Applying a Methodology to Identify Structural Variances in Interoperations

James A. Muguira
Systematic Sofware Engineering Inc.
Fairfax, VA 22030
jam@sseusa.com

Andreas Tolk
Virginia Modeling Analysis and Simulation Center
Old Dominion University
Norfolk, VA 23529
atolk@odu.edu

Specification of the interoperation of components in a system-of-systems is a current research problem. Though the literature has identified several means of breaking this problem into levels, the literature also reveals that no single specification scheme can cover all levels of interoperability well. The base to be covered is broad: one needs an overarching model, such as the DoDAF for the military domain, to guide the capture of conceptual detail and at the same time provide a low-level base set of implementation specific details, such as DEVS, and protocols, like HLA, that create the study system.

This paper will describe a framework that will merge the various specification schemes and apply a methodology to define the structural variances between components. An example of multiresolution modeling will demonstrate the application of the methodology.

Keywords: Agent-based simulation, continuous and discrete simulation, discrete-event simulation, distributed simulation

1. Introduction

The modeling of problems and the simulation and study of the problem on a computer is relatively young, but the field is changing and improving very rapidly. Today's environment contains many models of various aspects of the craft of warfare. Being designed and certified in a stand-alone mode the question is: Can these models be joined into a larger system-of-systems to study larger problems? As current researchers in the field are finding, this is a question with very challenging solutions. From a research perspective, the answer to this question is yes, we can compose models together, but there is a much more subtle challenge involved. The composition of these models requires a lot of time and expertise to find a mapping to bring the models together. Much of the expertise comes not from analysts studying the problem but from the programmers who must change the underlying code and from the subsequent discussions that the programmers and analysts need to have in order to create a good composition. Research into executable frameworks and directly executable architecture specifications may help to reduce the burden of transition from specification to executable system ready for experimentation, but there is still a gap between the specification and what is needed to complete the executable framework or architecture. This paper will describe a specification framework and methodology that help to close some of that gap.

In the ideal situation, the analyst captures systems specification information and builds models. These models are then translated via automatic tools into a form that can be studied on a computer. The Discrete Event System Specification (DEVS) [1] is one example of such a tool set that can render executable models. DEVS is a mathematical formalism that can specify a system and then provide multiple "back-end" systems that serve to execute the model and produce analyzable output. At the high level, using a DEVS-based tool, an analyst can model a system using a graphical notation that is similar to colored Petri nets. Once the model has been specified the system can be translated onto an executable structure for simulation. The DEVS system is not perfect, since for many real-world simulation

JDMS, Volume 3, Issue 2, April 2006 Pages 77–93

situations the services of a programmer are still needed.

The situation today is that there are thousands of tested and certified models in use, and as pointed out earlier, composition of these models is greatly desired. The good news is that the gap between the static specification of the system and the executable scheme is closing. This paper will demonstrate a framework and methodology that improve the specification of the system by identifying and controlling error-causing variance.

This paper utilizes different research efforts conducted at the Virginia Modeling Analysis and Simulation Center (VMASC) of Old Dominion University (ODU) in Norfolk, Virginia. The role of VMASC is special: it is, on the one hand, an enterprise of the ODU Research Foundation conducting funded research driven by the immediate needs of M&S users—many of them working in the U.S. Department of Defense (DoD) domain—and VMASC is, on the other hand, an ODU-affiliated center educating masters and Ph.D. students in M&S. This combination ensures academically highly-qualified results applicable to real problems.

The remainder of the paper will discuss the foundational aspects of model composability through specification, the simulation interoperability aspects, and the core intellectual tools that support the joining of modeling and simulation. The next section will describe the Levels of Conceptual Interoperability Model and its power to structure and organize the study of interoperability. Next, the U.S. Department of Defense Architecture Framework (DoDAF) is described and motivated as the core foundation of a framework to improve model specification [2]. The fourth section of the paper will present an example multiresolution modeling problem, apply the framework methodology, and demonstrate applicability. The paper will end with some concluding remarks.

2. Overview Levels of Conceptual Interoperability Model

The Levels of Conceptual Interoperability Model (LCIM) underwent several improvements and enhancements since its introduction by Tolk and Muguira [3]. The LCIM is founded on the idea that interoperability of systems is not a cookie-cutter function. There are various levels of interoperability between two systems ranging from no interoperability to full interoperability. In the technical domain, various models for levels of interoperability already exist and are used successfully to determine the degree of interoperability between information technology systems. The Levels of Information System Interoperability (LISI) [4] model developed within the U.S. DoD community or the NATO Interoperability Model (NIM) [5], which is described in NATO's Consultation, Command, and Control Technical Architecture (NC3TA), are examples. However, such models are not yet established in the domain of conceptual modeling. The current version of LCIM as depicted in Figure 1 is documented in Turnitsa [6]. It was developed to become a bridge between the conceptual design and the technical design for implementation, integration, or federation.

The LCIM introduced seven layers to address the different aspects of interoperation. The different levels are characterized as follows:

- Stand-alone systems have *no interoperability*.
- On the level of *technical interoperability*, a communication protocol exists for exchanging data between participating systems.
- The *syntactic interoperability* level introduces a common structure to exchange information, i.e., a common data format is applied.
- If a common information exchange reference model is used, the level of *semantic interoperability* is reached. On this level, the meaning of the data is shared.
- *Pragmatic interoperability* is reached when the interoperating systems are aware of the methods and procedures that each is employing. In other words, the use of the data—or the context of its application—is understood by the participating systems.
- As a system operates on data over time, the state of that system will change, and this includes the assumptions and constraints that affect its data interchange. If systems have attained *dynamic interoperability,* then they are able to comprehend the state changes that occur in the assumptions and constraints that each is making over time and are able to take advantage of those changes.
- Finally, if the conceptual models—i.e., the assumptions and constraints of the meaningful abstraction of reality—are aligned, the highest level of interoperability is reached: *conceptual interoperability*. This requires that conceptual models be documented based on engineering methods enabling their interpretation and evaluation by other engineers.

The seven layers will be explained in more detail in the next subsections. Furthermore, their relation to the categories composability, interoperability, and integratability as introduced by Page et al. [7] will be explained.

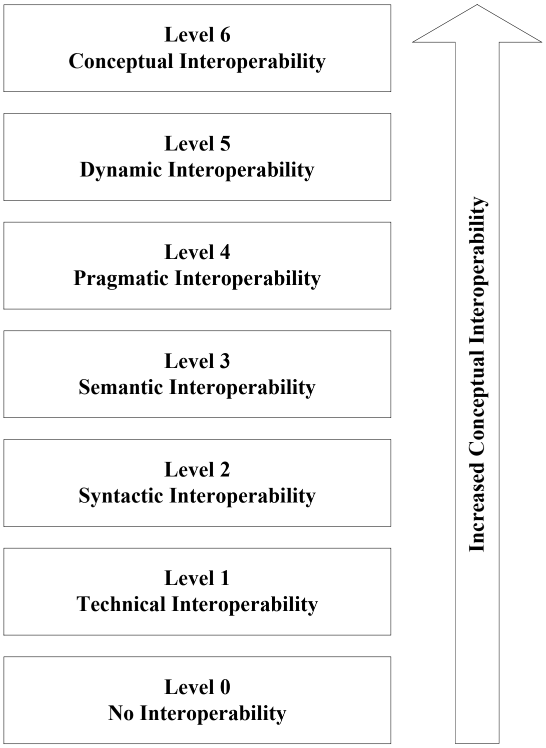


Figure 1. Levels of conceptual interoperability

2.1 The Integratability Level: Technical Solutions

The first level in the LCIM describes the state of stand-alone systems not being connected to any other systems. It was included for completeness of the model.

The first level described is technical interoperability, which comprises the physical connections necessary to enable information exchange. This level deals with hardware and firmware requirements and lower communication protocols, referred to as means of integration enabling *integratability* of components. This also includes problems of the underlying network including network connectivity. In simple terms, this layer enables the exchange of bits and bytes and their interpretation.

2.2 The Interoperability Levels: Software Solutions

The following two levels in the LCIM address implementation issues of interoperation, supporting *interoperability* of components.

First, the structure of information must be understood, which is the domain of syntax. The protocols ensuring that the data is interpreted in a common format—or that data can be exchanged following agreed structural conventions—is the domain of syntactical interoperability. The structure of the protocol data units (PDUs) of the distributed interactive simulation (DIS) protocol, the structures defined in the Interface Control Document (ICD) of the Aggregate Level Simulation Protocol (ALSP), the Federation Object Model (FOM) structures defined by the Object Model Template (OMT) of the High Level Architecture (HLA), and the Interface Description Language (IDL) of the Common Object Request Broker Architecture (CORBA) belong to this level: they all ensure that data can be exchanged following an agreed structure. However, these protocols support the structure of the information, not the content.

The inherent meaning of data is captured in the semantics. To capture the meaning of data is a non-trivial task and is not addressed by many standards. Generally, this task is left to the interpretation of the implementer. Verification and validation determine if the data chosen to populate the structure of a platform is congruent with the interpretation on the receiving side. Data management is planning, organizing, and managing data by defining and using rules, methods, tools, and respective resources to identify, clarify, define, and standardize the meaning of data and of their relations. Tolk [8] introduced the idea of model-based data management. Tolk and Diallo [9] generalized the ideas to be applicable for Internet computing. Within the DoD, the net-centric data strategy is intended to address this problem [10].

Nonetheless, no generally accepted standards have been established in the M&S community so far. In the application domain of HLA, it is still current practice to document these agreements in prose and distribute the document to support the implementers. The use of common reference models to establish not only the structure but also the meaning of the data to be exchanged in an unambiguous way is the topic of current research.

A strong candidate for a reference model for information exchange in the military domain is the Command and Control Information Exchange Data Model (C2IEDM) [11]. C2IEDM comprises data elements describing a common vocabulary consisting of 176 information categories that include over 1,500 content elements. It lays down a common approach to describe the information to be exchanged and is not limited to a special level of command, force category, etc. In general, C2IEDM describes all objects of interest on the battlefield, e.g., organizations, persons, equipment, facilities, geographic features, weather phenomena, and military control measures such as boundaries. The complete data model documentation and additional information is available on the Internet; see the website of the Multilateral Interoperability Programme (MIP)

[11]. The U.S. Army M&S Executive Council (AMSEC) recently accepted the recommendation to adopt C2IEDM as a reference model for coping with M&S to C2 system interoperability [12].

In particular, the M&S standards currently available support, at most, semantic interoperability by using common reference models such as the information captured within PDUs or the Real-time Platform Reference Federation Object Model (RPR-FOM). While these are necessary, they are not sufficient for meaningful interoperation. During a recent panel discussion on priorities for M&S standards, Professor Bernard P. Zeigler explicitly stated in his presentation that *standardization must be aimed at the modeling level* to ensure interoperability between systems; i.e., the standardized level must be higher than the programming level standards currently applied.[1] The higher composability levels of the LCIM do this.

2.3 The Composability Levels: Conceptual Solutions

The upper three levels of the LCIM deal with M&S–specific issues—or, following the arguments of Hofmann [13], *model-based information processing system*–specific issues. This is the domain of *composability*: aligning the assumptions and constraints of the underlying models and conceptualizations. However, it will become apparent that the frontier between *interoperability* and *composability* is "fluent," and the applicability of methods is not mutually exclusive.

The first level of the LCIM dealing with composability is the level of pragmatic interoperability. While semantics addresses the inherit meaning of data, pragmatics places data into the context of their use within the applications. As pointed out in the NATO Code of Best Practice for Command and Control Assessment [14], data must always be seen in the context of their application. While data may be sufficient and applicable for an analytic environment, their use for operational decision support may be insufficient. To this end, metadata dealing with source, reliability, state of accreditation and certification, credibility, resolution, and more are recommended.

A closely related aspect is how the transmitted data will influence the state of the sending and the receiving system. In current standards, this is often dealt with initially by defining pre- and post-conditions. However, this is often insufficient. While open source implementations are desired to support transparency of implementations, intellectual property rights and commercial interests often speak against this. Reference implementations as well as extensive documentation may be an alternative. The next section will address this issue more intensively.

Finally, the LCIM introduces a level of conceptual interoperability. The motivation for this additional level is that in the process of modeling a system the decision of what to model is sometimes overshadowed by the decision of what not to model. As stated before, *a model is a meaningful abstraction of reality*, and the levels below the conceptual level can only be applied to what *is* implemented, and not to what *is not* implemented (and potentially why the decision for exclusion was made). When two components interoperate, these decisions may become important. In particular, when two underlying models are conceptually interdependent, but both models excluded mapping aspects, the interdependence can only be seen on this highest level. Capturing the conceptualization's underlying components or federates to be used in a common system or system-of-systems is therefore essential. In the era of net-centricity and net-enabled operations in which services have to be composed on the fly, the importance even increases.

2.4 Composability, Interoperability, and Integratability Challenges

Generally, the underlying questions to address when talking about simulation system interoperability are: "Is interoperability of simulation systems different from interoperability of other IT systems?" and "What makes simulation systems special?" In the context of a special issue on *integrated system evaluation* dealing with solutions to enable net-centric operations and applications in service-oriented architectures, this research must result in recommendations on how to integrate simulation systems into net-centric or—using the term established within NATO—net-enabled environments, which include, in particular, the definition of necessary metadata. These metadata define the profile of the simulation system and shall enable the identification of applicable systems, the evaluation of the applicability of a special system, the general reuse of systems, and the composition of various services to deliver the operationally required functionality "on the fly."

Page et al. [7] state that within the military simulation domain, *composability* has arisen as a cousin of the longstanding DoD objective of *interoperability*. They support the view of Petty and Weisel [15], whose research resulted in the view that interoperability covers the technical aspects and composability the conceptual aspects. Page et al. [7] therefore suggest

---

[1] Simulation Interoperability Standards Organization (SISO)/ Society for Modeling and Simulation International (SCS) panel discussion on priorities for M&S standards; conducted during the IEEE Spring Simulation Interoperability Workshop in Orlando, Florida, March 2003.

defining composability as the realm of the model and interoperability as the realm of the software implementation of the model. (In addition, they introduce *integratability,* dealing with the hardware side and configuration side of connectivity.) The authors support this categorization and recommend the following distinction with issues of simulation system interoperability, regardless of meaningful simulation-to-simulation system or simulation-to-other-than-simulation system interoperation:

- *Integratability* addresses the physical and technical realms of connections between systems, which include hardware and firmware, protocols, etc.
- *Interoperability* addresses the software and implementation details of interoperations including exchange of data elements based on a common data interpretation, etc.
- *Composability* addresses the alignments of issues on the modeling level. The underlying models are meaningful abstractions of reality used for the conceptualization being implemented by the resulting simulation systems.

However, while Page et al. [7] introduce the categories as orthogonal concepts, the authors take a layered approach. Based on ideas of linguistics, LCIM distinguishes several layers coping with the different aspects of interoperation. Hofmann [13] introduces the term *model-based information processing system* to motivate composability discussions based on the LCIM ideas. He shows that for data-centric solutions, such as databases or tracking systems, the levels of integratability and interoperability are sufficient, but that composability becomes an issue as soon as you use a model to drive an implementation. This answers the questions posted in the beginning of this section: simulation systems are based on models and their assumptions and constraints. If two simulation systems are combined, these assumptions and constraints must be aligned accordingly to ensure meaningful results.

3. Overview of the DoD Architecture Framework

A literature research study conducted by Muguira, as part of ongoing research, showed that current standards support different levels of interoperation, but there is no generally accepted approach covering all aspects of interoperation. In the next section, the authors will propose a framework melding different ideas together to result in a consistent view covering all levels. However, in this paper, the focus is on the composability levels of interoperation. It will be shown that the DoD Architecture Framework (DoDAF) supports many of the necessary aspects already.

The remainder of this section deals with the motivation of using UML for conceptual models in general, introduces the DoDAF and shows how it is related to UML, and prepares the stage for the use of these methods in the recommended framework, described in the next section.

3.1 Conceptual Modeling Using UML and Related Standards

The conceptual model of a simulation is a device for capturing requirements and establishing their traceability throughout the simulation design, implementation, and reuse. As found in the U.S. DoD Instruction (DoDI) 5000.61, the conceptual model is defined as “the developer’s description of what the model or simulation will represent, the assumptions limiting those representations, and other capabilities needed to satisfy the user’s requirements.” To this end, the Recommended Practices Guide for Verification, Validation, and Accreditation (VV&A) states that the conceptual model should at least comprise the following: (a) the *simulation context* providing authoritative information about the domain that the simulation is to address; (b) the *simulation concept* describing the developer’s concept for the entire simulation application (explaining how the developer expects to build the simulation to satisfy the user-defined requirements and establishing constraints and boundary conditions for the simulation concept); and (c) the *simulation elements* consisting of all the information describing the concepts for the individual entities, as well as the concept for the collection of entities, behaviors, and relationships represented in a simulation.

While no standard approach for conceptual modeling has been agreed to, the use of the Unified Modeling Language (UML) [16] as a possible means is discussed in many application domains. Since having been standardized by the Object Management Group (OMG) in 1997, it has become of interest to management consulting firms, business analysts, system analysts, software developers, and programmers. In particular, it can be seen as the de facto standard for blueprints of software solutions. Over the past few years, the UML became something like the *lingua franca* for modeling purposes. Consequently, many artifacts of the DoDAF can be expressed using UML concepts. Industry white papers show the commercial support of these ideas, in particular embracing the new capabilities of UML Version 2.0, such as Telelogic [17]. A tremendous advantage is that UML enables the application of white box techniques to cope with the behavior of the model without having to reveal all the internal details of the implementation. This allows the necessary transparency to ensure alignment of models while at the same time protecting the intellectual property rights of special implementations.

The OMG's Model Driven Architecture (MDA) [18] is based on the UML, which in turn is based on the Meta-Object Facility, which unifies UML, its profiles and MDA models, and the Common Warehouse Metamodel (CWM), which standardizes the metadata interchange. The application of the MDA produces two artifacts: the platform-independent model (PIM) used to describe concepts, which is used to produce the platform-specific model (PSM), which is used to describe the implementation. A possible common view on the different system engineering aspects underlying the MDA, DoDAF, and the Federation Development and Execution Process (FEDEP) of HLA is given by Trbovich and Reading [19]; the authors will use this work in the second application example given in this paper in a following section.

An example for the operational use of UML is published in Blake et al. [20]. They introduce a UML-based model of single ship operations to align the various related projects of the U.S. Navy as a basis for a common concept of operations and use of models and real operations. The Simulation Interoperability Standards Organization (SISO) Study Group on Extensible M&S Framework (XMSF) Profiles promotes the same idea: the behavior of an XMSF component shall be described in UML and then codified using XML Metadata Interchange (XMI) methods.

While the use of UML does not meet all requirements to describe a conceptual model completely, significant aspects are covered. The authors are convinced that the advantages of using UML outweigh the current incompleteness. They recommend the use of UML where possible and the definition of additional views where needed to close the gap. These add-ons can be standardized by bodies like SISO in parallel and give feedback to the UML community as required enhancements. It seems to be better to start with a generally accepted "85% solution" and add new ideas to cover all requirements than to start from the beginning and have to go through all the necessary alignment with other user groups.

3.2 Technical Overview of DoDAF

One of the major challenges is the definition of a conceptual model capturing the operational ideas, the constraints, and the assumptions of the implemented models. This is true for every application domain. In the military domain, the DoDAF is a potential candidate for this. DoDAF is embedded into the systems engineering process, which includes the system design process composed of the requirements definition process and solution definition process EIA [21]. Lee et al. [22] show how the requirement definitions can be used to generate the necessary operational and system views of the DoDAF using an example of the Korean Armed Forces embedded in the systems engineering process. NATO applied the NATO Consultation, Command, and Control (C3) Systems Architecture Framework, which is the NATO equivalent of the U.S. DoDAF, within their Active Layered Theatre Ballistic Missile Defense (ALTBMD) study to align the command and control requirements and procurement efforts of the participating nations in setting up a missile defense initiative for Europe supported by simulation experiments on a big scale; Adshead et al. [23] describe the general process. In summary, the application of DoDAF for command and control systems is current practice. Furthermore, the Clinger-Cohen Act of 1996 mandated federal agencies to develop an enterprise IT architecture. For large-scale IT systems of the DoD, the DoDAF is mandatory. The authors will give a short overview on related M&S work at the end of this section.

In many articles, the artifacts or "products" that are produced only characterize DoDAF. However, DoDAF provides four broad categories of guidance for developing architectures, which are *guidelines* to build DoDAF-compliant architectures, a *high-level process* for developing architecture descriptions that fulfill a given purpose (describing a concept), *data and tool discussions* to facilitate descriptions of these architectures (implementation aids), and finally *detailed product descriptions*, the artifacts describing the different complementary views that are further described below. The resulting products are intended to ensure that the architectures developed by the geographic and functional unified commands, military services, and defense agencies are interrelatable between and among the organizations' operational, systems, and technical architecture views, and are comparable and capable of being integrated across joint and combined organizational boundaries. These efforts should be viewed in the light that government legislation is placing more emphasis on the need to pursue interoperable, integrated, and cost-effective business practices and capabilities within each organization and across the domains of defense. Although not all experts agree that the DoDAF is applicable without changes in the net-centric and net-enabled domain, the application in this new operational environment makes sense to many developers. In order to deal with these requirements, the artifacts of DoDAF are grouped into three categories describing the operational view of the architecture, the systems view of the architecture, and the technical view of the architecture. Some general products do not fall into these categories but are applicable to all views. The three categories are defined as follows:

- *The* ***operational architecture view*** *is a description of the tasks and activities, operational elements, and information flows required to accomplish or support a military operation.* In other words, the products describing the operational view capture the user requirements. Within this view, *what* has to be done is described in an operational context, i.e., what functionality is needed by the Warfighter! These artifacts are primarily focused on the composability levels.
- *The* ***systems architecture view*** *is a description, including graphics, of systems and interconnections providing for, or supporting, war-fighting functions.* The products being used in the systems view describe the functionality of the systems—which already can be in place or under development—in stand-alone operations as well as in the integrated context. In other words, the *how* is defined in terms of available functionality provided by the network-centric system of systems. These products are applicable to cope with the implementation, which is the interoperability level.
- *The* ***technical architecture view*** *is the minimal set of rules governing the arrangement, interaction, and interdependence of system parts or elements, whose purpose is to ensure that a conformant system satisfies a specified set of requirements.* The products of the technical view are the home for standards to be used to enable the system of systems delivering the functionality described in the systems view. The products are focused on integratability of implementations.

Although this paper will focus on the composability levels and applicable DoDAF artifacts, Table 1 gives an overview of all artifacts describing architectures in a DoDAF-compliant way. It should be pointed out that the tool/UML support is not standardized yet and alternatives are possible. UML artifacts are captured in italic in the table.

Not all of these views are mandatory. The views that must be supported by systems documentation to be DoDAF compliant are summarized in the Command, Control, Communications, Computers, and Intelligence (C4I) Support Plan (C4ISP) used for procurement. The C4ISP describes and evaluates the information, infrastructure, and other interfaces required, and the underlying analysis focuses on identifying derived information support requirements and ensuring each requirement is satisfied to meet a given capability need within the system's operational environment. The C4ISP targets the composability level of the LCIM. DoD policy specifies the following architecture products be included in the C4ISP: OV-1, OV-2, OV-3, OV-6c, SV-1, SV-6, and TV-1. Furthermore, DoDAF defines the term *integrated architecture*. An architecture description is defined to be an integrated architecture when artifacts and products and their constituent architecture data elements (see next paragraph) are developed such that architecture data elements describing the same aspect are aligned across the different views (such as using the same name, identification tags, etc.). The minimal artifacts to be used to describe an integrated architecture are AV-1, AV-2, OV-2, OV-3, OV-5, SV-1, and TV-1. Although C4ISP and integrated architecture artifacts do not overlap, the common focus on the operational views show their importance for dealing with concepts.

The last important aspect of DoDAF the authors want to address in this paper is the All-DoD Core Architecture Data Model (CADM), which defines the entities and relationships for architecture data elements. In other words, all artifacts of DoDAF can be captured by the CADM, allowing the definition of repositories. The DoD Architecture Repository System (DARS) is a common repository for storing and retrieving architecture data and automated tools, enabling the users of DoDAF to benefit from alternative earlier or parallel developments. While the artifacts themselves are not necessarily standardized and may look quite different based on the used tool, the data describing these artifacts stored in CADM are standardized. CADM is often confused with tactical data models capturing and structuring the information that has to be exchanged on the battlefield with other participating systems, such as C2IEDM. However, the role of CADM is to enable storing and exchanging information about the architecture of the systems, not operational data to be exchanged within these systems. In this sense, the tactical data models such as C2IEDM are "just a technical view" in the CADM describing what data model to use to implement the information flow specified; e.g., in SV-6, System Data-Exchange Matrix. In other words, currently CADM and C2IEDM are both applicable at the same time and are not alternatives. However, if in the future—as envisioned by Alberts and Hayes [24]—the tactical information to be exchanged via data models such as C2IEDM describes operationally necessary architectures, the required CADM data must be reflected by these information exchange models. Furthermore, if the systems described in the system view captured by CADM are identical with those on the battlefield, the C2IEDM elements describing the systems on the battlefield must map to the CADM elements describing the systems to be supported by the architecture. In other words, the net-centric and net-enabling applications mandate that

Table 1. DoDAF artifacts and tool/UML support

Applicable View	Framework Product	DoDAF Artifact Name	Description	Tool/UML Support
All Views	AV-1	Overview and Summary Information	Scope, purpose, intended users, environment depicted, analytical findings	Text documents
All Views	AV-2	Integrated Dictionary	Data repository with definitions of all terms used in all products	UML model queries and report generators
Operational	OV-1	High-Level Operational Concept Graphic	High-level graphical/ textual description of operational concept	*Class diagrams and use case diagrams*
Operational	OV-2	Operational Node Connectivity Description	Operational nodes, operational activities performed at each node, connectivity and information exchange need lines between nodes	*Composite structure diagrams*
Operational	OV-3	Operational Information Exchange Matrix	Information exchanged between nodes and the relevant attributes of that exchange	UML model queries and report generators
Operational	OV-4	Organizational Relationships Chart	Organizational, role, or other relationships among organizations	*Class diagram*
Operational	OV-5	Operational Activity Model	Operational activities, relationships among activities, inputs and outputs	*Use cases diagrams, activity sequence diagrams, and data-flow diagrams*
Operational	OV-6a	Operational Rules Model	Identifying business rules that constrain operation	Text documents linked to activities
Operational	OV-6b	Operational State Transition Description	Identifying business process responses to events	*State machine diagrams*
Operational	OV-6c	Operational Event-Trace Description	Tracing actions in a scenario or sequence of events and specifying timing of events	*Time sequence diagrams*
Operational	OV-7	Logical Data Model	Documentation of the data requirements and structural business process rules	*Class diagrams and use cases*
Systems	SV-1	Systems Interface Description	Identification of systems and system components and their interconnections within and between nodes	*Composite structure diagrams*
Systems	SV-2	Systems Communications Description	Systems nodes and their related communications lay-downs	*Composite structure diagrams*
Systems	SV-3	Systems-Systems Matrix	Relationships among systems in a given architecture	Traceability views, such as those used in the tool DOORS
Systems	SV-4	Systems Functionality Description	Functions performed by systems and the information flow among system functions	*Activity diagrams with object flows*
Systems	SV-5	Operational Activity to Systems Function Traceability Matrix	Mapping of systems functions to operational capabilities	Traceability views, such as those used in the tool DOORS
Systems	SV-6	Systems Data Exchange Matrix	Provides details of systems data being exchanged between the systems	UML model queries and report generator
Systems	SV-7	Systems Performance Parameters Matrix	Performance characteristics of hardware and software elements	UML model queries and report generator
Systems	SV-8	Systems Evolution Description	Planned incremental steps toward migration or evolution	Project planning documents

Table 1 (continued). DoDAF artifacts and tool/UML support

Applicable View	Framework Product	DoDAF Artifact Name	Description	Tool/UML Support
Systems	SV-9	Systems Technology Forecast	Emerging technologies that are expected to be available in a given timeframe affecting future development of the architecture	Text documents
Systems	SV-10a	Systems Rules Model	Constraints that are imposed on systems functionality due to some aspect of systems design or implementation	Text document linked to system functions
Systems	SV-10b	Systems State Transition Description	Responses of a system to events	*State machine diagrams*
Systems	SV-10c	Systems Event-Trace Description	System-specific refinements of critical sequences of events and the timing of these events	*Sequence diagrams*
Systems	SV-11	Physical Schema	Physical implementation of the information of the logical data model	*Class diagrams*
Technical	TV-1	Technical Standards Profile	Extraction of standards that apply to the given architecture	Text documents linked to system documents
Technical	TV-2	Technical Standards Forecast	Description of emerging standards applicable to the given architecture within an appropriate timeframe	Text document linked to system documents

tactical information exchange data models must be able to reflect system engineering concepts and vice versa. To enable the support of system integration evaluation with operational systems in the mid-term, data models such as C2IEDM and CADM must merge.

3.3 M&S Applications of DoDAF

A complete overview of M&S applications of the DoDAF goes beyond the scope of this paper. However, there are several papers—in particular in user-oriented workshops such as the Simulation Interoperability Workshops (SIW) of the Simulation Interoperability Standards Organization (SISO) or the Command and Control Research and Technology Symposia (CCRTS) of DoD's Command and Control Research Program (CCRP)—dealing with this issue. The interested reader is referred to, in particular, an overview paper presented by Atkinson [25] dealing with the use of M&S to cope with capability-based planning during the Spring SIW. The C4I view is addressed by Couture and Duval [26]. Both papers show the applicability of DoDAF artifacts embedded into UML descriptions to enable the use of M&S for capability-based planning in the procurement as well as in the operational sense. Tolk and Solick [27] introduce the idea to use DoDAF artifacts to facilitate verification and validation, as the user requirements are captured in a standardized way.

Furthermore, several efforts have been focused on applying engineering results from the control systems community developed to enable component-based design and assembly of complex systems. In this area there is a long history of composing components as design and implementation moves from software-only models to hardware-in-the-loop models to fielded implementations with sensors and actuators interacting with distributed physical and logical components, which overlaps with many challenges faced by current distributed M&S developers. Results on U.S. Army–sponsored research regarding the applicability of DoDAF in this domain are published by James [28]; the author discusses the use of the different views of the DoDAF to support partitioning of integrated architecture into components, construction of a view of information assurance processes, and subsequent online analysis to enable intrusion detection.

The idea to use the platform-independent models (PIM) of MDA to manage heterogeneous solutions for M&S application has been introduced several times, e.g., in Tolk [29]. To enable bridging the gaps between the conceptual ideas, their implementation, and ultimately their integration into a larger system-of-systems (e.g., as "functionality-as-needed" providing services in a net-centric environment) M&S–specific standards are necessary. Tolk and Muguira [30] evaluated the parallel application of MDA, HLA, and the Discrete Event System Specification (DEVS) formalism, but this publication lacks a rigorous methodology. DeSylva et al. [31] address the overlap of DoDAF artifacts and the Federation Development and Execution Process (FEDEP) recommended for the HLA, but the rigorous evaluation of DEVS has not been done before. This gap is currently closed by ongoing work on recommended DoDAF extensions based on DEVS. Zeigler et al. [32] and Zeigler and Mittal [33] apply DoDAF in the *bifurcated model-continuity–based development process*, which allows the definition of references master models in parallel to but aligned with experimental frames, which finally lead to the operational implementations in net-centric environments. They also show additional improvements in comparison with solutions exclusively based on UML or MDA. The work presented in the following section completes these valuable efforts of Zeigler and his team on the implementation-oriented middle levels with operational aspects on the composability-oriented upper levels.

4. Applying the Recommended Framework

This section of the paper describes the application of a framework that will aid in the identification of challenging areas of federation of a system-of-systems. The objective of this framework is to support military commanders as they explore alternative means to achieve positive results in a multi-resolution operation like Iraqi Freedom, which comprises traditional theatre-wide warfare but includes various "hot spots" in the form of relevant urban operations. Derivatives of this framework have been used by VMASC to support the U.S. Joint Forces Command (JFCOM) in experimentation and the analysis from the framework will be used to create tools to support NATO as explained in section 4.3.

The M&S community further supports the military with multi-resolution modeling (MRM) as in Davis [34], which points out that resolution refers to the manner in which time is handled, the manner in which entities are governed, and the manner in which entity movements are considered. Interoperation of models of different resolutions is fraught with problems. Petty and colleagues [35–37] have also studied the multi-resolution simulation problem. This set of papers deals with the specific problems of aggregation and de-aggregation within simulation systems. The authors highlight the main problems of information loss when considering aggregated forces and information overload as a result of close quarter interaction of entities with in the simulated battlefield.

Consider the mix of forces in an operation the size of Iraqi Freedom. Analyzing the battle plans from an operational point of view, one is going to have ground forces and air units that will need to interoperate in close cooperation with each other. Operational analysis shows that ground forces and air units will be engaged in theatre-wide warfare, while urban warfare will be supported by air units, but mainly conducted by ground forces. These different roles played by the force mix translate directly into the responsibilities of the underlying simulation infrastructure. To show the applicability of the DoDAF in connection with capturing information, such as the different roles and the impacts of actions within a role, we evaluate the solution proposed by Bowers et al. [38] to generate a specification.

Evaluation of this specification reveals that two models of different resolution are needed to adequately provide for all of the capabilities that operational analysis demands. In the Bowers et al. paper the situation unfolds in and around the city of Khafji as theater level combat is happening north of and outside the city while tactical combat is also present in the city. The purpose of the Bowers et al. study is to create an experimental system that could facilitate MRM simulation and study. The authors' models of choice were JTLS for theater-wide operations and JCATS [38] for urban operations. The federation of these models magnifies the challenge of providing continuous support to the war fighter, where *continuous,* in this case, means addressing all required capabilities.

4.1 Analysis of the Framework

The framework builds upon the information collected during the application of the DoDAF to specify the deviance of the model from the intent. Hawkins [41] and then Tolk [42] have analyzed these deviances, and Hawkins has shown that the cause of the deviance is internal structural problems with the model. The framework creates a catalog of these "structural variance" instances, which can later be used to understand the discontinuous nature of the implemented system.

These instances of structural variances spring directly from the consistent nature of the DoDAF analysis and the layering of each view from abstract idea down through specification into implementation. Each layer of the DoDAF—operational, systems, and technical—builds up or adds to information in the previous layer. Starting analysis with the operational aspects captures the required capabilities, functions, information flows, and event generation independent of the systems or technology applied during implementation. The systems view of the DoDAF further specifies what the system will be capable of by assignment of concrete models to handle specific instances of functionality. Ongoing research by Muguira reveals that one possible source of structural variances comes from the confluence of required capability at both operational and systems layers. The DoDAF version one provides an audit of the transition from operational thinking to systems level thinking. The DoDAF does cover the other two important areas of variance between federated models: information flow coverage between federated models and event generation between federated models. But unlike the mapping of operational to systems level functional specification, the DoDAF does not directly provide for an audit of information flow or event generation. The methodology applied here closes that gap.

The DoDAF leaves the instrumentation of the specification up to interpretation so analysts can apply notation that suits the situation (see Table 1). At each stage the analyst applying the DoDAF is trying to capture a scripted set of information artifacts. The content of any one artifact is guided by two things: the artifact classification (operational, systems, or technical) and the artifact information category (e.g., there are six operational views). Since UML is understood and applied by a large number of M&S practitioners as well as programmers, utilization of UML for DoDAF notation makes each artifact interpretable by a large audience. UML does have some deficiencies, and for the purpose of this framework a little guidance is needed in utilization as will be demonstrated later in the paper.

Four core ideas need to be captured from the DoDAF operational analysis: operational entities or nodes and the functions required of each node, what set of activities each node will participate in, what roles can each node play, and a specification of all the information exchanges possible by each node. Inconsistencies in or between any of the four core operational ideas generate structural variances. The systems views amplify the information captured in the operational study, and the DoDAF technical views add volume to the analysis. Muguira's dissertation enumerates additional sources for structural variances. This framework's method creates an additional constrained audit of the DoDAF analysis process that evaluates the transition from operational to systems views for consistency. Later in this paper an example is presented to demonstrate the framework method for the mismatch of functionality aspect.

A literature review and analysis showed that three critical interoperation domains exist: behavior, time, and scope, which break down to information exchange transparency or resolution.

- The need to align the *behavior* of simulation components to the intended use of the system is pointed out by Sutton [43] as well as by Wartik et al. [44].
- Davis and Biglow [34] highlight the problems of differing *resolution* in linking together multiple simulation systems as components.
- Fujimoto [45] defines three different types of *time* that are dealt with in simulation: physical time, or the time base of the physical system that is being modeled; simulation time, or the time base of the simulator; and, finally, wall clock time, or the time during which the simulation is being executed.
- Levytskyy et al. [46] showcase the need for a uniform and generic *transparency* or exchange of information between simulation systems.

The framework motivates the idea of describing the structural variances from the composition of the three interoperation domains with the top three levels of the LCIM. As motivated before, DoDAF is mainly contributing to the pragmatic, dynamic, and conceptual level of the LCIM. The three targeted levels of interoperability and the three interoperation domains define a matrix to organize the categories of structural variances in form of a Structural Variance Matrix (SVM). Using the matrix, structural variances can be broken down and studied by interoperation domain and/or LCIM level.

While experienced federation developers may be able to immediately identify critical areas, novices need support based on engineering methods. This framework will apply the DoDAF and first gather operational information about the capabilities and required functionality needed in the federation then fill in systems and finally technical information describing the specified system. The framework method can be stated as a recipe for identifying structural variances:

1) Use combinations of UML diagrams to capture operational information (see the following paragraphs).
2) Use combinations of UML diagrams to capture systems information (see below).
3) Compare systems data flows to operational information flows:
   a. Look for gaps in coverage;
   b. Look for data dimensionality problems that are inconsistent when federated;
   c. Look for data handling functionality in multiple federates and check to see if these implementations are consistent.
4) If the system under study can report generated events, then compare the systems and operational views:
   a. Look for systems coverage of all operational views;
   b. Look for consistent handling of events by the federated nodes;
   c. Look for redundant inconsistent event processing.
5) If one of the criteria is met generate a SVM entry:

a. Classify the structural variances in terms of interoperation domain;
b. Categorize the structural variances in terms of LCIM level;
c. Capture a short one paragraph description in terms of interoperation domain and LCIM level;
d. Draw UML;
e. Create a short one-line description for SVM.

Inspection of the DoDAF Deskset reference reveals that there is an order in which the artifacts from the DoDAF can be completed. The analyst first completes the all-view, which describes the projects goals. Starting with the operational view, we are concerned with what operational entities or nodes are active in the federation. The DoDAF refers to the nodes diagram as Operational View 2 (OV-2). Most of the DoDAF artifacts amalgamate a number of ideas on each diagram. OV-2 demonstrates the basic entities or nodes active in the system and the basic lines of communication between the nodes. To capture these ideas, the UML provides a class diagram to describe the operational nodes and the functionality of the each node. Paired with a UML collaboration diagram the UML class diagram can capture both node and connectivity aspects. UML comments are used liberally to capture additional information. Once the nodes have been identified, further information is gathered to specify what activities these nodes are engaged in (DoDAF diagram OV-5) and the information that is flowing between them (DoDAF diagram OV-3). UML collaboration diagrams can be used since many times what are needed are both the nodes and some simple explanation of relationship or general information flow. UML sequence diagrams are also used to show in a more detailed manner the sequencing of communication and communications partners. In practice, the analyst will combine UML diagrams that together capture the required ideas found in the operational views, thus OV-2 could be a class and collaboration diagram, while OV-3 could be a collaboration and sequence diagram, and OV-5 is an activity and collaboration and possibly a sequence diagram. Once the analyst has identified the operational nodes, their activity and the information flowing between them the analyst will consider which operational activities are implemented by more than one of the supporting nodes. These activity overlaps are a good place to study the system for variance that will cause discontinuities in the coverage of intent by the implemented system. The last of the operational views the analyst may deploy are the DoDAF operational rules (business logic) description, the state transition description and the event trace (OV-6a, OV-6b, OV-6c) description. As indicated in Table 1, OV-6a is a text document, while OV-6b is a state transition diagram, and OV-6c is a UML sequence diagram.

Armed with the operational picture and activity overlaps the analyst completes the systems views of the DoDAF analysis. The DoDAF systems views focus on specific systems that contribute functionality and cope with required capabilities. The analyst, in completing the DoDAF operational analysis, focuses on gathering information about the operational entities or nodes, activities, their roles, and information flows between them. The DoDAF systems views add additional information to the four operational views just gathered by assigning specific systems to the nodes and then considering the impact and implications of the choice.

The information depicted in DoDAF Systems View 5 (SV-5) provides a mapping between what capabilities are desired and what is finally implemented. The need is to show capabilities and how they have been handled. This same audit can be performed on information exchanges and event exchanges. The DoDAF provides analysis information about the systems versus operational rules governing the system, the relative transparency of information exchanges and the handling of event information. First, systems rules are compared with operational rules, then information exchanges are compared, and finally, event exchanges. DoDAF SV-6 describes the system level exchange of data. Comparing SV-6 with OV-3 provides a mapping of operational to systems information flow. DoDAF SV-10a, SV-10b, and SV-10c round out the systems views with views corresponding to the operational rules, capture of state, and event trace information. Comparing the systems level events that are generated with the operational events called for highlights where the framework and methodology of this paper locate structural variances.

4.2 An Application Example

Though the DoDAF provides more informational views that combine to become the specification of a system, and more could be said about the views that have been described, the goal is to present the framework. Bowers et al. [38] identified nine critical cases, which are supported by our analyses.

- Ground forces moving into the play box
- Ground forces moving out of the play box
- Air units moving into play box
- Air unit moving out of play box
- Area munitions
- Ground-to-air
- Air-to-air
- Air-to-ground
- Ground-to-ground

To demonstrate the application of the framework, the authors demonstrate the application for the field of *area munitions*. A critical junction in this case is when entities in the theatre level have moved to a point where they interact with entities of the tactical level. In the case of area munitions from Bowers et al. [38], artillery or bombs are exchanged. Figure 2 and Figure 3 document the DoDAF OV-2. The DoDAF calls for node and relationship information to be captured in OV-2. In practice, this takes multiple UML diagrams to capture the required information.

These diagrams document the capability to exchange munitions in either direction. Each node has the capability to compute attrition, and report status and resolution dependent position. The systems view must describe how each of the capabilities is handled.

The next figure, Figure 4, presents the DoDAF OV-3 information flow diagram. This diagram shows the two general message types that are exchanged between nodes in the federated system: adjudicate, which refers to the processes of adjusting force levels after combat, and shoot, which refers to an individual systems ability to inflict damage on the opposing forces. If there were structural variances, the systems-level diagrams for information flow (called data exchange and labeled SV-6) corresponding to the operational information flow would demonstrate gaps in coverage or other problems related to federating systems.

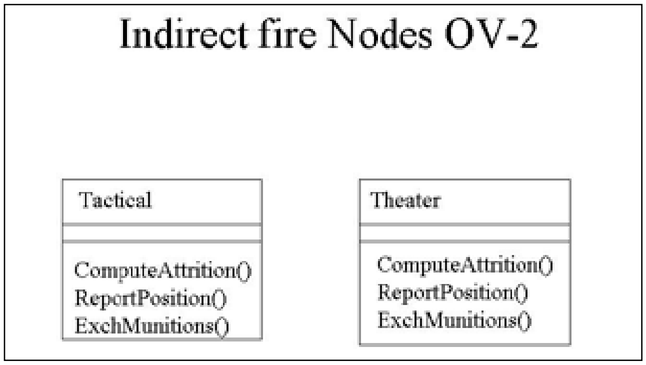


Figure 2. Nodes and functions

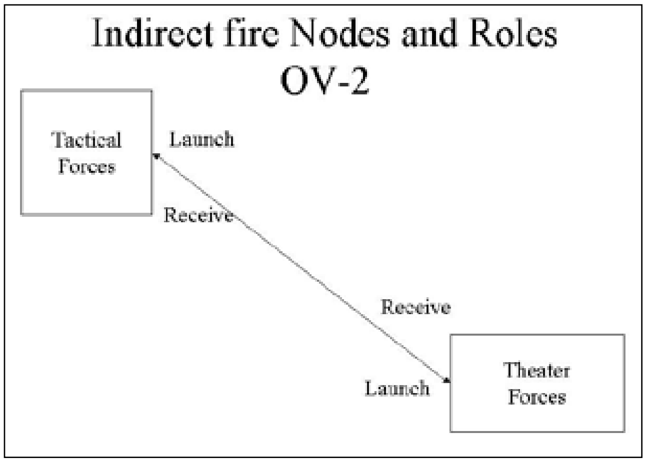
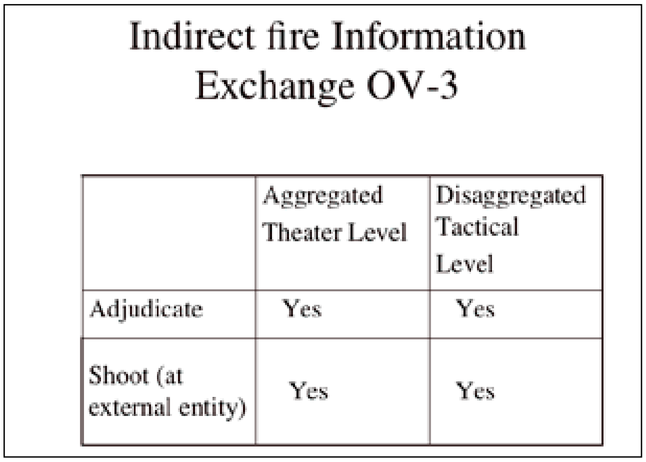


Figure 3. Roles and basic communications

The next diagram presented is the DoDAF data exchange diagram SV-10a. This diagram would be compared to the purpose stated in the AV-1 view, but simple inspection reveals that the different systems components use different combat models.

The next diagram presented is SV-6. Comparing this diagram with OV-3 reveals a number of problems at the systems information exchange level.

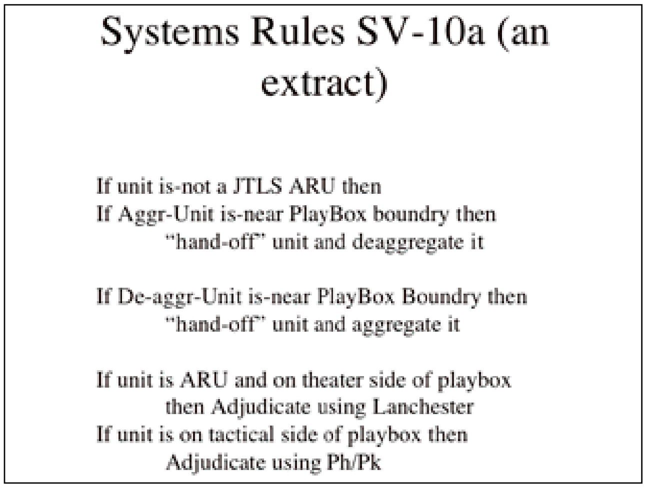
The most noticeable problem is that all data exchanged between the components will refer to aggregated information. The final structural variances information that can be extracted from the set of DoDAF artifacts comes from the SV-4 system functionality description SV-4. Summarized here as comments, SV-4 reveals that the two components



Indirect fire Information Exchange OV-3

	Aggregated Theater Level	Disaggregated Tactical Level
Adjudicate	Yes	Yes
Shoot (at external entity)	Yes	Yes

Figure 4. Information flow



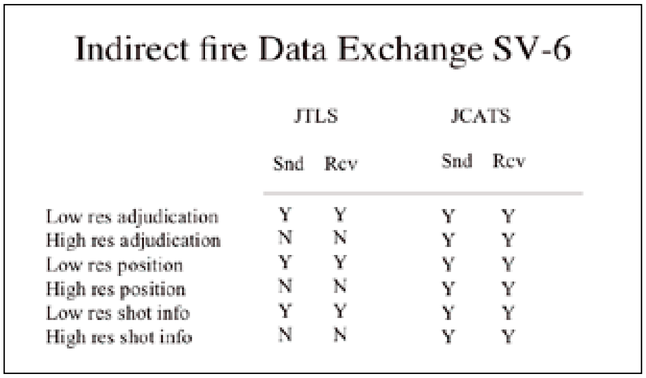
Systems Rules SV-10a (an extract)

If unit is-not a JTLS ARU then
If Aggr-Unit is-near PlayBox boundry then
"hand-off" unit and deaggregate it

If De-aggr-Unit is-near PlayBox Boundry then
"hand-off" unit and aggregate it

If unit is ARU and on theater side of playbox
then Adjudicate using Lanchester
If unit is on tactical side of playbox then
Adjudicate using Ph/Pk

Figure 5. System rules



Indirect fire Data Exchange SV-6

	JTLS		JCATS	
	Snd	Rcv	Snd	Rcv
Low res adjudication	Y	Y	Y	Y
High res adjudication	N	N	Y	Y
Low res position	Y	Y	Y	Y
High res position	N	N	Y	Y
Low res shot info	Y	Y	Y	Y
High res shot info	N	N	Y	Y

Figure 6. Data exchange diagram

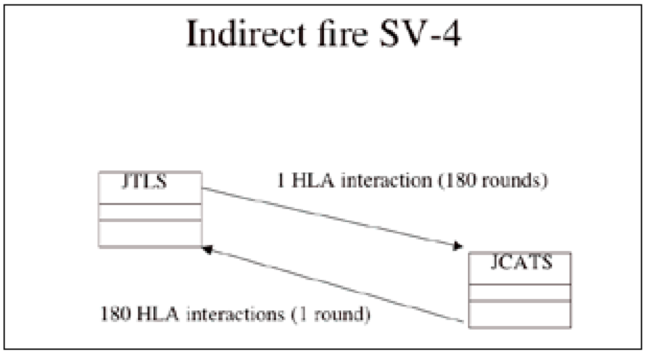
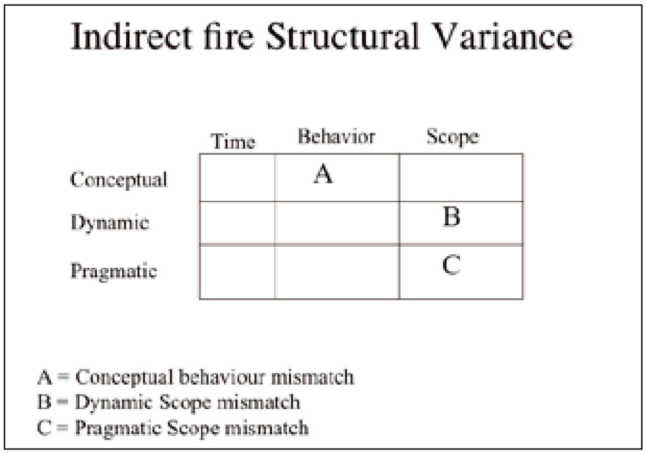


Figure 7. Systems functionality description



Indirect fire Structural Variance

	Time	Behavior	Scope
Conceptual		A	
Dynamic			B
Pragmatic			C

A = Conceptual behaviour mismatch
B = Dynamic Scope mismatch
C = Pragmatic Scope mismatch

Figure 8. SVM for area munitions

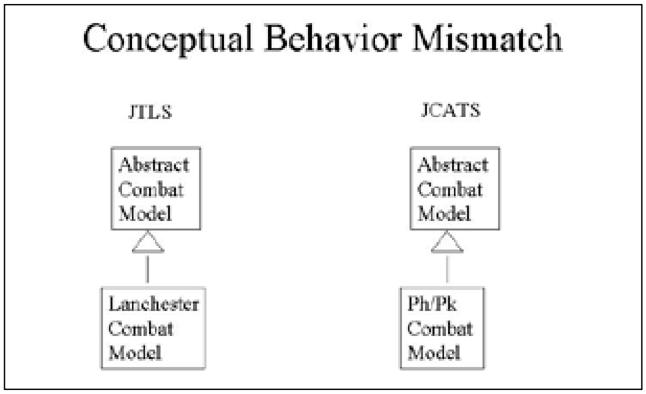


Figure 9. Behavior mismatch

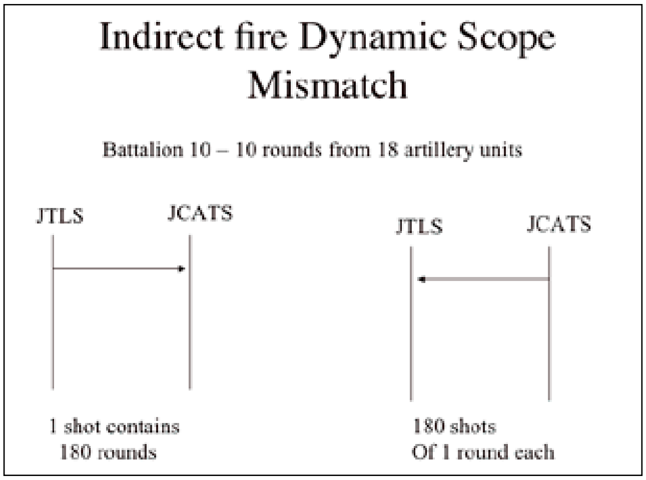


Figure 10. Only artillery and bombs are exchanged

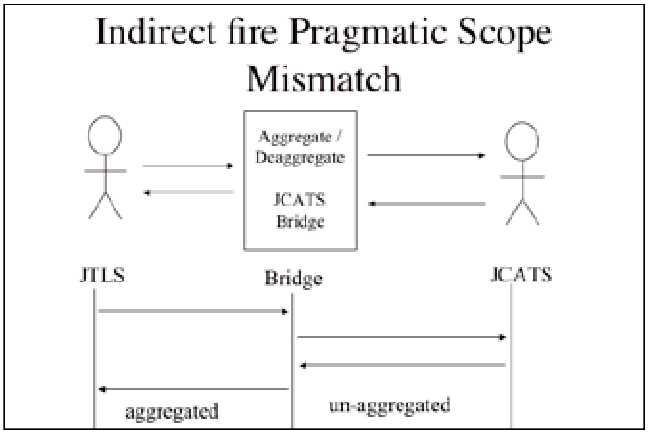


Figure 11. Exchange information and comments

do not use the same methods to adjudicate conflict. The theatre-level component, JTLS, utilizes only one low-level communications message to initiate conflict while the tactical level JCATS utilizes a number of low-level communications messages to implement the same conflict activity.

In the case of area munitions, the analysis gathered a number of structural variances just from considering the operational and systems views. Other cases may require evaluation of more of the possible artifacts produced by a DoDAF analysis. Representing the three structural variances in one matrix format is Figure 8.

To provide clarity and consistency the three structural variance items are further depicted in UML. Figure 9 demonstrates a UML class diagram representing the behavior mismatch of different combat model implementations. Figure 10 demonstrates a UML collaboration diagram that represents the mismatch in the exchange of munitions. Figure 11 documents the pragmatic mismatch in resolution.

To complete the methodology the analyst will work through more DoDAF artifacts toward the completion of SV-10c, which describes the events that can be exchanged throughout the system-of-systems. Comparing the operational events of OV-6c and the systems events of SV-10c could reveal more coverage gaps or inconsistently implemented federated handling of event information.

After the prototype was completed, Bowers and Prochnow [47] enhanced their system to resolve problems and add functionality. The three specific structural variance entries found by the methodology in this paper were addressed in the following manner:

1) Conceptual behavior difference
    - Different combat models – no fix at this time
2) Transparency mismatch
    - Changed JTLS to group incoming volleys of shots
3) Pragmatic resolution mismatch
    - No change

In summary, the application of method described in the first section delivered the results as observed in the papers of Bowers and colleagues [38, 47] and explain the observed structural variances. Applying the framework in the early stages of a project can avoid the composition of approaches that are not well aligned on the conceptual level.

4.3 Implementation and Integration Levels

Once the operational and systems aspects of the modeled system-of-systems have been captured and the conceptual alignment is ensured, the analyst may turn toward the implementation of executable systems to study various configurations of the model. To map this resulting ensemble of captured information onto an executable framework, such as represented by DEVS, the DoDAF artifacts must be completed and additional methods must be applied, such as those introduced in Zeigler and Mittal [33]. In this paper, Zeigler and Mittal provide a mapping between the various artifacts of the DoDAF analysis and DEVS:

Following the standard notation as introduced in [1], a mapping between DoDAF and DEVS can be done as follows:

X = inputs (from OV-3 and OV-5)

S = states (from OV-6b and OV-6c)

Y = outputs (OV-3 and OV-7)

Ti = internal transition function (from OV-6a/b/c, SV-10a/b/c)

Te = external transition function (from OV-6a/b/c, SV-10a/b/c)

Fo = output function (from OV-6a/b/c, SV-10a/b/c)

Ta = time advance function (from OV-6a/b/c, SV-10a/b/c)

Additional ideas on the applicability of DoDAF as the core for executable architectures are published in Dryer and Berbesi [48]. Although DoDAF artifacts are not sufficient to enable executable architectures, the DoDAF's guidance in creating a specification structures the necessary operational concepts in a mature way. The application of the framework based on the DoDAF is necessary to ensure meaningful composability on the conceptual level without which the technical interoperability on the implementation level loses its applicability. More research is necessary to close the gaps.

5. Summary and Conclusion

The artifacts of DoDAF support integrated system evaluation. The two application examples in this paper show their use within a framework to support the evaluation of M&S applications in a given operational context and their use to define metadata for reuse and composition. Efforts as described by Tolk and Solick, [27] or Zeigler et al. [32] show more applications. Although DoDAF is not an exclusive solution, the artifacts support a core for further analyses. The use of UML diagrams to capture the DoDAF ideas broadens the applicability and is an advantage when dealing with organizations outside the DoD, such as in Homeland Security operations based on inter-agency operations.

DoDAF's strength lies in the operational domain. Although the systems view already enables good analysis of potential variances in the model continuity, evaluations as published in Dryer and Berbesi [48] and James [28] show the need for extensions when applied in the domain of executable architectures complex systems. The DoDAF-independent LCIM is a simple but powerful model to cope with the interoperation challenges in the areas of *composability* (the modeling level), *interoperability* (the simulation and software level), and *integratability* (the technical and networking level). The work of Zeigler and Mittal [36] and Tolk and Muguira [30] show that the integration of additional DEVS artifacts can be used to improve the applicability of DoDAF for M&S and increase DoDAF's support to facilitate interoperation in the net-centric and net-enabled domain. Methodology frameworks as published by Trbovich and Reading [19] show the potential for merging so-far independently developed methods. The framework proposed in this paper shows the applicability of software engineering methods supporting integrated system evaluation. If operational future applications as envisioned by Alberts and Hayes [24], which currently is technically prepared by first steps such as the net-centric data strategy [14], will be supported by IT in the future, the operational use of M&S functionality over the complete lifecycle must become an integral part of such systems. Standardization committees are already starting to discuss how to apply DoDAF for service-oriented architectures [49] as envisioned for M&S application in the GIG [50].

6. References

[1] Zeigler B. Theory of Modeling and Simulation. John Wiley and Sons; 1976.

[2] Office of the Assistant Secretary of Defense (Networks & Information Integration, NII). Department of Defense Architecture Framework (DODAF); 2003. Available from: http://www.defenselink.mil/nii

[3] Tolk A, Muguira JA. The Levels of Conceptual Interoperability Model (LCIM). In: Proceedings of the Fall Simulation Interoperability Workshop; IEEE CS Press; 2003.

[4] Levels of Information Systems Interoperability (LISI); 1998 Mar 30; C4ISR Architectures Working Group. Available from: http://www.c3i.osd.mil/org/cio/i3/

[5] NATO Allied Data Publication 34 (ADatP-34); Mar 2003; NATO C3 Technical Architecture (NC3TA), Version 4.0. Available from: http://www.nato.ino/docu/standard.htm

[6] Turnitsa C. Extending the Levels of Conceptual Interoperability Model. In: Proceedings of the IEEE Summer Computer Simulation Conference; IEEE CS Press; 2005.

[7] Page EH, Briggs R, Tufarolo JA. Toward a Family of Maturity Models for the Simulation Interconnection Problem. In: Proceedings of the IEEE Spring Simulation Interoperability Workshop; IEEE CS Press; 2004.

[8] Tolk A. XML Mediation Services Utilizing Model Based Data Management. In: Proceedings of the IEEE Winter Simulation Conference; 1476–1484, IEEE CS Press; 2004.

[9] Tolk A, Diallo SY. Model Based Data Engineering for Web Services. IEEE Internet Computing. 2005; 9(4): 65–70.

[10] Department of Defense Chief Information Office (CIO). Department of Defense Net-Centric Data Strategy; 2003. Available from: www.afei.org/pdf/ncow/DoD_data_strategy.pdf

[11] Multilateral Interoperability Programme Website: http://www.mip-site.org

[12] U.S. Army Modeling & Simulation Executive Council (AMSEC). Minutes of Meeting; 2005 Jul 11; Item 2. Available from: http://www.amso.army.mil

[13] Hofmann M. Challenges of Model Interoperation in Military Simulations. SIMULATION. 2004; 80: 659–667.

[14] NATO Research and Technology Organization (RTO). The NATO Code of Best Practice for Command and Control Assessment; Revision 2002. Command and Control Research Program; NATO Support Series (ISBN 1-893723-09-7). Available from: http://www.dodccrp.org

[15] Petty MD, Weisel EW. A Composability Lexicon. In: Proceedings of the Spring Simulation Interoperability Workshop, IEEE CS Press; 2003.

[16] Object Model Group (OMG) Unified Modeling Language (UML). Available from: http://www.omg.org/uml

[17] Kobryn C, Sibbald C.Modeling DoDAF Compliant Architectures: A Telelogic Approach for Complying with the DoD Architecture Framework; 2004. Available from: http://www.telelogic.com/download/paper/Modeling-DoDAF-WhitePaper.pdf

[18] Object Model Group (OMG) Unified Modeling Language (UML). Available from: http://www.omg.org/mda

[19] Trbovich S, Reading R. Simulation and Software Development for Capabilities Based Warfare: An Analysis of Harmonized Systems Engineering Processes. In: Proceedings of the IEEE Spring Simulation Interoperability Workshop; IEEE CS Press; 2005.

[20] Blake DW, Little C, Morse J. The Navy's Probability of Raid Annihilation Assessment Process Standards & Architecture and Systems Engineering Concept Model. In: Proceedings of the IEEE Fall Simulation Interoperability Workshop; IEEE CS Press; 2003.

[21] Electronic Industries Alliance (EIA). EIA-632: Processes for Engineering a System; 1998.

[22] Lee J, Choi M, Jang J, Park Y, Jang J, Ko B. The Integrated Executable Architecture Model Development by Congruent Process, Method, and Tools. In: Proceedings of the Command and Control Research and Technology Symposium; Paper 259; CCRP Press; 2005.

[23] Adshead S, Kreitmair T, Tolk A. Definition of ALTBMD Architectures by Applying the C4ISR Architecture Framework. In: Proceedings of the IEEE Fall Simulation Interoperability Workshop; IEEE CS Press; 2001.

[24] Alberts DS, Hayes RE. Power to the Edge: Command and Control in the Information Age. Command and Control Research Program; CCRP Press; 2003.

[25] Atkinson K. Modeling and Simulation Foundation for Capabilities Based Planning. In: Proceedings of the IEEE Spring Simulation Interoperability Workshop; IEEE CS Press; 2004.

[26] Couture M, Duval A. On the Building of a UML Profile for the Description of Army Architectures in the Context of Complex Systems. In: Proceedings of the Command and Control Research and Technology Symposium; Paper 124; CCRP Press; 2005.

[27] Tolk A, Solick S. Using the C4ISR Architecture Framework as a Tool to Facilitate V&V for Simulation Systems Within the Military Application Domain. In: Proceedings of the IEEE Spring Simulation Interoperability Workshop; IEEE CS Press; 2003.

[28] James J. Modeling of Information Dominance in Complex Systems: A System Partitioning and Hybrid Control Framework. In: Proceedings of the Hawaii International Conference on System Science; 2003.

[29] Tolk A. Avoiding Another Green Elephant – A Proposal for the Next Generation HLA Based on the Model Driven Architecture. In: Proceedings of the IEEE Fall Simulation Interoperability Workshop; IEEE CS Press; 2002.

[30] Tolk A, Muguira JA. M&S within the Model Driven Architecture. In: Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC); Paper 1477; 2004.

[31] deSylva MJ, Lutz RR, Osborne SR. The Application of the DoDAF Within the HLA Federation Development Process. In: Proceedings of the IEEE Fall Simulation Interoperability Workshop; IEEE CS Press; 2004.

[32] Zeigler BP, Fulton D, Hammonds P, Nutaro J. Framework for M&S-Based System Development and Testing in Net-Centric Environment. To be published in ITEA Journal (Nov 2005).

[33] Zeigler BP, Mittal S. Enhancing DoDAF with DEVS-Based System Life Cycle Process. In: Proceedings of the IEEE Conference on Systems, Man, and Cybernetics (SMC05); 2005.
[34] Davis PK, Biglow JH. Experiments in Multiresolution Modeling (MRM). RAND Graduate School of Policy Studies; 1998.
[35] Weisel E, Petty M, Mielke R. Computational Complexity for Selecting Components for Composition. In: Proceedings of the IEEE Fall Simulation Interoperability Workshop; IEEE CS Press; 2003.
[36] Weisel E, Petty M. A Composability Lexicon. In: Proceedings of the Spring Simulation Interoperability Workshop; IEEE CS Press; 2003.
[37] Weisel E, Petty M. A Formal Basis for a Theory of Semantic Composability. In: Proceedings of the IEEE Spring Simulation Interoperability Workshop; IEEE CS Press; 2003.
[38] Bowers A, Prochnow DL, Roberts J. JTLS-JCATS: Design of a Multi-Resolution Federation for Multi-Level Training. In: Proceedings of the IEEE Fall Simulation Interoperability; IEEE CS Press; 2002.
[39] Joint Theater Level Simulation (JTLS). Available from: http://www.jfcom.mil/about/fact_jtls.htm
[40] Joint Conflict and Tactical Simulation System. Lawrence Livermore National Laboratory L-184; Livermore, CA. Available from: http://www.llnl.gov/str/Shimanmoto.html
[41] Hawkins G. Structural Variance and Other Related Topics in the SHAPE Armour/Anti-Armour Study. In: Huber RK, editor. Systems Analysis and Modeling in Defense - Development, Trends, and Issues. New York: Plenum Press; 1984.
[42] Tolk A. Non-Monotonicities in HLA-Federations. In: Proceedings of the IEEE Spring Simulation Interoperability Workshop; IEEE CS Press; 1999.
[43] Sutton PW. Interoperability: A New Paradigm. In: Proceedings of the IEEE Spring Simulation Interoperability Workshop; IEEE CS Press; 1999.
[44] Wartik SP, Haugh BA, Loaiza F, Hieb MR. A Methodology for Comparing C4I Data Models and Simulation Object Models. In: Proceedings of the European Simulation Interoperability Workshop; ACM Press; 2001.
[45] Fujimoto RM. Time Management in the High Level Architecture. SIMULATION. 1998; 71(6): 388–400.
[46] Levytskyy A, Kerckhoffs EJH. Integration of Simulation Tools and Models in a Collaborative Environment. In: Proceedings of the European Simulation Interoperability Workshop; ACM Press; 2001.
[47] Bowers A, Prochnow DL. Multi-Resolution Modeling in the JTLS-JCATS Federation. In: Proceedings of the IEEE Fall Simulation Interoperability; IEEE CS Press; 2003.
[48] Dryer D, Berbesi H. Evolving DODAF: An Integrated Training Enterprise - Delivery Architecture Framework. In: Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC); Paper 1695; 2004.
[49] Dandashi F, Ang H-W, Bashioum C. Tailoring DODAF to Support a Service Oriented Architecture. OMG Press; 2004.
[50] Chaum E, Hieb MR, Tolk A. M&S and the Global Information Grid. In: Proceedings of the Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC); Paper 2450; 2005.

Author Biographies

James A. Muguira *is a Systems Architect and Engineer with Systematic Software Engineering (SSE) of Fairfax, VA. He is supporting the Multilateral Interoperability Program (MIP) community by providing enhancements to SSE's MIP framework, SitaWare, to support interoperability at the coalition level. He has several years of software development experience and has conducted various development projects for NASA and other U.S. corporate customers. He is a Ph.D. candidate in Old Dominion University's M&S program and received his MS in applied physics and BS in information science from Christopher Newport University in Newport News, VA.*

Andreas Tolk *is Associate Professor in Engineering Management and Systems Engineering in the College of Engineering and Technology at Old Dominion University (ODU) in Norfolk, Virginia. He has over sixteen years of international experience in the field of Applied Military Operations Research and Modeling and Simulation of and for Command and Control Systems. He is affiliated with the Virginia Modeling Analysis & Simulation Center (VMASC). His domain of expertise is the integration of M&S functionality into real-world applications based on open standards and open architectures. He received his Ph.D. and his MS in computer science from the University of the Federal Armed Forces in Munich, Germany*