The DMT Master Conceptual Model

Dr. Tony Valle
SPARTA, Inc.
Orlando, FL
Tony_Valle@trwdmt.com

Bruce McGregor
Northrop Grumman Mission Systems
Orlando, FL
Bruce.McGregor@ngc.com

ABSTRACT

The Distributed Mission Training (DMT) program links high fidelity cockpit simulators and crewmember workstations over a wide-area network into a single virtual battlespace. Simulators in the DMT system have been procured from different vendors and are largely designed to provide the best possible team training for that crew. Achieving interoperability among these disparate simulators and support items is the primary challenge for the DMT Operations and Integration (O&I) contractor.

The key to achieving effective interoperability in the DMT environment is the integration of constructive and virtual simulations and models through the use of a common battlespace content and a consistent set of data protocols. It depends upon a common data model for the battlespace that covers the full scope of required inter-team training simulation content, in a manner that can be implemented in a variety of ways for different federations and mission types. The DMT Master Conceptual Model (MCM) provides the description of the shared battlespace necessary for team training interactions.

The MCM is a set of products that together define a battlespace or virtual world at a level of abstraction suitable for defining training objectives, guiding implementation, and identifying limitations to inter-team training scope. It provides the conceptual framework for DMT battlespace entities and interactions, and guides the development of the standards and the DMT Portal software. The data model of the battlespace provides the common semantic content in the form of entities, events, interactions, and phenomena along with the parameters that describe them.

ABOUT THE AUTHORS

Dr. Tony Valle is the manager of the Training Simulation Division of SPARTA, Inc. He supports the DMT O&I contract as a member of the System Engineering and Integration Team (SEIT), and contributes to several standards, the Mission Packages, and R&D efforts including the MLS Guard and HLA Portal Prototype tasks. He has more than 20 years experience in both military and commercial simulations.

Bruce McGregor is the Program Manager for the DMT Operations and Integration Program for Northrop Grumman, and was formerly the Chief Architect for the project. He has served as Chief Engineer on numerous Northrop Grumman Simulation and Training efforts, and was the Principle Investigator for Simulation and Training Initiatives Project for Northrop Grumman.

The DMT Master Conceptual Model

Dr. Tony Valle
SPARTA, Inc.
Orlando, FL
Tony_Valle@trwdmt.com

Bruce McGregor
Northrop Grumman Mission Systems
Orlando, FL
Bruce.McGregor@ngc.com

CONCEPTUAL MODELS

The term "conceptual model" refers to the description of the content of a simulation, normally presented in a form that guides the implementation (but is abstract from the actual implementation) and educates end users about the simulation's limits and utility. Though every simulation has a conceptual model, they are rarely formally documented (if at all) and linger on only in the forms of Analyst's or User's Guides, software documentation, and the accumulated knowledge of the simulation developers and maintainers.

As the complexity and scope of distributed simulations has grown over the last decade, the lack of formally documented conceptual models has led to considerable difficulty in achieving simulation interoperability and attaining the promise of a seamless, integrated, virtual battlespace with rich content. This is especially true when disparate simulations are brought together for large-scale exercises, resulting in long planning cycles, substantial amounts of engineering re-work, and little or no "leave-behind" available for future events.

The requirement for regular training of multiple aircrews, or "inter-team" training, on the Distributed Mission Training (DMT) program leads to the imposition of more formal specification of the battlespace content in the form of Standards and associated engineering documentation. While the DMT Master Conceptual Model (MCM) is not a complete representation of the Distributed Mission Operations (DMO) battlespace, it does regulate critical interoperability areas that are essential to effective training. Over time, the scope and formality of the MCM will grow as additional platforms and associated training objectives become part of the DMO Network.

Typical Content

At the most abstract level, a conceptual model must describe the simulation content in four areas:

Entities: are those simulated objects that have persistent and time-varying state. These are the "objects" that populate a simulation and usually consist of vehicles, missiles, facilities, individual combatants, and all the other simulated physical things that are explicitly represented.

Environment: is the stage over which the actors (entities) play. The environment also has persistent state, but is really "background" that fills in the gaps between the "foreground" of entities and their interactions. The dividing line between entities and the environment is often blurry, and rather than offer a precise definition here, we will acknowledge that there are simulated objects that could be characterized as either in most implementations. Typically, the environment consists of the terrain, atmospheric phenomena, physical properties such as temperature, pressure, or illumination, and non-explicit features and culture such as forested areas.

Events and Interactions: are the discrete mechanisms that drive dynamic behavior in a simulation. Events include critical state transitions that effects multiple entities such as radar mode changes, weapon firings, or munition detonations. Interactions are events that involve an exchange between entities such as collisions, communications, or resupply activities.

Modeling Approaches: are the "rules of the game" for a simulation. They govern how simulated entities move and interact, how events are resolved, and how effects that arise from the interactions are portrayed and represented in the simulation. A complete conceptual model defines the basic mechanisms that drive the simulation implementation, but usually not completely specified, i.e. there is some (often considerable) latitude in the implementation details.

Another important objective of a conceptual model is to characterize the level of fidelity of the models to be implemented. As a first approximation, we find it useful to characterize models at one of four levels of fidelity: emulative, faithful, abstract, and simple.

Emulative models are those that are normally thought of as "engineering grade". They represent a real world system with sufficient accuracy that they can be used to conduct design tradeoffs, failure analyses, and stimulate hardware-in-the-loop simulations. The most important characteristic of emulative models is that they have a one-to-one correspondence between real world components of the modeled system, and simulated objects. That is, an emulative level model of an aircraft engine will have simulated representations of the inlet, compressor, combustor, turbine, nozzle, etc.

Faithful models are those that accurately represent the behavior of the system being modeled, without representing its constituent parts. Faithful models can be calibrated or fit to real data to improve their accuracy, and show the same trends to changes in inputs as the real world system. A faithful model of an aircraft engine could be done using a family of thrust curves as a function of altitude and Mach number, for example.

Abstract models represent the simulated system in some statistical or approximate sense. Because they don't necessarily share the same input and outputs as the system being simulated (if they did, they would be faithful), it may not be possible to calibrate them in the engineering sense. An abstract model of an aircraft engine might use a sea level static thrust and a lapse exponent that characterized thrust loss with altitude, but would show no variation with Mach number.

Simple models are most often found in simulations when a phenomenon must be present, but there's no known way of representing it numerically. Examples include the effects of fatigue, morale, or training level on human performance, or the value of implicitly modeled communications. A simple model of an aircraft engine might be binary: the aircraft flies as long as the engine is not damaged, and crashes when it is.

Representing Conceptual Models

One of the reasons that conceptual models are rarely formally documented is that representing them in a useful way can be difficult. The most common approach is simply to organize the content in the form of a traditional document with text and figures. While this is a flexible and familiar approach, it can result in documents that are too bulky to be easily digested by developers and users. Examples of partial conceptual models in document form include the RPR-FOM GRIM [GRIM03], the General Requirements section of the DIS standard [IEEE 1278.1-1995], and the Analyst's Guide for the Corps Battle Simulation [JPL92].

A second common means of representing conceptual model content is in the form of formal object-oriented artifacts such as UML diagrams. These have the advantage of completeness and hierarchy that make them easier for developers to use. But they are also often obscure to end users, and still more frequently they are too close to the implementation to serve as good abstract, i.e. implementation independent, representations of the battlespace. They are effectively software design documentation rather than conceptual models. A good example of a conceptual model in this form is the OneSAF Objective System Conceptual Model.

In fact, the best representation for a complex conceptual model is not known. The difficulty arises from the abstract, implementation-independent character of a good conceptual model, and the fact that very few good examples of formally documented ones exist.

Our experience suggests that a human-executable wargame is the best format for educating simulation engineers on the key content of a conceptual model. A commercial wargame is essentially a packaged "map exercise", with representations of terrain, features, and culture on a printed map, often overlayed with a hexagonal grid to regulate movement, pieces that represent entitles, and rules and charts that capture the events, interactions, and resolution mechanisms. In essence, such a wargame is a physical embodiment of a well-constructed conceptual model (at an abstract level appropriate for the purpose of the game).

It is, of course, not practical to build this kind of conceptual model representation for a complex, high-fidelity simulation because of the number of models, events, and interactions that have to be specified. Nevertheless, the thought processes that go into building a commercial wargame are exactly those that underlie a well-formed conceptual model, regardless of the means used to capture the decisions that result.

Importance and Lessons Learned

The conceptual model defines an artificial world: the one that will be simulated in the final implementation. Because the real world is infinitely complex, no simulation is able to represent reality with complete fidelity. So the conceptual model for a simulation serves not only to organize the content of the battlespace, but also to define (explicitly or implicitly) what *is not* simulated. This is the most important function of a formal conceptual model (i.e., capturing these decisions).

Another way of putting this is to say that the primary customers of a conceptual model are the end users and the developers who will write the simulation code.

These two very different audiences must both understand the content of the simulation in order to assess its utility and limitations (users) and to design and implement it (developers). That means the product must be in a format that is comfortable and useful to both customer types. In addition, conceptual models are an excellent resource in support of validation and testing, because they bound the scope of the battlespace and constrain the implementation details.

One thing is clear from past experience in the simulation industry: every simulation has a conceptual model, but very few have been written down. The modeling decisions must be made at some point in time, i.e. the choice of what *not* to simulate will be made by someone. Knowledgeable domain engineers can make it in advance in conjunction with subject matter experts, or software engineers with little or no domain understanding can make it at coding time. A formal conceptual modeling process allows greater programmatic control over these critical decisions.

LEGACY SIMULATION IMPACTS

Frequently we must deal with simulation systems that consistent of numerous legacy simulations combined with newly developed products, and DMT is in this category. Both the Boeing F-15C and Plexsys E-3C simulator systems were under development before the establishment of the O&I contract. And while the Lockheed-Martin F-16C simulators are new, older systems such as the EC-135 Rivet Joint, and the B-2 Spirit are scheduled to be added to the DMO Network in the next few years.

Legacy simulations are built upon some conceptual model, documented or not. When several are put into a common simulation environment, they interoperate correctly only in those battlespace areas where the conceptual models agree. Achieving interoperability requires first agreeing on a common conceptual model (which may involve changes to the legacy models), and then designing and implementing the revised battlespace in some common simulation protocol.

Effect of DIS and HLA/RPR-FOM

The DIS protocol arose out of a relatively compact distributed simulation problem: tactical, man-in-the-loop mounted ground combat. Over time, the protocol has been extended to cover a wide variety of simulation domains, but the early development served to produce a kind of implied conceptual model that became the basis for DIS compatible implementations to interoperate. Tying DIS simulations together still requires engineering effort, however, especially when they are operating outside the traditional ground combat domain. This is due, in large part, to the implicit nature of the DIS conceptual model, the fact that the standard has some ambiguities that allow divergent implementations, and the many parts of the standard that are open to extension or convention. A substantial history of the level of effort required for large-scale exercises can be found in organizations such as the Distributed Mission Operations Center (DMOC) at Kirtland AFB, or the Joint Warfighting Center (JWFC) in Suffolk, VA.

The RPR-FOM is the result of translating the basic functionality of DIS into the DMSO High Level Architecture (HLA). As such, the underlying conceptual model is very similar to that in DIS. The GRIM makes more of the conceptual model explicit than is the case for DIS, but there is still considerable flexibility and variability in the choice of battlespace content that complies with the HLA standard [IEEE 1516] and uses the RPR-FOM.

Unfortunately, the assumption is often made that a common simulation protocol (and associated data model) assures simulation interoperability. Of course, a common protocol and data model are *necessary* but by no means *sufficient*. Simulation interoperability requires a common conceptual model—at least in the areas where the simulations are expected to interact. Requiring HLA (or DIS) compliance will not guarantee interoperability, and may not in fact even substantially reduce the engineering effort required to achieve it if the conceptual models of the component simulations are too divergent.

Implementation Hiding

Once a simulation has been designed and implemented as running code, the content of the conceptual model is often lost unless it has been formally captured. The modeling choices made by the simulation architect may have been captured in the form of software requirements, or may have been given as guidance to the developers over the course of implementation, but they aren't made explicit in terms that are readily accessible by users of the simulation. The user often receives a "black box" and must infer the battlespace content from the observed behavior of the system. This can be very misleading, because complex emergent behaviors can appear to be very faithful to reality and further blur the essential distinction between the real world and the simulated battlespace.

On the DMT program, the standards development process serves to mitigate this effect. By bringing together the engineers responsible for the simulation implementation and asking them to participate in development of the shared conceptual model content, we gain a better, collective understanding of the properties and limitations of the shared battlespace.

Change Complexity

Another common problem that arises from incomplete or undocumented conceptual models is the difficulty in estimating the cost of changes to the battlespace, and the complexity of testing and debugging the revised implementations. The difficulty is that two distinct complex systems are being modified simultaneously: a conceptual model, which is a complete logical artificial world, and a large body of software that implements it. This makes testing more difficult because the source of continued interoperability issues could lie with the software implementation, the protocol content, or the fundamental assumptions about the battlespace that are implicit in the code.

The same issue shows up when new battlespace content is added to existing simulations. The complexity of the change is substantially increased if the solution is not defined in battlespace terms and agreed to by all parties in advance. The discussion and agreement is much easier to do in terms of the more abstract, implementation independent language of the conceptual model, than in terms of detailed software design artifacts.

DMT MCM COMPONENTS

The DMT program was not afforded the luxury of an early effort dedicated to developing a conceptual model for the entire DMO battlespace. Instead, the conceptual model is partial and focuses on those aspects of the simulated world that are both 1) critical to inter-team training and 2) not already relatively well-defined by the implicit conceptual model underlying DIS and the HLA/RPR-FOM. For example, there is no need to define the model for entity movement beyond the familiar linear, 1^{st}-order dead reckoning technique well-known from DIS simulations.

As a consequence, the DMT MCM is, like many other conceptual models, distributed across multiple documents, and specifies both abstract and implementation specific information that would be separated in an ideal world. The MCM is built on the Mission Package specifications, the DMT Standards documents, and a body of engineering analysis that is incorporated in the standards process in the form of point papers, engineering briefings, and technical notes.

Mission Package Definition

The DMO Network links together Mission Training Centers (MTCs) that provide training to aircrews of a specific platform. For example, the F-15C MTC contains four F-15C cockpits and supporting simulations, while the AWACS (E-3) MTC is able to train a complete AWACS mission crew. As one might expect, the local missions flown in support of the training objectives for F-15C two-ship or four-ship flights do not provide a broad enough scope for effective training of an AWACS Weapons Director.

The DMT Mission Package address this by laying out a scenario of sufficient scope to ensure a baseline capability for inter-team training of all the simulated platforms on the DMO Network. The Mission Packages are organized by calendar year, corresponding to the cycle in which new assets are added to the DMO Network. Mission Package MP04, for example, contains a battlespace appropriate to train F-15C, F-16CJ, and E-3 crews, while MP05 adds scenario content and scope to permit effective training of Rivet Joint and strike aircraft crews as well.

The Mission Packages contain conceptual model content in the form of entity lists and environment scope, and some interaction requirements. The entities define the minimum set of battlespace objects that all participants are required to represent at some level of abstraction, and they include friendly and threat forces, both airborne and land-based. The environment section lists minimum representation requirements such as the ability to model day and night operations, smoke, haze, clouds, and severe weather such as thunderstorms. The interaction section describes mandatory datalink and voice communications interactions, as well as sensor systems that must be supported. At the most abstract level, the Mission Packages capture the scope of the common DMT battlespace for the calendar year they are associated with.

DMT Standards

The central engineering product of the DMT O&I contract are the twelve standards that emerge from the standards development process. The standards process is collaborative, with full participation from the developers of the existing simulators, the O&I contractor team, the Air Force, and developers of simulators that are scheduled for addition to the DMO network in the future, but are not yet participating in operational exercises.

Several of the standards are purely process oriented, such as the Conformance Test standard, while others are purely associated with implementation details such as the Network standard. Five standards, however, capture a significant portion of the current DMT MCM content: Common Models, Synthetic Natural Environment (SNE), Threat Representations and Computer Generated Forces (TR&CGF), and the two data standards DMT Tailored DIS and the Reference FOM.

Common Models. The Common Models standard prescribes specific algorithms and modeling approaches when a single approach is required to support inter-team training objectives. For example, the Common Model standard specifies the algorithm used for damage assessment and requires that DMO simulations populate the content of the interaction in a well-defined way, depending on the class of the munition. This standard also specifies the choice of dead reckoning algorithms, the geode used for the representation of the earth's surface, and coordinate systems that ensure all participants can meaningfully operate in a shared battlespace.

Synthetic Natural Environment. The SNE standard defines the geographic extents that define the DMO "playbox" in several theaters. With the advent of MP05, this standard will grow to include battlespace-level definitions of atmospheric effects, common representations of essential visible phenomena like smoke and haze, and diurnal effects. The SNE standard is therefore the home of the Environment portion of the DMT MCM, and though an early focus on beyond-visual-range fighter combat downplayed the significance of the standard, it will have to mature rapidly to support missions such as close air support, time critical targeting, and search and rescue that will be rely on the DMO Network soon.

Threat Representation and Computer Generated Forces. Technical and operational considerations demand that a DMO simulation have the flexibility to relocate computer generated forces assets from one MTC to another during the course of a training event. The TR&CGF standard governs the behavior of CGF systems and ensures that they share enough of a common battlespace to permit the handoff of threat entities. It also sets bounds on the scope of the battlespace that must be supported by CGF systems, and the level of representation of threat entities regardless of the fidelity of the underlying models.

DMT Tailored DIS and Reference FOM. The two data standards define the interaction protocol data content and therefore provide additional detail about the parameterization of the battlespace information in the MCM. Because every battlespace interaction required for effective inter-team training must pass between MTCs at some point, the data standards are closely associated with the Events and Interaction content of the MCM.

Other Engineering Products

In addition to the Mission Packages and DMT Standards, other engineering products contain information vital to the conceptual content of the DMO Network. We often define new modeling approaches or battlespace phenomena in the form of an engineering brief to the Standards Development Working Group (SDWG). These briefings are then elaborated in the form of point papers that undergo formal review and assessment by the SDWG members (which includes developers from all the participating DMO MTCs). Point papers have been produced on several conceptual modeling topic areas including Damage Assessment, Identification Friend or Foe (IFF), and several other areas, some of which are discussed in the next section.

EXAMPLE CASES

There have been a number of cases of both failures and successes on DMT related to conceptual model content. It is useful to consider some of these to see where conceptual modeling helps achieve long-term, stable simulation interoperability, and to see where failure to document the conceptual content of a simulation makes interoperability considerably more difficult.

IFF Implementation

During the engineering testing prior to initial operational capability, it became clear that the F-15C and E-3 simulators experienced degraded training because of limitations in the interoperability of their IFF systems. Both simulation systems were using the DIS protocol, and both were using a valid (in the sense of the DIS standard) IFF implementation. There were no issues related to the network implementation (byte swapping, PDU formatting, etc.), but the E-3 aircrew were unable to properly interrogate friendly F-15s as required for them to train effectively. It was eventually discovered that the IFF approaches used by the two simulators were fundamentally different at the level of the conceptual model, and therefore no amount of testing and debugging at the implementation level would resolve the problem.

The E-3 simulator used a faithful representation of IFF, one that sent out interrogation interactions with each simulated sweep of the APY-2 radar beam, and expected response interactions in return. The F-15 simulator, on the other hand, used an abstract representation of IFF, one that built a local database of valid IFF codes when they were received. No interrogation interactions were ever generated as a result of a simulated radar sweep, instead the simulator produced the proper response by consulting the local database. Similarly, the simulator did not respond to interrogations with an interaction.

As a result, F-15 pilots were able to interrogate the E-3 and receive correct IFF replies, but the E-3 was unable to get proper responses from the F-15s, which therefore showed up as “unknowns” on the E-3 radar displays. A problem like this one is very difficult to debug at the protocol level since it’s difficult to know why the F-15 responses are not making it to the E-3 simulator, and much fruitless speculation about the networks, simulator implementations, and data packet contents can be investigated to no avail. Without a common conceptual model for the interaction, no implementation details are relevant.

Resolving the issue requires a common battlespace approach and once that has been agreed to, a change in one or both simulator implementations.

Simulated Radios

The DMO Network uses the well-known DIS simulated radios to carry aircrew voice between the MTCs. These radios are highly parameterizable, with settings for the major and minor system types, digital encoding scheme and bit rate, and modulation characteristics including frequency, method, and analog and digital sampling. The radio models are designed to emulate every important characteristic of their real world counterparts, and to do so with knowledge of the simulated location of their associated entity to permit very high fidelity communications models in distributed exercises. One might expect, therefore, that radio interoperability on the DMO Network would be very straightforward. Once again, however, the difference in the conceptual approach used by various simulators interferes.

Some installations don’t wish to bother with realistic communications models and have taken the expedient measure of locating the simulated radios at the center of the earth (x,y,z = 0,0,0) so that range and terrain effects can be ignored. When interacting with other radios that are using terrain effects, the simulated 6,000 Km of dirt between the radios precludes communication.

In other cases, a simulated radio that is supposed to represent a particular piece of equipment (such as a Have Quick radio) is parameterized as a different piece of equipment, or using modulation techniques that are not used by the real world radio. In these cases, coordinating the event by describing the real radios in use will fail.

As before, the problem is that some sites are using faithful (but distinct) radio models, while others are building abstract models. Without a common conceptual model of simulated radios, no amount of pre-coordination of parameters such as frequency or power levels will produce interoperability.

Tracked Munitions

An example of a success built on conceptual modeling principles is the definitization of tracked munitions modeling for DMO. Munitions are “tracked’ if they are explicitly represented as an entity in the battlespace with a (dead-reckoned) location and velocity. Nothing in the DIS standard defines which weapons should be tracked and which should be treated implicitly. In other words, the DIS standard does not define which munitions should be faithfully represented, and which should be abstractly represented.

By working through the standards development process, the issue has been resolved in every case but one on DMT. Guided and precision weapons are tracked, while others are not. The need to track AAA to provide consistent visual feedback to pilots is still being debated and will likely be subjected to test in the near future. This resolution of the conceptual model issues for munitions satisfies the inter-team training objectives and will ease the testing burden and reduce debugging effort.

Emissions and Secondary Modeling

For some time, we have recognized the fact that the simulated electromagnetic emissions environment necessary to support an intelligence platform such as Rivet Joint is far broader in scope and richer in detail than one needed to properly stimulate F-15 or E-3 simulators. Furthermore, such a large scope battlespace cannot even be represented by the tactical platforms, and requiring them to do so would impose too harsh a burden on their implementation.

As a result, no “one size fits all” solution for the battlespace is appropriate for DMO, and an alternative conceptual approach is required. The solution currently being investigated is an explicit partitioning of the emissions battlespace that permits lower fidelity models of systems to flow to the tactical assets, and simultaneously to stimulate higher fidelity models internal to the intelligence simulators. The technique is called "external secondary modeling" because it uses data provided by a low fidelity simulation at another location (external) to be elaborated (secondary) by a local simulation (modeling). Success will require careful coordination of the battlespace content, and is the subject of a DMT Tiger Team (Emissions) for the conceptual model, and on-going experimentation for the implementation.

SUMMARY

Conceptual modeling is a critical activity in simulation development, often poorly documented or relegated to a part of the software development process. This is unfortunate because conceptual modeling mismatches can be difficult to identify at the implementation level and will prevent simulation interoperability.

The DMT Master Conceptual Model (MCM) is captured in a variety of artifacts associated with the DMT standards and other engineering activities on the program. It serves to define the critical shared battlespace content required for effective inter-team training, as well as to identify areas requiring additional engineering attention.

REFERENCES

[GRIM03] Rybka, M., Fischer, J., Bertin, R. (2003). Guidance, Rationale, and Interoperability Manual for the Real-time Platform Reference Federation Object Model (RPR-FOM), v 2.0D17v3

[JPL92] Jet Propulsion Laboratory. (1992). CBS 1.4 Analyst's Guide Volume 3, Air. JPL Internal Document D- 7849, Rev. A. Pasadena, California.

Distributed Mission Training Operations & Integration website, http://www.trwdmt.com/

IEEE 1278.1A - IEEE Standard for Distributed Interactive Simulation.

IEEE Standard P1516 Framework and Rules.

IEEE Standard P1516.1 Federate Interface Specification

IEEE Standard P1516.2 Object Model Template (OMT) Specification

Real-time Platform Reference Federation Object Model (RPR FOM). 1998.