framework that is both focused on supporting composability and adaptive to different object modeling architectures. The one framework that seems to be the most relevant in supporting the interoperability and composability objectives we have identified is the Base Object Model (BOM). In addition, the Base Object Model has recently been adopted as an M&S community standard within the Simulation Interoperability Standards Organization (SISO).

Base Object Model (BOM) as a unifying approach to object modeling

The Base Object Model (BOM) concept provides a flexible component approach that, based upon our analysis and experience, can be applied for resolving the object modeling issue for both HLA and TENA. It is an ideal candidate because it is specifically intended to encourage “composability”. Consider that a BOM is defined as “a piece part of a conceptual model, simulation object model, or federation object model, which can be used as a building block in the development and/or extension of a simulation or federation.” (BOM Template, 2006) This piece part / building block concept is the modularity capability that is sought for both HLA

and TENA. If BOM modules can be used to define HLA and TENA object models, then BOMs could very well be the enabler to facilitate interoperability across Live Systems within a TENA environment and the Virtual and Constructive Simulations within an HLA environment.

The modularity offered by BOMs, provides the first and most important step for LVC interoperability. According to Paul Davis, Research for RAND Corporation, “Modularity is necessary when dealing with complex systems, and some degree of composability is surely possible and desirable.” (Davis, 2003) He shares that creating a simulation requires breaking the problem into parts that can be addressed separately. In our case those parts can be codified using Base Object Models – or BOMs. As illustrated in Figure 2, a BOM can be made up of four major structure elements: Model Identification, Conceptual Model Definition, Model Mapping and the underlying Object Model Definition.

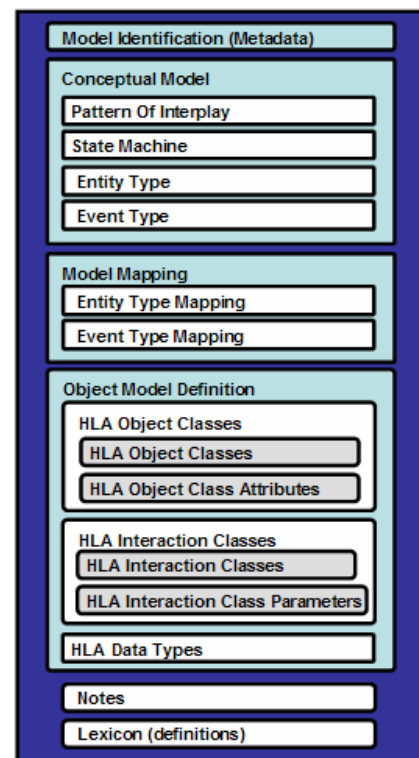


Figure 2 – BOM Elements

The Model Identification element identified in Figure 2 is used for providing the essential metadata for documenting a BOM. Figure 3 provides a view of the metadata attributes found within the Model Identification structure.

Name	POCs *
Type	Type
Version	Name
Modification Date	Organization
Security Classification	Telephone
	Email
Release Restriction *	References *
Purpose	Other
Application Domain	Glyph
Description	Type
Use Limitation	Image
Use History *	Alternate Text
	Height
	Image
Keyword	
Taxonomy/Value *	

* Multiples Allowed

Figure 3 – BOM Metadata Elements

The primary purpose of this metadata is to allow the discovery and understanding of models & components. Such a metadata structure could be prescribed for a number of resources such as describing TENA LROMs, HLA FOMs, software components, scenarios and much more. In fact, this metadata information is intended to provide a generic way to describe and reflect characteristics of information-bearing entities. It is based on the convergence and best practices within the greater commercial and DoD industry – borrowing from initial HLA model identification structure, the Dublin Core, the DDMS and the VVA Recommended Practice Guide. Also included is a mechanism to capture and reflect integration use through “use history” and the ability to graphically represent an item on a tool palette, web page or data base using the Glyph metadata element.

The Conceptual Model Definition element identified in Figure 2 provides a mechanism to capture the conceptual model. Conceptual model information is often the sparsest information available for a model and yet, can be the most useful for understanding a model. By providing a mechanism to capture the conceptual model, it enables effective use of that model for different types of systems (e.g. Live, Virtual or Constructive) and different architectural environments (e.g., HLA, or TENA). Figure 4 illustrates further details of what is offered by the Conceptual Model Definition.

What is not prescribed at this conceptual model layer is a specific object model structure. The developers of the BOM standard, which was represented by a wide variety of M&S implementers and users, felt it was important that the Conceptual Model Definition of a BOM provide a simple mechanism to identify generic Entity Types and Event Types. In addition, they felt it important to be able to capture and identify common Patterns of Interplay, and state machines associated to one or more Entity Types.

An example of an Entity Type might be an “entity that shoots” that we might call “shooter”, or a vehicle or person that carries fuel or ammunition that we might call “supplier”. It is important to recognize that these Entity Types persist in different states. Such states can be documented using a State Machine within the BOM. An example of an Event Type might be a “weapon fire”, “resupply” or “refuel.” From a process view, such Entity Types and Event Types might be part of the design that dictates

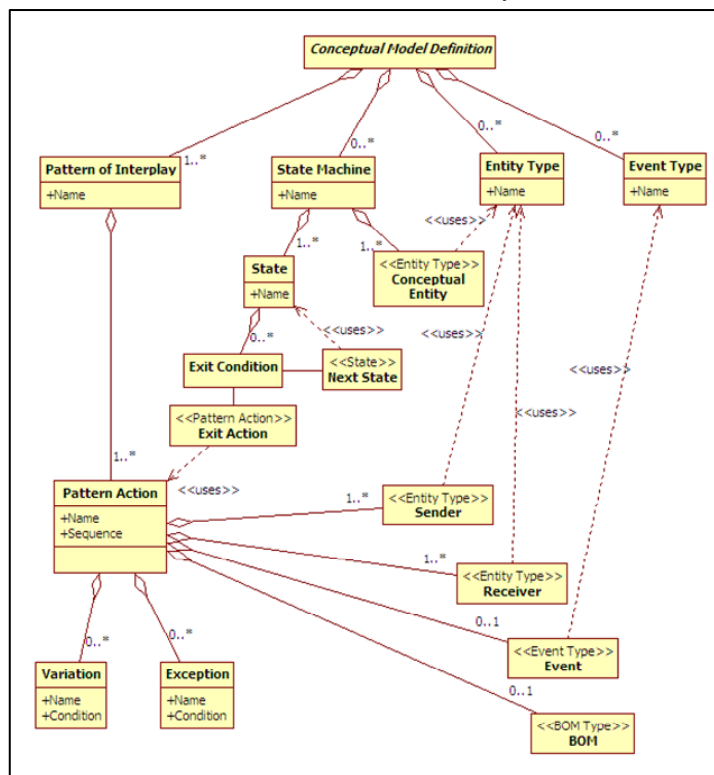


Figure 4 – BOM Conceptual Model Elements

and derives the specific Object classes (and Interactions) that are defined and used by a system or simulation. Such specific Object Classes and Interaction classes can (and should) be mapped to the general Entity Types and Event Types, which is supported using the Model Mapping element

described later. Having this conceptual model information captured and conveyed in a way that can be accessed, viewed and understood by others is what helps encourage reuse. Composition is a powerful enabler for reuse of components.

The Object Model Definition element identified in Figure 2 is used to identify the core class structures intended to be represented by the system, simulation or model. While these elements of the BOM specification are borrowed from the HLA Object Model Template (OMT), it is important to note that this aspect of a BOM is not limited to HLA. The HLA OMT merely provides a common mechanism for describing classes that is understood by the general M&S interoperability community. Also important to point out, is that while the BOM structure identifies multiple elements (i.e., Model Identification, Conceptual Model Definition, Model Mapping, and Object Model Definition), not all these elements are required for use by a single BOM. It is permissible for a BOM to have a Model Identification and only one other component element.

The Model Mapping element also identified Figure 2 may be defined in one BOM and provide linkage to other BOMs. For instance, the Conceptual Model Definition and Model Mapping might exist in one BOM, whereas the specific class structure that can be used to support the Entity Types and Event Types defined in the Conceptual Model Definition may exist in one or more other BOMs.

This ability for BOMs to be loosely coupled, allowing entity types to link externally with specific class structures, is only one aspect of composability offered by the BOM. It is also possible to take a collection of BOMs that describe various patterns of interplay and state machines that are to be exhibited and aggregate them together to constitute a larger, more federation focused object model. This capability is illustrated in Figure 5.

Such a capability to collect and stitch BOMs in this fashion and produce a BOM Assembly, provides a useful mechanism for supporting multiple architectures such as HLA or TENA. In fact, as evidenced by Chase and Gustavson in their paper "From FOMs to BOMs and Back Again" (Chase 2006), existing object models can be decomposed

into more reusable object models, which are defined as BOMs, and then coupled to reformulate the capabilities that were initially offered in the original FOM prior to its decomposition (Chase, 2006). One advantage of this modular approach is that an individual BOM could be changed or swapped with another, without requiring a major editing change to the entire FOM or LROM and with minimal code impact to a system or simulation that uses such a FOM or LROM.

A well-defined BOM can be used within and across the HLA space and the TENA space. Consider the decomposition, reconstitution, and modular exchange capability offered by BOMs, and the ability for a BOM Assembly to serve up compatible HLA FOMs and TENA LROMs. Based on these capabilities, it is sufficient to say that BOMs provide an enabling capability for supporting LVC interoperability.

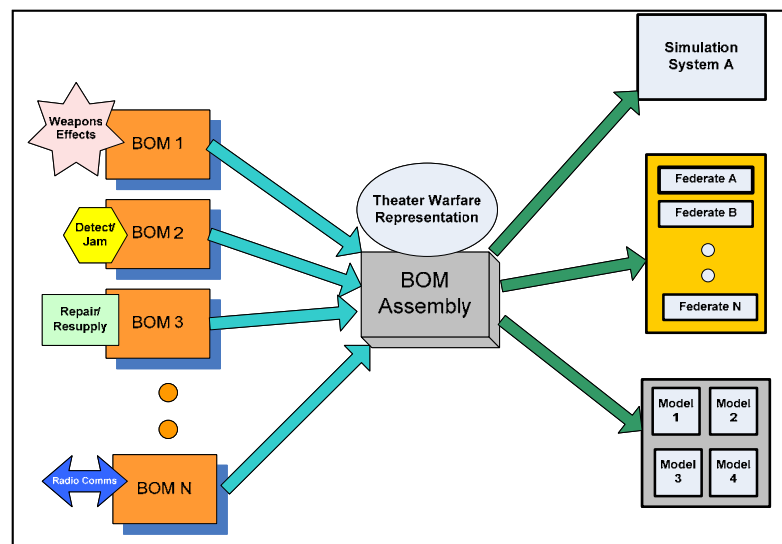


Figure 5 – Composability through BOM Assembly

SUMMARY

Both HLA and TENA have similar goals of providing interoperability between, and reuse of assets. Although their goals are similar, the two architectures are not natively interoperable. Seamless LVC environments will require a higher degree of interoperability between HLA, TENA, and any future architecture. Ultimately, consideration must be given to a "grand convergence" of TENA and HLA into an LVC Integrating Architecture. A critical part of that convergence will involve a single, cohesive, object modeling approach supporting the

full object modeling lifecycle, from conceptual modeling to code generation. We believe that the Base Object Modeling (BOM) approach offers a viable solution to meeting the needs of the Live, Virtual and Constructive communities today and in the future. Whether the existing architectures are merged, or whether they are replaced by a future architecture, a common object modeling approach will offer greatly improved interoperability between Live, Virtual and Constructive environments.

REFERENCES

- DuSD(P&R) (2006), *Strategic Plan for Transforming DoD Training*, May 8, 2006, <http://www.t2net.org>
- DoD T2, 2006, *Department of Defense Training Transformation Implementation Plan FY2006-FY2011*, February 23, 2006, http://www.oft.osd.mil/library/library_files/document_214_Training%20Transformation%20Implementation%20Plan.pdf
- TENA MetaModel, <https://www.tena-sda.org/display/MW/Metamodel>
- Rules (2000), *IEEE 1516-2000 – IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) Framework and Rules*, 21 September 2000, <http://standards.ieee.org/>
- Ifspec (2000), *IEEE 1516.1-2000 – IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) Federate Interface Specification*, 21 September 2000
- OMT (2000), *IEEE 1516.2-2000 – IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) Object Model Template (OMT) Specification*, 21 September 2000
- FEDEP (2003), *IEEE 1516.3-2003 – IEEE Recommended Practice for High Level Architecture (HLA) Federation Development and Execution Process (FEDEP)*, 20 March 2003,
- Zimmerman (2001), Zimmerman, Phil, & George Rumford (2001). *Interoperability Efforts in Modeling & Simulation and in Test & Evaluation. NDIA SBA Conference*, May 16, 2001
- M&S Master Plan (1995), *U.S. DoD Modeling and Simulation (M&S) Master Plan*, DoD 500059-p, October 1995, <https://www.dmsi.mil/public/library/policy/guidance/500059p.pdf>
- Chase (2006), Chase, Tram; Paul Gustavson and Larry Root, “From FOMs to BOMs and Back Again”, 06S-SIW-116, April 2006
- TENA Architecture (2002), *TENA Architecture Reference Document, Version 2002*, 4 November 2002, <https://www.tena-sda.org/download/attachments/6750/TENA2002.pdf?version=1>
- Davis (2003), Davis, Paul K 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>
- BOM Template (2006), *Base Object Model (BOM) Template Specification*, -SISO-STD-003-2006, 8 May 2006, <http://www.sisostds.org>
- BOM Guidance (2006), *Guide for Base Object Model (BOM) Use and Implementation-SISO-STD-003.1-2006*, 8 May 2006, <http://www.sisostds.org>