Simulation environment architecture development using the DoDAF

Tom van den Berg
Netherlands Organisation for Applied Scientific Research (TNO)
Oude Waalsdorperweg 63
2597 AK The Hague
The Netherlands
tom.vandenberg@tno.nl

Robert Lutz
Johns Hopkins University Applied Physics Laboratory (JHU/APL)
11100 Johns Hopkins Road
Laurel, MD 20723-6099
USA
robert.lutz@jhuapl.edu

Keywords: DoDAF, DSEEP, UML, SysML, SoaML, UPDM, architecture development

ABSTRACT: *The US Department of Defense (DoD) Architecture Framework (DoDAF) provides a common approach for architecture description development. The primary use of DoDAF is capability development and system acquisition in the military domain. Although DoDAF was not designed to support the development of simulation environments, many of the artifacts necessary to produce and document such simulation environments can be described by the architecture models in this framework.*

This paper examines the application of the DoDAF and the related systems engineering concepts within IEEE 1730-2010 Distributed Simulation Engineering and Execution Process (DSEEP), a recommended practice for the engineering and execution of a distributed simulation environment. This paper uses the Unified Profile for DoDAF and MODAF (UPDM) for the construction of architecture models. UPDM is a UML profile of the Object Management Group (OMG) that provides a modeling standard for both DoDAF and MODAF.

This paper builds on an earlier SISO paper on architecture description development (04F-SIW-015: The Application of the DoDAF within the HLA Federation Development Process), which was based on DoDAF 1.5. This paper adds amongst others new DoDAF 2.0 viewpoints, provides additional examples of models, uses UPDM for model construction, and references the DSEEP as engineering process.

1 Introduction

The process of integrating a collection of dissimilar simulation components into a unified, interoperable simulation environment introduces an array of significant engineering challenges. Many of these challenges are technical concerns, but the need to ensure coordinated and timely interaction among the organizations that participate in such developments raises a variety of challenges from the project management perspective as well. Recognizing and mitigating these issues are critical to controlling risk across a simulation development effort.

In 2010, a standard was published to provide better user insight into the process of engineering a distributed simulation system. The IEEE 1730 Distributed Simulation Engineering and Execution Process (DSEEP) was designed as a high-level process framework that could effectively address the needs of a very large and diverse user community. This framework allows DSEEP users to tailor the details of process implementation to their specific application requirements. However, since the DSEEP tends to focus much more on "what" needs to be done rather than "how" to do it, many different approaches have arisen in recent years with respect to tools and methods for implementing DSEEP steps and activities.

This paper describes how a DoD/MoD architecture modeling standard can be used to execute certain DSEEP activities. While this standard was not developed for simulation systems, the architectural constructs described by this standard show great promise in terms of applicability to the simulation domain. By reusing these constructs, users may leverage a very broad and deep knowledge base of systems engineering experience to facilitate more capable and robust simulation environments in the future.

2 DoDAF overview

The Department of Defense Architecture Framework (DoDAF) [1] provides a framework for developing and representing system architecture descriptions that ensure a common denominator for understanding, comparing, and integrating architectures across organizational boundaries.

The Framework defines a number of related architecture viewpoints. Each architecture viewpoint defines several model kinds to represent aspects of the system. The name "model kind" refers to the conventions for a type of modeling, such as Unified Modeling Language (UML) activity diagrams for behavioral modeling. An *architecture view* expresses the architecture of a system from the perspective of an architecture viewpoint. An architecture view is composed of one or more architecture models that are constructed according to the conventions specified by the model kind governing each model. Some of the elements that are used in the model kinds bridge two viewpoints and provide integrity, coherence, and consistency to the integrated architecture definitions of the viewpoints.

The Operational Viewpoint, for example, defines model kinds for the description of operational activities and performers, workflow, information flow, and event traces for operational scenarios. The different model kinds in DoDAF are named as AV-1, OV-3, OV-5, etc. The numbers do not really have a meaning, other than identifying the model kind.

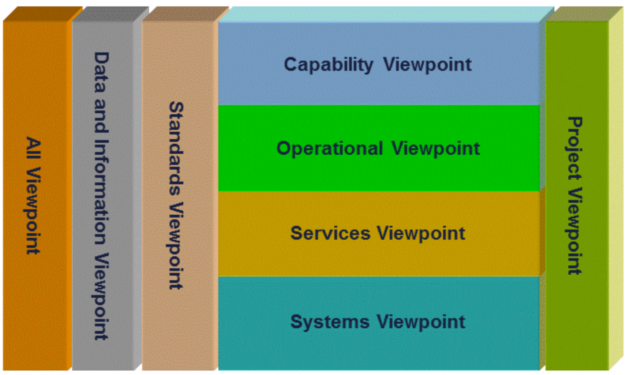

Figure 1: DoDAF viewpoints.

DoDAF version 2.0 defines the following viewpoints, as is depicted in Figure 1:

- The Capability Viewpoint (CV) articulates the capability requirements, delivery timing, and deployed capability.
- The Operational Viewpoint (OV) articulates operational scenarios, processes, activities and requirements.
- The Services Viewpoint (SvcV) articulates the performers, activities, services, and their exchanges providing for, or supporting, DoD functions.
- The Systems Viewpoint (SV) articulates the systems or independent systems, their composition, interconnectivity, and context providing for, or supporting, DoD functions.
- The Project Viewpoint (PV) describes the relationships between operational and capability requirements and the various projects being implemented.
- The Standards Viewpoint (StdV) articulates applicable operational, business, technical, and industry policy, standards, guidance, constraints, and forecasts.
- The Data and Information Viewpoint (DIV) articulates the data relationships and alignment structures in the architecture content.
- The All Viewpoint (AV) describes the overarching aspects of the architecture context that relate to all models.

3 DSEEP overview

As distributed simulations become more complex, and tend to be systems in their own right, a structured systems engineering approach is needed to develop them. Although traditional software development processes may be applied to the development of distributed simulation environments, these processes lack simulation specific steps and activities that are important for distributed simulation environments. For example, the development of a simulation conceptual model and simulation scenario, and the development of a simulation data exchange model with associated operating agreements between member applications. The only recognized industry standard process for distributed simulation environment development is described in [2], called Distributed Simulation Engineering and Execution Process (DSEEP). This process is independent of a particular simulation environment architecture (e.g. HLA) and provides a consistent approach for objectives definition, conceptual analysis, design and development, integration and test, simulation execution, and finally data analysis.

The DSEEP was originally developed under the umbrella of the the Simulation Interoperability Standards Organization (SISO) by a large community of (distributed) simulation practitioners, and became an IEEE standard in 2010. A top-level illustration of this process is provided in the figure below. The DSEEP identifies a sequence of seven basic steps with activities to design, develop, integrate, and test a distributed simulation environment of disparate simulation models. Each activity in the DSEEP is further broken down in tasks and work products. The guidance provided by the DSEEP is generally applicable to standalone simulations as well.

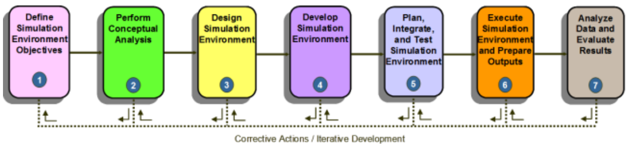

Figure 2: DSEEP steps.

First, a brief summary of each step of the DSEEP is provided below. For more information the reader is referred to the standard itself.

The seven steps are in summary:

- Step 1: Define simulation environment objectives. Define and document a set of user/sponsor needs that are to be addressed and transform these needs into a more detailed list of specific objectives for that environment.
- Step 2: Perform conceptual analysis. Develop an appropriate representation of the real-world domain that applies to the defined problem space and develop the appropriate scenario. Transform the objectives for the simulation environment to simulation environment requirements.
- Step 3: Design simulation environment. Produce the design of the simulation environment. This involves identifying applications that will assume some defined role in the simulation environment (member applications) that are suitable for reuse, creating new member applications if required, allocating the required functionality to the member application representatives.
- Step 4: Develop simulation environment. Define the information that will be exchanged at runtime during the execution of the simulation environment, establish interface agreements, modify member applications if necessary, and prepare the simulation environment for integration and test.
- Step 5: Integrate and test simulation environment. Integration activities are performed, and (formal) testing is conducted to verify that interoperability requirements are being met.
- Step 6: Execute simulation. The simulation is executed and the output data from the execution is captured and pre-processed.
- Step 7: Analyze data and evaluate results. The output data from the execution is analyzed and evaluated, and results are reported back to the user/sponsor.

4 Unified Profile for DoDAF and MODAF (UPDM)

The Unified Modeling Language (UML) is a general purpose modeling language to describe the design of a system. It is currently managed as a standard by the Object Management Group (OMG) [3]. UML provides a generic mechanism to extend the language for building models in particular domains. This extension mechanism is called a UML Profile. A UML Profile is defined using stereotypes and tagged values that are applied to UML elements, attributes, methods, links, link ends and more. A UML Profile is a collection of such extensions that together describe some particular modeling problem and facilitate modeling constructs in that domain.

The Unified Profile for DoDAF and MODAF (UPDM) [4] is a UML Profile that provides a modeling standard for DoDAF and MODAF (UK Ministry of Defence Architecture Framework). The profile incorporates SysML (Systems Modeling Language) [5] and parts of SoaML (Service oriented architecture Modeling Language) [6], enabling the modeler to leverage the modeling elements that are defined in these two profiles. SoaML is a modeling language for the specification and design of services within a service-oriented architecture. SysML is a general-purpose modeling language for systems engineering applications. UPDM version 2.1 can be used to construct DoDAF version 2.0 models.

Referring to [4], UPDM version 2.1 will support the capability to:

- model architectures for a broad range of complex systems, which may include hardware, software, data, personnel, and facility elements;
- model consistent architectures for system-of-systems down to lower levels of design and implementation;
- model service oriented architectures;
- support the analysis, specification, design, and verification of complex systems; and
- improve the ability to exchange architecture information among related tools that are UML based and tools that are based on other standards.

UPDM enables the engineer to describe the System of Interest (SOI) and the simulation environment representing the SOI with a common modeling language, using the DoDAF as architecture framework.

Figure 3 from [4] illustrates how the various profiles relate. The UPDM profile consists of two levels. UPDM Level 0 extends UML and imports several SoaML stereotypes from the SoaML profile. UPDM Level 1 includes everything in Level 0 and imports the complete SysML profile.

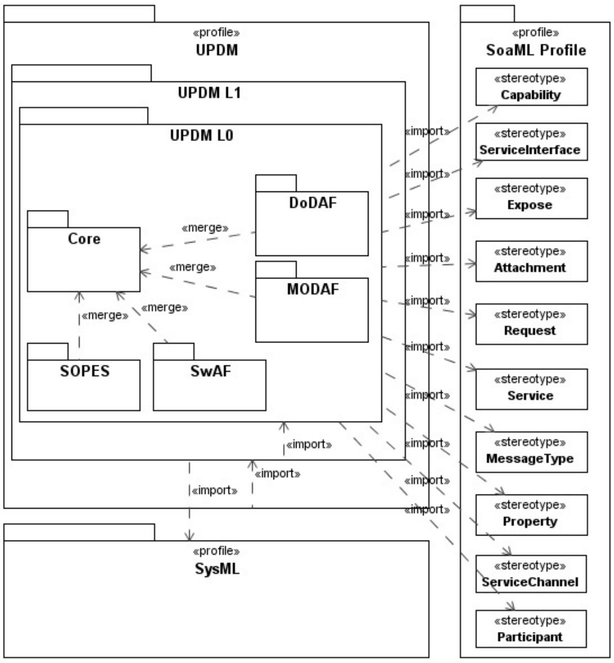

Figure 3: UPDM profile.

5 Architecture overlay on the DSEEP

5.1 Overview

Although the DSEEP identifies numerous products that are created as a result of development activities, the structure, format, and content of many of these products are largely left to the discretion of the simulation environment development team. While different user domains may desire at least some flexibility in how their products are defined, it does introduce barriers for cross-domain collaboration. Due to the increased importance and visibility of the DoDAF as a standard mechanism for system architecture development, and because distributed M&S environments are certainly systems, it makes sense to explore the potential utility of DoDAF constructs as a framework for producing selected simulation environment products. Such usage can introduce the common language necessary for Integrated Product Teams within a system acquisition program to work together more effectively and to achieve common goals.

The following materials provide recommendations as to how the DoDAF can be successfully applied in:

- DSEEP step 1 - Define Simulation Environment Objectives
- DSEEP step 2 - Perform Conceptual Analysis
- DSEEP step 3 - Design Simulation Environment
- DSEEP step 4 - Develop Simulation Environment

Figure 4 summarizes the viewpoints most applicable to each step of the DSEEP. As can be seen from the figure some of the viewpoints span multiple steps in the DSEEP.

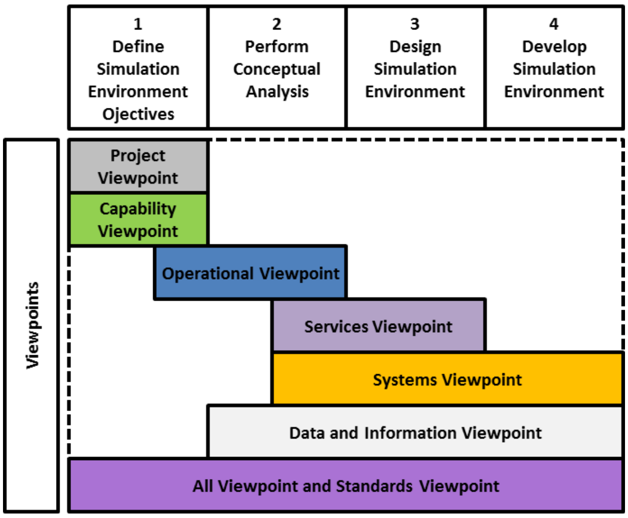

Figure 4: DoDAF viewpoints per DSEEP step.

Although the application of the guidance given here can certainly facilitate communication among acquisition participants, users should not feel constrained by the DoDAF constructs identified here. Rather, users should experiment on their own to look for additional ways in which the DoDAF can provide standard structures for simulation environment products.

5.2 Case study

This section uses a notional case study to provide examples of DoDAF models. The case study is called "Situational Awareness in a maritime task force", where units of a maritime task force exchange information to create a common "air picture" for real world objects in a real world open sea or littoral environment. Real world objects are, for example, missiles and aircraft. Each unit has sensors to track real world objects, and a tactical data link to exchange and manage tracks of real world objects with the other units in the task force. The air picture consists of tracks, with amongst others position, classification and identification information.

The main concept is shown in Figure 5. This figure is a DoDAF OV-1 model (High Level Operational Concept Graphic) that is typically used to show the main operational concept and interesting or unique aspects of the operation. This figure shows a task force consisting of four units, the sensor range of each unit (depicted as a circle), a tactical data link that units use for information exchange, and a number of aircraft. All situated in some geographical maritime environment.

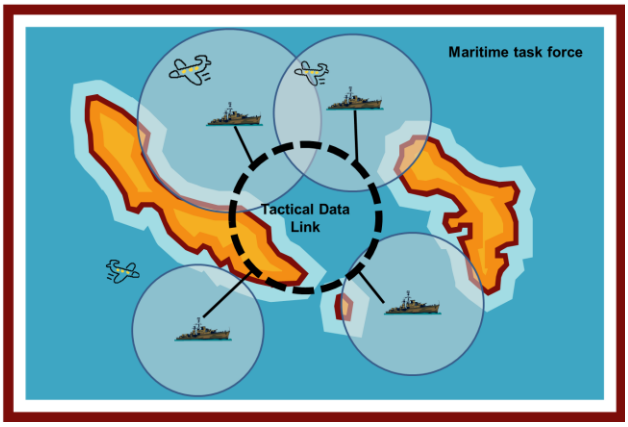

Figure 5: OV-1 (High Level Operational Concept Graphic).

The objective of the simulation environment is to measure and evaluate the different options in creating a common air picture between units. The simulation environment models the real world environment and real world objects, the units and their sensors, the relevant command and control processes, and the tactical data link. The simulation is a non-real time constructive Monte Carlo simulation, with stochastic variations in sensor parameters.

The DoDAF models that are provided as an example are constructed with UPDM version 2.1, using Sparx Enterprise Architect SE edition, version 11.0 [7].

5.3 DSEEP step 1 - Define Simulation Environment Objectives

The purpose of DSEEP step 1 "Define Simulation Environment Objectives" is to define and document a set of needs that are to be addressed through the development and execution of a simulation environment, and to transform these needs into a more detailed list of specific simulation environment objectives. The needs statement may vary widely in terms of scope and degree of formalization, but should include high-level descriptions of critical systems of interest, fidelity requirements, required entity behaviors, output data requirements, and key events that must be represented in the simulation environment scenario. The subsequent translation of high-level user/sponsor expectations into more concrete, measurable simulation environment objectives includes early assessments of simulation environment feasibility and risk. This assessment is based on such practical constraints as cost, schedule, availability of personnel and facilities, and limitations on the state-of-the-art of needed technology.

The M&S team is not the only group that is concerned with critical systems of interest, entity behaviors, etc. These are program-wide concerns for any real-world system and a driver behind the DoD's decision to mandate DoDAF architectures for new acquisition programs. The desire is to put the systems engineering rigor in at the beginning, saving time, money, and effort later.

The benefits in taking the time up front to develop a robust system architecture are apparent in these initial tasks. The physical system and the environment in which it operates are captured by the architecture represented using the DoDAF. The physical system includes the system of interest (SOI) and its associated subsystems. The environment includes external nodes the SOI must interact with, the command and control structure it operates within, and the natural environment.

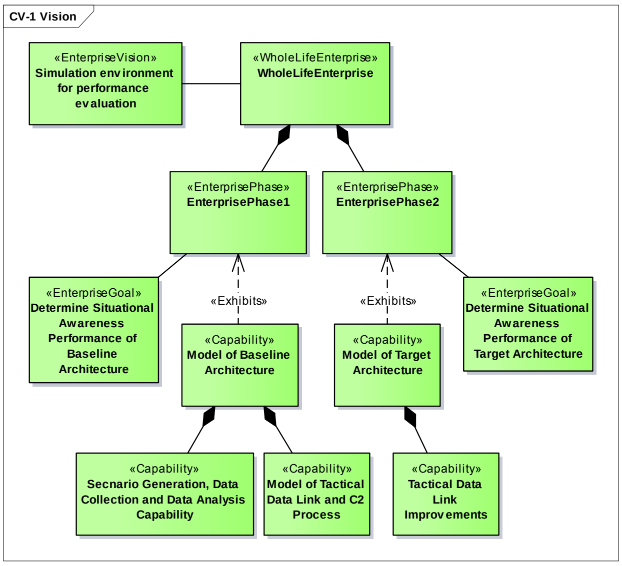

Figure 6: CV-1 (Vision).

The architectural model will be at a level of detail consistent with the requirements of the program. This makes it well suited to defining the simulation environment objectives. The simulation environment should be capable of modeling the real-world systems represented in the architecture to the required level of detail. It should be capable of modeling the entities, activities, behaviors, interactions, environment, etc., that are modeled in the DoDAF architectural model.

The objectives of the simulation environment can be captured in a CV-1 (Vision) model. This model outlines the vision for the simulation environment over a specified period of time and describes how high level goals and strategy are to be delivered in terms of capability. Figure 6 provides an example of a CV-1 model for a simulation environment where the capabilities are provided in two phases.
Several OV-1 (Operational Context) Fit for Purpose models can be used in this step to further describe the objectives listed in the CV-1 model and to provide high level requirements to the M&S team. Fit for Purpose models are user-defined models that are not listed in the DoDAF and that are created for some specific purpose. The OV-1d is a Fit for Purpose model and is titled "Operational Context Use Case". It provides a good starting point and is useful for showing the real-world activities the SOI must participate in. Figure 7 provides an example of an OV-1d model, depicting the high-level use cases for Situation Awareness and the actors that are involved. This model uses the UML use case diagram to represent the model. The letter 'd' in the model name is taken from the UPDM Fit for Purpose model numbering in [4].

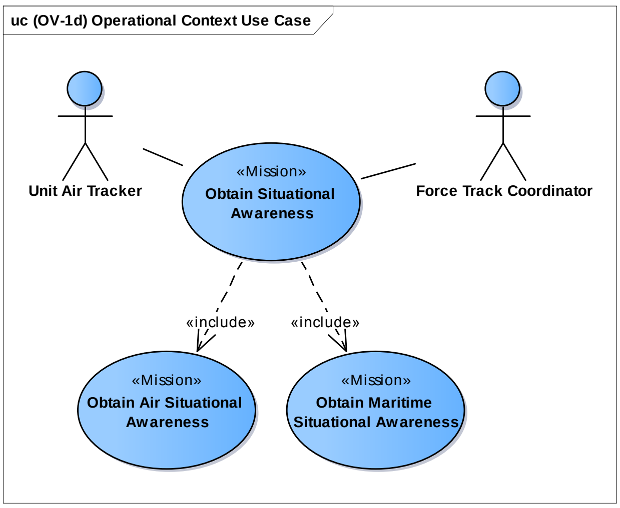

Figure 7: OV-1d (Operational Context Use Case).

These use cases can be considered high-level requirements for the simulation environment and may be used to generate an operational requirements document. Normally, the simulation environment should have the capability to model these use cases for requirements verification and analysis. Each of the high-level use cases can be broken down into a set of lower-level, higher fidelity use cases. In DSEEP step 2 "Perform Conceptual Analysis" the use cases are further elaborated in OV-5 models.

Other DoDAF models can be used to derive specific data requirements, measures of effectiveness (MOEs), measures of performance (MOPs), etc. These are all important factors in defining the simulation environment objectives, and they are represented in a common standardized framework that is understood by the program managers, the system designers, and the modelers. The Fit for Purpose OV-1c model titled "Operational Context Measurements" can be used to show the type of measurements (MOEs and MOPs) that the simulation environment must be able to provide. Figure 8 provides a number of MOPs for Situational Awareness.

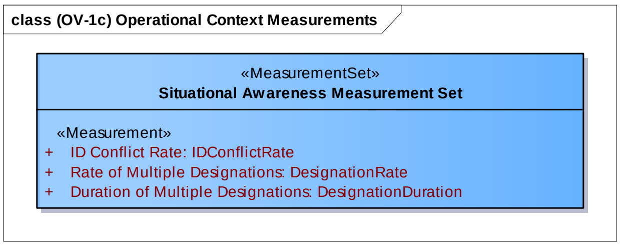

Figure 8: OV-1c (Operational Context Measurements).

The result of the tasks performed should be a needs statement, a simulation environment objectives statement, and an initial planning document. These can be detailed in a simulation environment AV-1 model, which provides general information on the architecture and the system being represented. The AV-1 is titled, "Overview and Summary Information," and is used in the initial phases of architecture development as a planning guide. In this case, an M&S-specific AV-1 model is produced to detail the simulation environment plans and objectives and to delineate them from the SOI architecture objectives. The DoDAF products mentioned provide an ideal starting point for defining the simulation environment objectives but are not an end-all solution. The degree to which they can be used is determined by the completeness of the products that have been developed for the SOI. In addition, there will be other considerations that are not well represented in the DoDAF architecture. An example might be M&S constraints such as funding, timelines, personnel, expertise, model availability, etc. The designer may wish to include these in an AV-1 model, a PV-1 (Project Portfolio Relationships) and PV-2 (Project

Timelines) model, or may opt to represent those aspects outside the scope of the DoDAF views. The M&S team should not feel constrained by the DoDAF products in defining the M&S objectives, but should use and tailor them where appropriate.

5.4 DSEEP step 2 - Perform Conceptual Analysis

The purpose of DSEEP step 2 "Perform Conceptual Analysis" is to develop an appropriate representation of the real-world domain that applies to the simulation environment problem space (conceptual model) and to develop the simulation environment scenario. The conceptual model provides an implementation-independent representation that serves as a vehicle for transforming simulation environment objectives into functional and behavioral descriptions for system and software designers. A set of highly specific simulation environment requirements is also produced during conceptual analysis that will be used in simulation environment design, development, testing, execution, and evaluation.

The conceptual model provides an implementation-independent representation of the systems and processes that the simulation environment must model. In the early stages of the conceptual model development, the SOI architecture described in DSEEP step 1 "Define Simulation Environment Objectives" is identical to the simulation environment conceptual model. In this step the operational activities are identified that must be modeled in the simulation environment. These are captured in an OV-5a (Operational Activity Decomposition Tree) and OV-5b (Operational Activity Model) model.

An example of an OV-5a model is shown in Figure 9. This model uses a UML activity diagram to describe the activity decomposition for Situational Awareness. Each of the sub activities may be further decomposed in additional activity diagrams.

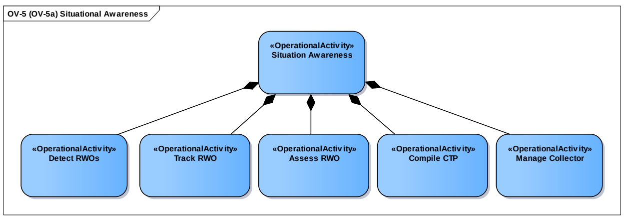

Figure 9: OV-5a (Operational Activity Decomposition Tree).

The activities in the OV-5a model are used in the OV-5b model shown in Figure 10 to describe the activity workflow, also using a UML activity diagram. Sub activities of Situational Awareness appear as actions in the OV-5b model. Activity swim lanes are used to allocate actions to operational nodes. In this example the actions Manage Collector, Detect RWOs and Compile CTP are performed by the Unit Air Tracker. The activity Track RWO and Assess RWO are further decomposed and described in additional activity diagrams.

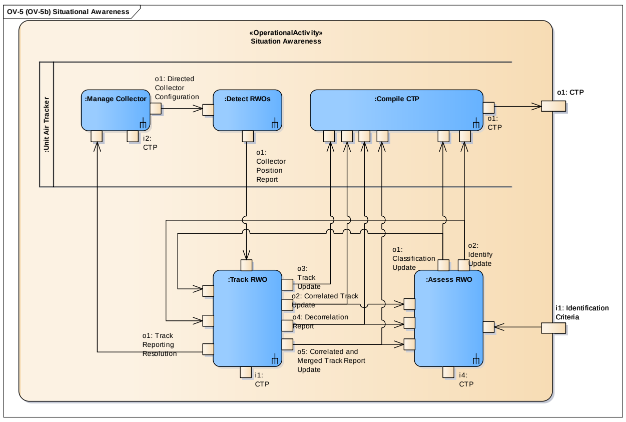

Figure 10: OV-5b (Operational Activity Model).

One artifact derived from development of the OV-5 models is a set of information exchanges among operational activities. The information exchanges are described in a DIV-1 (Conceptual Data Model) model, and are referenced from the other OV models. These information exchanges, when associated with the operational nodes that perform the activities, become elements of the OV-3 (Information Exchange Matrix) model. An example of an OV-3 model is provided in Figure 11.

Conveyed_Name	Producer_Type	Producer_Na...	Consumer_Type	Consumer_Name
Resolved Identify Conflict	Node	Force Track Co...	Node	Unit Air Tracker
Resolved Classification Conflict	Node	Force Track Co...	Node	Unit Air Tracker
Track Correlation Resolution	Node	Force Track Co...	Node	Unit Air Tracker
Track Reporting Resolution	Node	Force Track Co...	Node	Unit Air Tracker
Correlated and Merged Track Report Update	Node	Unit Air Tracker	Node	Force Track Coordinator
Correlated Track Update	Node	Unit Air Tracker	Node	Force Track Coordinator
CTP	Node	Unit Air Tracker	Node	Force Track Coordinator
Classification Update	Node	Unit Air Tracker	Node	Force Track Coordinator
Identify Update	Node	Unit Air Tracker	Node	Force Track Coordinator
Decorrelation Report	Node	Unit Air Tracker	Node	Force Track Coordinator

Figure 11: OV-3 (Information Exchange Matrix).

These information exchanges are essential elements of the OV-6c (Operational Event-Trace Diagram) model discussed later. For example, the OV-5b model for Assess RWO indicates there will be an information exchange (Classification Update) between the Unit Air Tracker and the Force Track Coordinator associated with the Classify RWO activity. This should correspond to needlines in the OV-2 (Operational Node Connectivity Description) model and information exchange requirement(s) in the OV-3 model. In this way, the OV-5 models can lead to other models that define the communications architecture to the required level of detail. An example of an OV-2 model without node ports is provided in Figure 12.

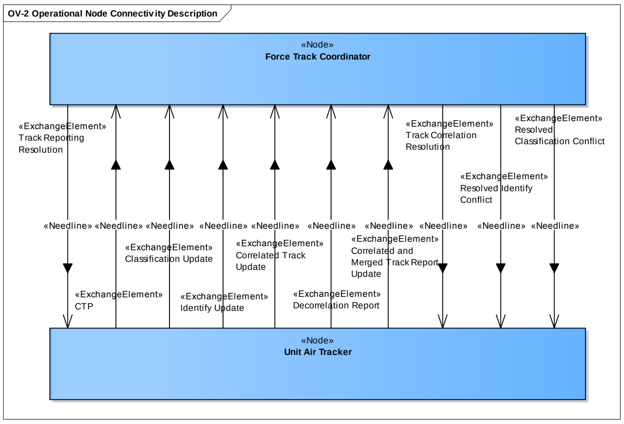

Figure 12: OV-2 (Operational Node Connectivity Description).

Utilizing the various DoDAF views, the conceptual model depicts the operational nodes, behaviors, and activities that must be modeled. This is one of the real benefits of using the same DoDAF products for the M&S effort as those being used by the program management and systems engineering teams. In some cases, the simulation environment requirements will be a subset of the real system requirements. For example, a real Unit Air Tracker will have to communicate with the Unit Anti-Air Warfare Officer, but that interaction may not have to be modeled. The degree to which the existing architecture corresponds to the simulation environment conceptual model is dependent on the completeness of the SOI architecture as well. Since the DoD is mandating that DoDAF architectures be developed for new acquisition programs, this should be a given. In any case, the conceptual model can be documented nicely using the same DoDAF views used to describe the actual system, and in most cases it will be nearly identical, requiring little additional work on the part of the M&S team. As the simulation environment development process proceeds and the simulation environment conceptual model evolves, it will be transformed from a general representation of the real-world domain to a more specific articulation of the capabilities of the simulation environment as constrained by the member applications and available resources. This will become clear as the follow-on steps are discussed.

In addition to developing the conceptual model, the tasks include the development of a simulation environment scenario that will exercise the activities, include the important operational nodes, and provide an analysis framework for the requirements. The scenario is what puts boundaries on the aspects of the conceptual model that is relevant to the study. That is, the views generated up to this point in the process may to a certain extent be similar with regard to the simulation environment and the system itself. However, not all aspects of the conceptual model necessarily need to be exercised to produce the metrics needed to satisfy the requirements of the case study. The scenario defines the storyline of events that have to be represented in the simulation, and thus defines the parts of the conceptual model that are relevant to the immediate case study and which are not. This can be accomplished in a number of ways. One approach is to examine the activities from the OV-5 models. At a high level the activities from the OV-5 models are used to detail the types of activities that will have to be included in the scenario. The OV-6c (Operation Event-Trace Description) model helps define node interactions and operational threads. These are essentially UML sequence diagrams that show the tracing of actions in a scenario or critical sequence of events. An example of an Assess RWO OV-6c model is shown in Figure 13.

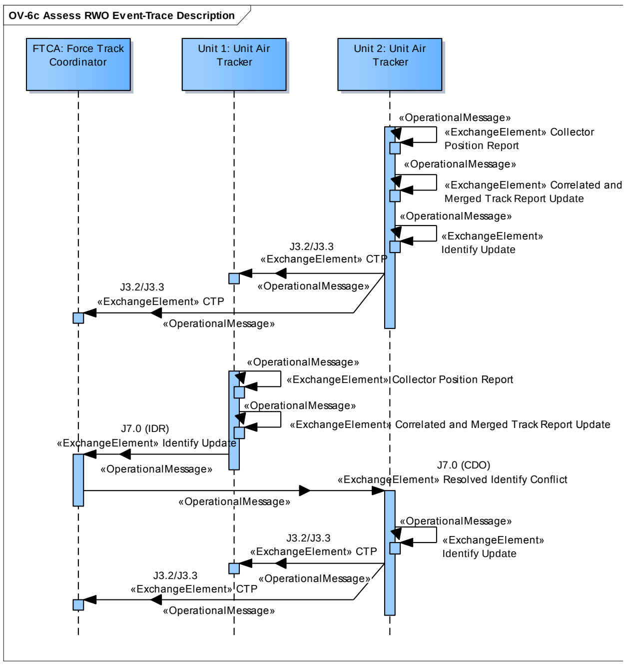

Figure 13: OV-6c (Operation Event-Trace Description).

The functional scenario can then be detailed in the form preferred by the simulation environment developer. This may be textual, graphical, tabular, or a combination of different forms. It will include a combination of several sequences which capture the essential activities and operational nodes. The scenario can be based on an existing standardized scenario or designed from scratch based on the requirements.

The final task is to produce a detailed set of simulation environment requirements. This is accomplished by examining the conceptual model and then detailing specific requirements. The requirements can be documented in a SV-5 (SV-5b, Operational Activity to Systems Traceability Matrix) model. Essentially, the activities from the conceptual model are placed down the left column of a matrix. The entries in the other columns are unknown at this stage, so the matrix dimensions are unimportant. The idea of the SV-5 model can be extended to also include operational node, operational activity, and environment modeling requirements. The benefit of using the SV-5 model for this task is apparent in DSEEP step 3 "Design Simulation Environment".

In addition to the operational activities that need to be modeled by the simulation environment, there are often other requirements on the simulation environment. For example, security requirements, monitoring and control requirements, data collection requirements, timing (real time or faster than real time) requirements, computer and networking constraints, etc. These requirements relate to the activities that the user wants to perform with the simulation environment, or the constraints that the user has on the simulation environment. Thus the SV-5 model with modeling requirements can be supplemented with an additional SV-5 model, which specifies the details related to the simulation environment itself. Constraints that the user has on the simulation environment can be documented in an additional OV-6a (Operational Rules Model) model. Requirements related to the use of standards can be captured in the standards viewpoint, StdV-1 (Standards Profile) model.

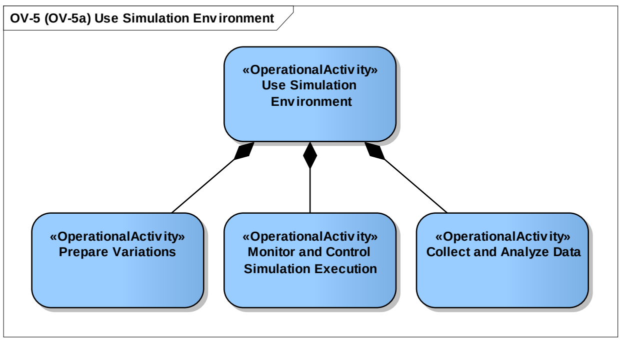

Figure 14: OV-5a (Operational Activity Decomposition Tree).

Figure 14 shows an OV-5a model with activities that the user wants to perform with the simulation environment: i.e. prepare the variations (e.g. threat level variations, number of simulation runs), monitor and control the simulation execution, and collect and analyze the resulting simulation data to produce the required MOEs and MOPs. These activities should be decomposed to an appropriate level of detail to serve as requirements for the simulation environment.

5.5 DSEEP step 3 - Design Simulation Environment

The purpose of DSEEP step 3 "Design Simulation Environment" is to develop the design of the simulation environment. Two approaches are possible in this step as explained in reference [8]: a so called "systems oriented" approach that follows more closely the DSEEP activities, and a "services oriented" approach that involves service oriented activities. The systems oriented approach uses mainly the DoDAF Systems Viewpoint to describe the design of the simulation environment. The services oriented approach introduces the notion of service and uses the DoDAF Systems Viewpoint as well as the DoDAF Services Viewpoint to describe the (service oriented) design of the simulation environment.

5.5.1 Systems oriented approach

The systems oriented approach involves identifying applications that will assume some defined role in the simulation environment (member applications), identifying applications that are suitable for reuse, identifying new applications if required, allocating the required functionality to the identified applications, and developing detailed planning documents. As agreements on assigned responsibilities are negotiated, various simulation environment design trade-off investigations may be conducted as appropriate to support the development of the simulation environment design. Many of these investigations can be considered to be early member application implementations and may include technical issues such as time management, federation management, infrastructure design, runtime performance, and potential implementation approaches.

To select the member applications, the SV-5 model that was started in DSEEP step 2 "Perform Conceptual Analysis" can now be extended to include existing models and simulations that could potentially satisfy the requirements. This process starts as an open-ended brain-storming session where all models with the potential to satisfy any or all of the requirements are listed. They can be placed in the SV-5 matrix as column headings, allowing models to be matched with requirements. If a model has a capability to satisfy a requirement, a suitability value can be placed in the SV-5 matrix at that intersection. In this way, the SV-5 model will trace the requirements to the models best able to analyze them.

Figure 15 provides an example of the SV-5 model, with additional columns for potential member applications and suitability values in the range 1 to 3. Obviously other kinds of values than a suitability value may be used too, such as a value to indicate the maturity or risk associated with a potential member application.

Source \ Target	SV-1 Candidate::SIM F	SV-1 Candidate::SIM E	SV-1 Candidate::SIM D	SV-1 Candidate::SIM C	SV-1 Candidate::SIM B	SV-1 Candidate::SIM A
1 OV-5::Deploy Weapon	2					
1.1 OV-5::Launch	2					
1.2 OV-5::Fly	2					
1.3 OV-5::Detonate	1					
2 OV-5::Situation Awareness		2	3	3		3
2.1 OV-5::Assess RWO		2	2	2		2
2.1.1 OV-5::Classify RWO		2	2	2		2
2.1.2 OV-5::Identify RWO		2	2	2		2
2.1.3 OV-5::Resolve Identification and Classification Conflicts			2	2		2
2.2 OV-5::Compile CTP		2	3	3		3

Figure 15: SV-5 (Operational Activity to Systems Traceability Matrix) – suitability table.

Once an exhaustive list of potential member applications is compiled and the resulting requirements mapping is complete, the selection process begins. If there is one single standalone application that can sufficiently satisfy the requirements, a distributed simulation environment may not be necessary. If not, the required member applications can be selected by comparing capabilities and requirements. Some member applications can be eliminated right away; others will require further scrutiny. It is an iterative process where many member applications are eliminated, some added, and the composition of the simulation environment gradually takes shape. The SV-5 matrix will be reduced in size as member applications are eliminated, and the final product will include only those member applications in the simulation environment. Member applications are selected based not only on their ability to satisfy a requirement but also on run-time performance, availability, personnel expertise, cost, etc. This selection process should be accomplished in a team environment. If there are requirements not met by any of the potential member applications, it may be necessary to enhance an existing member application or to design a completely new one.

Once the candidate member applications are selected, the responsibility for modeling the operational activities from the OV-5 models is allocated to the individual member applications. The SV-5 matrix is updated to reflect the allocation, as shown in Figure 16. Operational information exchanges between swim lanes in the OV-5 models will potentially lead to information exchanges between member applications.

Source \ Target	SV-1::Unit	SV-1::SA Federation	SV-1::RWO Generator	SV-1::Run Time Infrastructure	SV-1::Recorder	SV-1::Execution Manager	SV-1::Dashboard	SV-1::Coordinator
1 OV-5::Deploy Weapon			↑					
1.1 OV-5::Launch			↑					
1.2 OV-5::Fly			↑					
1.3 OV-5::Detonate			↑					
2 OV-5::Situation Awareness	↑							↑
2.1 OV-5::Assess RWO	↑							↑
2.1.1 OV-5::Classify RWO	↑							
2.1.2 OV-5::Identify RWO	↑							
2.1.3 OV-5::Resolve Identification and Classification Conflicts								↑

Figure 16: SV-5 (Operational Activity to Systems Traceability Matrix) – allocation table.

The allocation of responsibility for modeling the operational activities may also be visualized using the allocation relationship as shown in Figure 17. In this example the allocation relationships are stereotyped with IsCapableOfPerforming. Note that the figure shows exactly the same information as the SV-5 matrix, only the representation is different.

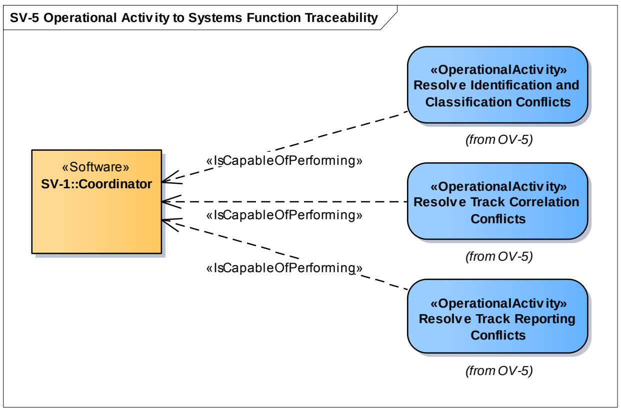

Figure 17: SV-5 (Operational Activity to Systems Traceability Matrix) – graphical.

Also the responsibility to produce and collect the required performance data shown earlier in the OV-1c model is allocated to the member applications. This allocation works in a similar way as the allocation of operational activities.

After the basic member applications responsibilities are determined, diagrams showing how the various member applications relate can help the team visualize and clarify relationships in preparation for simulation environment development. The SV-1 (Systems Interface Description) lends itself to this description. It is meant to identify nodes (in this case, the member applications) and their connections. It can be used to start the process of determining which member application will model which objects and activities, a process that leads to DSEEP step 4 “Develop Simulation Environment”.

An example of an SV-1 model is provided in Figure 18. In this example the structure of the simulation environment is represented. The figure shows a software component named SA Federation, which is composed of a software component named Execution Manager, a software component named Real World Object Generator, etc. The composition relationship shows that the SA Federation is composed of one Execution Manager that fulfills the role execution management, one Real World Object Generator that fulfills the role of threat generation, and one or more Units that fulfill the role unit simulation. The SV-1 model may also include information about deployment of member applications, such as the physical location.

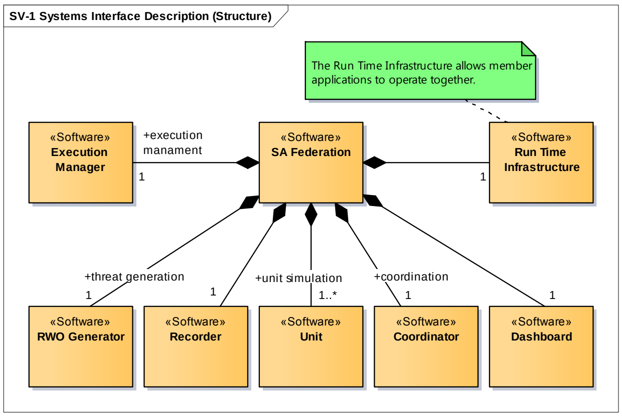

Figure 18: SV-1 (Systems Interface Description).

Simulation environment performance characteristics can be captured in a SV-7 (Systems Measures Matrix) model. For example, the minimum and maximum execution time that the simulation environment and each of its member applications shall support can be described here. Since the structure of the simulation environment is known at this stage, required performance characteristics can be allocated to individual member applications.

5.5.2 Services oriented approach

In [6] a service is defined as: “*A service is value delivered to another through a well-defined interface and available to a community (which may be the general public). A service results in work provided to one by another.*” A service is provided by a participant acting as the provider of the service for use by others. Service orientation is an approach to the design of heterogeneous, distributed systems in which solution logic is structured in the form of interoperating services, provided and consumed by participants.

The services oriented approach in the context of the DSEEP is described in [8] and involves identifying services, allocating modeling responsibilities to services, specifying service interfaces and service agreements, specifying relationships between services, evaluating service realization options by member applications, and designing member applications.

The service oriented activities in this step are:

- Identify services: identify candidate services that are involved in the simulation environment.
- Specify services: elaborate and detail identified services, and specify service interfaces.
- Realize services: evaluate service realization options and decide on which member application will realize what service.

This approach is not re-iterated here, but a number of examples of models are provided below.

When analyzing the conceptual model and simulation environment requirements the following services may be identified: Real World Object Generation Service, Surveillance Service, Information Management Service, Recording Service and Execution Management Service.

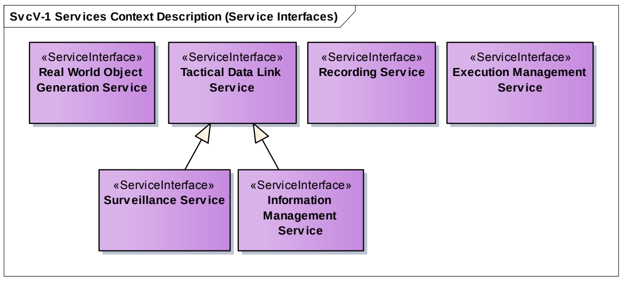

Figure 19: SvcV-1 (Services Context Description).

The hierarchy of identified services can be documented in a SvcV-1 (Services Context Description) model, as shown in Figure 19. The elements in the hierarchy are service specifications (i.e. service interfaces), and the relationships between the elements are generalizations (i.e. one service is a general type of another). This example shows that the Surveillance Service and Information Management Service are a specialization of the Tactical Data Link Service.

A SvcV-2 (Services Resource Flow Description) model can be used to specify the service interfaces. A service presents one or more interfaces to consumers and uses one or more interfaces exposed by other service providers. Figure 20 shows the interfaces associated with the Information Management Service, Surveillance Service, and Real World Object Generation Service. The Surveillance Service, for example, provides a Surveillance interface (to receive Air Tracks on), and uses a Surveillance Interface (to send Air Tracks on).

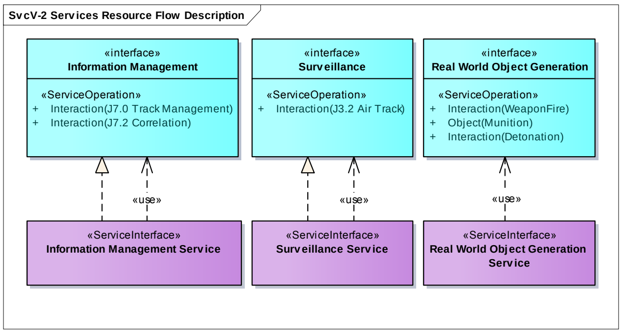

Figure 20: SvcV-2 (Services Resource Flow Description).

Service realization can be documented in a SvcV-3a (Systems-Services Matrix) model. This model references the member applications described in the SV-1 model and indicates what member application will realize what service. Figure 21 provides an example how this relationship can be established, using the UML realizes relationship between service participant and member application. A service participant is a SoaML model element and is defined as an abstract provider or consumer of a service. In this example it provides and requests a number of services via service ports. By using the UML realizes relationship a member application inherits the service ports that are defined for the service participant. The service ports are reflected in the SV-2 model that is developed in the next step of the DSEEP.

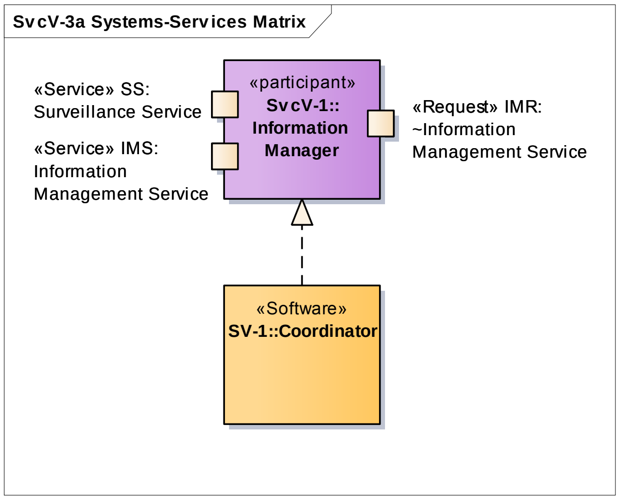

Figure 21: SvcV-3a (Systems-Service Matrix).

Other models can be developed to document service constraints (SvcV-10a), service states and interactions (SvcV-10b and SvcV-10c), and the functions that a service is expected to perform (SvcV-4). The service realization activity also covers the development of several system view models discussed earlier in section 5.5.1.

5.6 DSEEP step 4 - Develop Simulation Environment

The purpose of DSEEP step 4 "Develop Simulation Environment" is to define the information (the Simulation Data Exchange Model, SDEM) that will be exchanged at runtime during the execution of the simulation environment, to develop a set of simulation environment agreements, and to develop or modify member applications as necessary. Agreements among member applications include agreements on common/consistent databases and algorithms, initialization procedures, synchronization points, save/restore policies, and security procedures.

When a services oriented approach was followed in DSEEP step 3 the SDEM and agreements need to be assembled from the individual service specifications, constraints and other models developed in DSEEP step 3.

The DoDAF products are very useful during the initial phase of the SDEM development. The simulation environment SV-1 model developed in DSEEP step 3 "Design Simulation Environment" can be updated to include additional member application responsibility details. The information that will have to be communicated between member applications will be further specified.

The allocation of responsibility of modeling operational activities to member application will lead to requirements on information exchange between member applications. The information exchange requirements can be specified in a DIV-2 (Logical Data Model) model. The entity items in the DIV-2 model represent the data exchanged by member applications. The entity items may be grouped in modules, and augmented with attributes and data types.

Figure 22 and Figure 23 provide examples of DIV-2 models. Figure 22 shows three modules with entity items that are described in additional DIV-2 models. Figure 23 shows the entity items of the Link16 module. This figure also shows the relationship between the operational exchange elements in the DIV-1 model developed earlier in DSEEP step 2 "Perform Conceptual Analysis" and the entity items in the DIV-2 model.

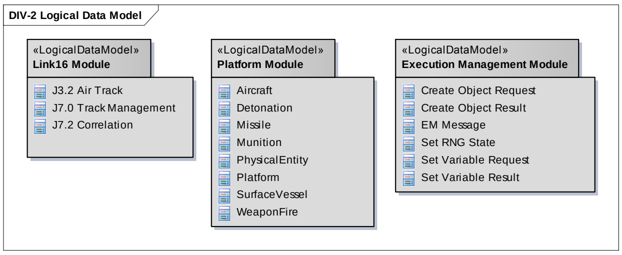

Figure 22: DIV-2 (Logical Data Model): Modules.

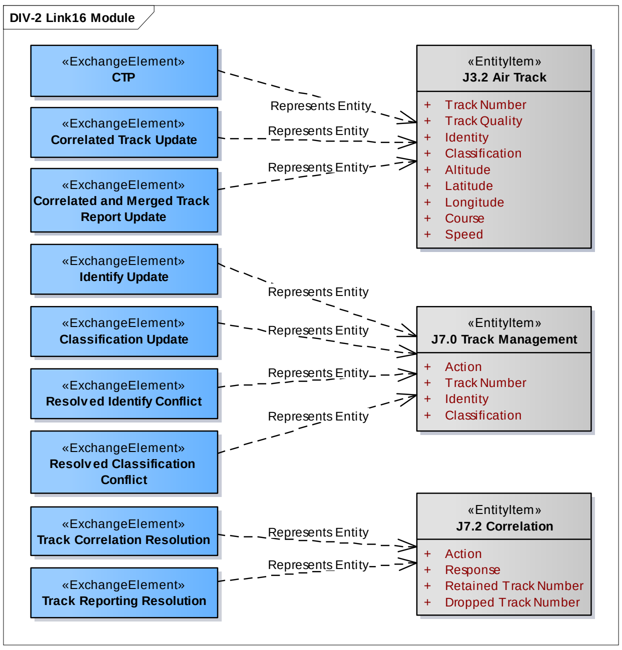

Figure 23: DIV-2 (Logical Data Model): Link16 Module.

Based on the data items specified in the DIV-2 models, the Simulation Data Exchange Model (SDEM) can be developed. A visual representation of the objects, interactions, and the attributes of each is useful in producing a well-designed SDEM and the DIV-2 provide this framework. The SDEM itself is at the level of a DIV-3 (Physical Data Model); for HLA this corresponds to the Federation Object Model (FOM).

At the same time the SV-1 model is expanded to describe the resource flow between member applications. This will lead to a SV-2 (Systems Resource Flow Description) model that shows the member applications and the data they exchange. Depending on the situation it may be useful to combine the SV-1 and SV-2 models in a single SV-1 model. The SV-2 model shown in Figure 24 provides an example of a simulation environment with a number of member applications and the information flow between them: a Coordinator (CO), a number of Units (UNx) and a Real World Object Generator (TG). For simplicity, Recorder, Dashboard, Execution Manager and Run Time Infrastructure are left out of the picture. In this example, each member application has one or more ports to exchange information with other member applications. A port has a name, e.g. P1.1, followed by the name of the service that is provided or requested via the port. The ports and services refer in fact to the service ports defined earlier in DSEEP step 3 in section 5.5.2.

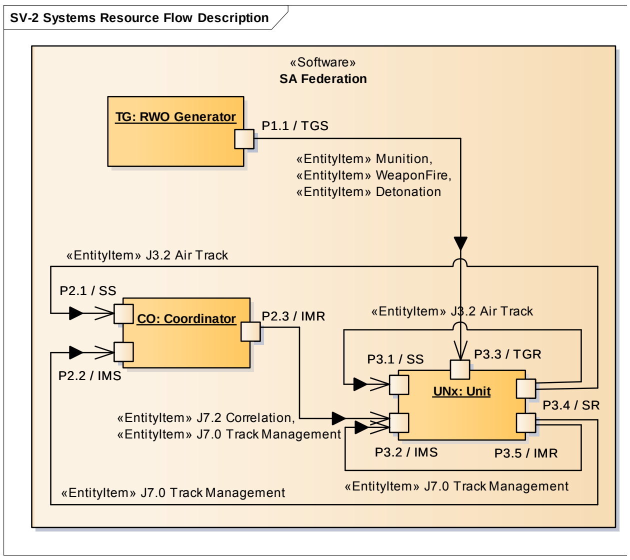

Figure 24: SV-2 (Systems Resource Flow Description).

Once the data exchange between member applications is specified, the entity items become elements of the SV-6 (System Resource Flow Matrix) model. The standard SV-6 matrix lists the complete data exchange between each communication port of each member application. Since the direction of the data exchange is known, additional SV-6 models may be created to list the data produced by or consumed by each member application. These additional models reflect in fact the publication and subscription agreements between member applications. An example of an SV-6 Publication matrix is shown in Figure 25. Using the appropriate modeling tools, these matrices can be generated automatically.

Producer	Conveyed_Type	Conveyed_Name
⊟ Producer: CO		
CO	EntityItem	J7.2 Correlation
CO	EntityItem	J7.0 Track Management
⊟ Producer: TG		
TG	EntityItem	Munition
TG	EntityItem	WeaponFire
TG	EntityItem	Detonation
⊟ Producer: UNx		
UNx	EntityItem	J7.0 Track Management
UNx	EntityItem	J3.2 Air Track

Figure 25: SV-6 (Systems Resource Flow Matrix).

As the development progresses beyond the structural design and the specification of the SDEM, towards a more functional design and the development of simulation environment agreements, various models may be used to capture agreements and describe the way in which simulation architecture services are used. Simulation environment agreements cover issues such as initialization, synchronization, termination, progression of time, events, life cycle of entities, update rates, etc.

An important set of agreements concerns simulation execution management. Execution management involves amongst others the initiation and coordination of state transitions within the simulation environment, and the management of activities performed in each state. The SV-10b (Systems State Transition Description) model can be used to define the simulation execution states, triggers, actions and conditions using a UML state transition diagram. Figure 26 provides an example, where HLA synchronization points are used to coordinate state changes.

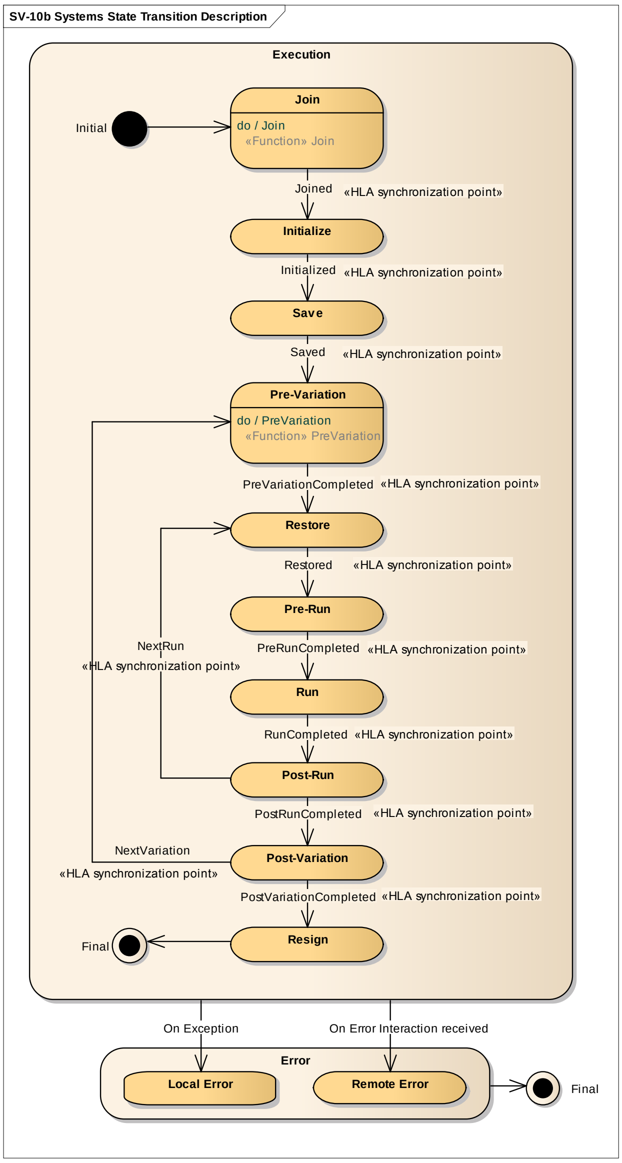

Figure 26: SV-10b (System State Transition Description).

For each state in the state transition diagram, activities may be defined that are to be performed in that state. These activities (or functions) are described in the SV-4 (Systems Functionality Description) model, and are cross-referenced from the SV-10b model. For example, in the Join state and PreVariation state, the Join function and PreVariation function are to be performed. The specifics of these functions are shown in Figure 27 and Figure 28. Functions may also require data exchange between member applications, as is the case with the PreVariation function.

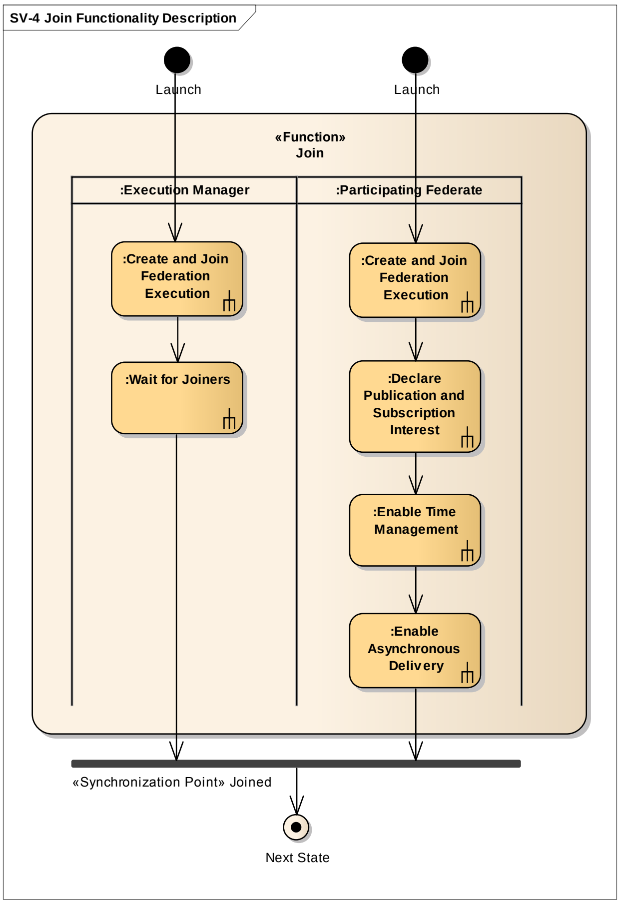

Figure 27: SV-4 (Systems Functionality Description): Join.

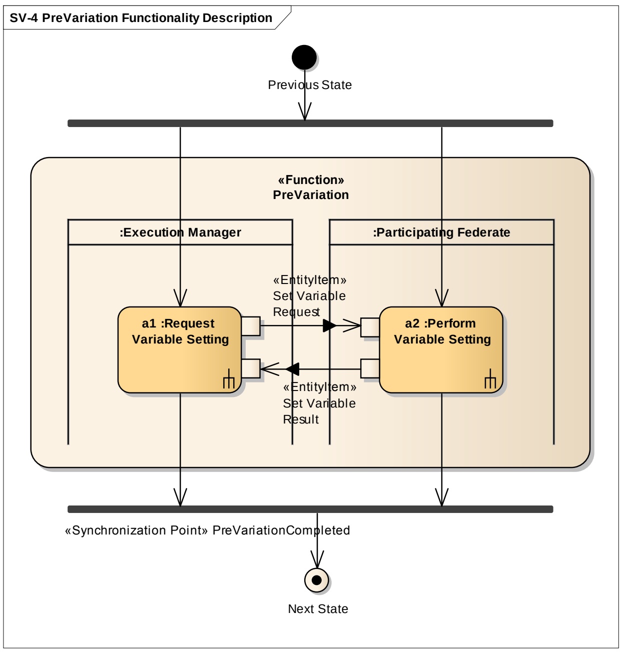

Figure 28: SV-4 (Systems Functionality Description): PreVariation.

Agreements on the transmittal of messages over time between member applications may be captured in the SV-10c (Systems Event-Trace Description) model. Messages include the update of object attributes and the transmittal of interactions. Figure 29 shows the message exchange related to the launch, flight and detonation of a missile. These UML sequence diagrams are also useful input to the development of test cases for the simulation environment.

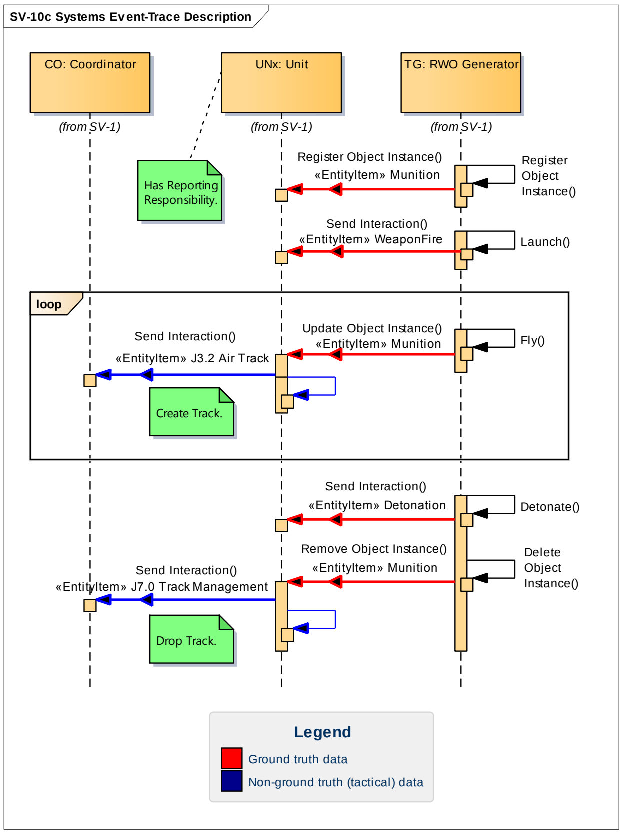

Figure 29: SV-10c (Systems Event-Trace Description).

Other agreements on for example the progression of simulation time and the ownership of object attributes may be described in the SV-10a (Systems Rules Model) model. This model is used to describe the rules or constraints under which the architecture or its systems behave under specified conditions. The DoDAF does not provide a specific format, so that is up to the author. A textual document describing the simulation environment agreements in detail may also be useful.

6 Guide to documentation

The previous chapter provided many examples of architecture models that may be developed in DSEEP steps 1 to 4. Depending on the kind of project there are different approaches as to how these architecture models and the DSEEP products in general can be documented. The models can be captured in different documents, or can all be rolled up in just a couple of documents. To take as example the case study used in the previous chapter, the approach here would be to document the simulation environment architecture in just two documents called the "Experimentation Plan" and the "Simulation Environment Design Description".

The Experimentation Plan captures the models of DSEEP steps 1 and 2. This plan describes amongst others the objectives of the simulation environment (including experimentation hypotheses), the initial plan for the development and execution of the simulation environment, the conceptual model and scenario, and the simulation environment requirements. The plan should include sufficient detail to guide the design activities in DSEEP steps 3 and 4, and should also define a high-level schedule of key design, development and execution events.

The Simulation Environment Design Description captures the models of DSEEP 3 and 4. The design describes amongst others the member applications of the simulation environment, the modeling responsibilities, the simulation data exchange model, and the simulation environment agreements. This document should include sufficient detail to guide the development, integration and test activities in DSEEP steps 4 and 5.

In defining an outline for both documents reference [9], an IEEE standard for architecture description, can be used as guidance. This standard specifies the content topics that should generally be included in an architecture description. Since the DoDAF is used as architecture framework, the used architecture viewpoints for both the Experimentation Plan and Simulation Environment Design Description are a given. Using the content topics listed in [9], the main outline for both documents is shown below. The outline also lists the DoDAF model kinds per chapter.

Experimentation Plan

1. Architecture description identification and overview. Title of document, version, date of issue, summary, scope, and references.
2. Overview and Summary.
    - AV-1
3. Needs and objectives.
    - CV-1, OV-1, OV-1c, OV-1d
4. Initial plan.
    - PV-1, PV-2, PV-3
5. Conceptual model.
    - OV-2, OV-3, OV-4, OV-5, OV-6, DIV-1
6. Scenario.
    - OV-6
7. Simulation Environment Requirements.
    - SV-5 or SvcV-5
8. Architecture rationale.
9. Terminology.
    - AV-2

Simulation Environment Design Description

1. Architecture description identification and overview. Title of document, version, date of issue, summary, scope, references.
2. Overview and Summary
    - AV-1
3. Simulation environment design.
    3.1. Services view.
    Services identification.
    - SvcV-1, SvcV-5, SvcV-7

    Services interfaces.
    - SvcV-2, SvcV-4, SvcV-6, SvcV-10, DIV-2

    Services realization.
    - SvcV-3

    3.2. Systems view.
    Member applications.
    - SV-1, SV-2, SV-3, SV-5, SV-6, SV-7

    Simulation data exchange model.
    - DIV-2

    Simulation environment agreements.
    - SV-4, SV-10

    3.3. Standards view.
    - StdV-1
4. Architecture rationale.
5. Terminology.
    - AV-2

7 Summary

This paper provided an overview of the DoDAF and the application of the DoDAF to simulation environment architecture development. Although the DoDAF was not designed for simulation environment architecture development, this paper has shown that many of the architecture viewpoints and views in the framework can be applied to simulation environment architecture development. Some fit for purpose models may still be needed, e.g. to document operational context measurements and use cases. Most relevant and useful viewpoints are the Operational Viewpoint for conceptual analysis, the Services and Systems Viewpoints for simulation environment design and development, and the

Data and Information Viewpoint for capturing conceptual model data and simulation data exchange model data.

Most of the DoDAF viewpoints nicely fit in the DSEEP. The Services Viewpoint, however, is currently not well addressed by the DSEEP, in particular by step 3. Suggested service oriented activities in step 3 are described in [8] and the development of a services overlay for the DSEEP is one of the topics of NATO MSG-136 [10].

This paper also demonstrated the application of UPDM for describing DoDAF models. UPDM can be used for describing the SOI, as well as the simulation environment representing the SOI. It provides a common language for both. Since UPDM includes SysML, many of the models can be described with SysML. For the Services Viewpoint SoaML is useful.

This paper can also be applied to other related architecture frameworks, such as MODAF (UK Ministry of Defence Architecture Framework) and NAF (NATO Architecture Framework). Reference [4] includes a mapping of models between DoDAF, MODAF and NAF.

This paper has (re-)used some of the results of an earlier SISO paper on architecture description development ([11]) and the authors acknowledge the work done in this earlier paper.

8 References

[1] "DoD Architecture Framework Version 2.02", US DoD, August 2010.

[2] "IEEE Standard 1730-2010 - IEEE Recommended Practice For Distributed Simulation Engineering and Execution Process (DSEEP)", Institute of Electrical and Electronics Engineers, 2010.

[3] http://www.omg.org.

[4] "Unified Profile for DoDAF and MODAF (UPDM), Version 2.1", Object Management Group, August 2013.

[5] "Systems Modeling Language (SysML), Version 1.3", Object Management Group, June 2012.

[6] "Service Oriented Architecture Modeling Language (SoaML), Version 1.0.1", Object Management Group, May 2012.

[7] http://www.sparxsystems.com.

[8] "Service Orientation for the Design of HLA Federations", SISO Simulation Interoperability Workshop, Paper 14F-SIW-048, Fall 2014.

[9] "IEEE Standard 42010-2011 - Systems and software engineering - Architecture description", Institute of Electrical and Electronics Engineers, 2011.

[10] MSG-136: Modelling and Simulation as a Service (MSaaS), Rapid deployment of interoperable and credible simulation environments; http://www.cso.nato.int/ACTIVITY_META.asp?ACT=5642.

[11] "The Application of the DoDAF within the HLA Federation Development Process", SISO Simulation Interoperability Workshop, Paper SIW 04F-SIW-015, Fall 2004.

9 Author biographies

TOM VAN DEN BERG is a scientist in the Modeling, Simulation and Gaming department at TNO, The Netherlands. He holds an M.Sc. degree in Mathematics and Computing Science from Delft Technical University and has over 25 years of experience in distributed operating systems, database systems, and simulation systems. His research area includes simulation systems engineering, distributed simulation architectures, systems of systems, and concept development & experimentation.

ROBERT LUTZ is a principal staff scientist at The Johns Hopkins University Applied Physics Laboratory in Laurel MD (USA). His background includes 34 years of practical experience in the development, use, and management of models and simulations across all phases of the DoD systems acquisition process. He has led the development of several M&S standards, including the IEEE 1730 Distributed Simulation Engineering and Execution Process (DSEEP) and the IEEE 1730.1 DSEEP Multi-Architecture Overlay (DMAO). Mr. Lutz also serves as the Chair of the Simulation Interoperability Standards Organization (SISO) Board of Directors, serves on the Tutorial Board and Fellows Committee at the Interservice/Industry Training, Simulation and Education Conference (I/ITSEC), and is a guest lecturer on M&S-related topics in The Johns Hopkins University Whiting School of Engineering.