Annex F – STANDARDS

F.1 INTRODUCTION

As indicated in Section 3.2.6 "Standards Review and Evaluation" of the accompanying text, the Team invested significant effort reviewing standards potentially relevant to conceptual modeling in hopes of leveraging existing standards and standards types to the greatest extent possible in developing the best-practice guidance contained herein.

An indication of the scope and evaluative interest in such standards is provided in Table F-1 following. Based on an initial review of the standards indicated therein, the Team selected nineteen (19) standards of particular interest to conceptual modeling and for further analysis and commentary.

Table F-1: Standards with Applicability in NATO Modeling and Simulation Domain.

STANDARD	Standard Development Organization	Status	Functional Area	Define Application Objectives	Perform Conceptual Analysis	Design Federation	Design Simulation	Develop Simulation	Develop Federation	Plan, Integrate and Test Application	Execute Application and Prepare Outputs	Analyze Data and Evaluate Results
CMMI	Carnagie Mellon University	Published	Business Process Management	X								X
BOMs	SISO	Published	Conceptual Modeling		X	X	X		X			
CML	OneSAF PMO	Published	Conceptual Modeling	X	X	X	X	X	X			
IDEF0	IEEE	Published	Conceptual Modeling		X							
IDEF5	KBS	Published	Conceptual Modeling		X	X	X	X	X			
OWL-Web Ontology Language	W3C	Published	Conceptual Modeling		X							
Simulation Conceptual Modeling RP	SISO	Concept	Conceptual Modeling		X	X	X					
UML	OMG	Published	Conceptual Modeling		X	X	X	X				
DCMF	Open Source	Published	Data Engineering		X	X	X					
IDEF1X	NIST	Published	Data Engineering		X		X	X				

STANDARD	Standard Development Organization	Status	Functional Area	Define Application Objectives	Perform Conceptual Analysis	Design Federation	Design Simulation	Develop Simulation	Develop Federation	Plan, Integrate and Test Application	Execute Application and Prepare Outputs	Analyze Data and Evaluate Results
OKBC	DARPA	Published	Data Engineering		X	X	X					
RDF	World-Wide-Web Consortium (W3C)	Published	Data Engineering			X	X	X	X			
XML	W3C	Published	Data Engineering					X	X		X	
C-BML	SISO	Draft	Data Mediation		X			X	X	X	X	
CMSD	SISO	Draft	Data Mediation					X	X	X	X	
JC3IEDM	MIP	Published	Data Mediation			X	X	X	X			
OpenFlight	MultiGen-Paradigm	Published	Data Mediation				X	X	X			
SEDRIS	ISO	Published	Data Mediation				X	X	X	X	X	
DFAD	U.S. DoD	Published	Data Production Format				X	X	X	X	X	
DTED	U.S. DoD	Published	Data Production Format				X	X	X	X	X	
DEVS	SISO	Concept	M&S-Miscellaneous				X	X				
SCORM Sim	SISO	Concept	M&S-Miscellaneous			X	X	X	X	X	X	
SRML	SISO	Draft	M&S-Miscellaneous				X	X				
HLA FEDEP	SISO/IEEE	Published	M&S-Process	X	X	X	X	X	X	X	X	X
SEDEP	EUCLID	Published	M&S-Process	X	X	X	X	X	X	X	X	X

STANDARD	Standard Development Organization	Status	Functional Area	Define Application Objectives	Perform Conceptual Analysis	Design Federation	Design Simulation	Develop Simulation	Develop Federation	Plan, Integrate and Test Application	Execute Application and Prepare Outputs	Analyze Data and Evaluate Results
Link 11 Simulations	SISO	Draft	M&S-Representation				X	X	X			
Link 16 Simulations	SISO	Published	M&S-Representation				X	X	X			
MOD-5/S IFF	SISO	Concept	M&S-Representation				X	X	X			
RPR FOM	SISO	Published	M&S-Representation			X	X	X	X			
MSDL	SISO	Draft	M&S-Scenarios		X			X	X	X	X	
CSPI	SISO	Draft	M&S-Simulation Inter-Operability					X				
DIS	SISO/IEEE	Published	M&S-Simulation Inter-Operability				X	X	X	X	X	
HLA	SISO/IEEE	Published	M&S-Simulation Inter-Operability		X (OMT)	X (OMT)		X	X	X	X	
TENA	US DoD	Published	M&S-Simulation Inter-Operability			X	X	X	X	X	X	
CORBA	OMG	Published	Software Engineering					X	X	X	X	
MDA	OMG	Published	Software Engineering		X	X	X	X				
RUP	IBM	Published	Software Engineering			X	X	X	X			
Software QA Plans (1220)	IEEE	Published	Software Engineering				X	X				

STANDARD	Standard Development Organization	Status	Functional Area	Define Application Objectives	Perform Conceptual Analysis	Design Federation	Design Simulation	Develop Simulation	Develop Federation	Plan, Integrate and Test Application	Execute Application and Prepare Outputs	Analyze Data and Evaluate Results
Software Reuse Data Model (1420)	IEEE	Published	Software Engineering				X	X				X
UML	OMG	Published	Software Engineering			X	X	X	X			
DoD Architecture Framework	US DoD	Published	Systems Engineering	X	X	X						
MOD Architecture Framework	UK MOD	Published	Systems Engineering	X	X	X						
SysML	OMG/INCOSE	Published	Systems Engineering			X	X	X	X			
GM-V&V	SISO	Concept	V&V	X	X		X	X		X	X	X
REVVA1	EUCLID	Published	V&V	X	X	X	X	X	X	X	X	X
V&V Information Exchange	ITOP	Published	V&V		?					X	X	
VV&A Overlay to FEDEP	SISO/IEEE	Draft	V&V	X	X	X			X	X	X	X
VV&A RPG	US DoD	Published	V&V	X	X	X	X	X	X	X	X	X

F.2 RELEVANT STANDARDS CHARACTERIZATION

In the tables that follow, selected standards identified in Table F-1 above are described in considerable detail. The intention of the Task Group in providing this description is to provide users of the best-practice guidance contained formally in Annexes G and H below to leverage to best advantage – contingent enterprise and technical constraints and motivations – some of the several standards that are known by the Group to be relevant to both the specification of conceptual models themselves and to the expression of best-practice within enterprise contexts.

Standards described in detail in the tables following include:

- Based Object Model (BOM).
- Conceptual Modeling Language (CML).
- Capability Maturity Model Integration (CMMI).
- Defence Conceptual Modeling Framework (DCMF).
- Discrete Event Systems (DEVS).

- Department of Defense Architecture Framework (DoDAF).
- Generic Methodology for Verification, Validation (GM-VV).
- Integrated Definition Methods (IDEF0).
- Integrated Definition Methods (IDEF5).
- Joint Command Control Communications Information Exchange Data Model (JC3IEDM).
- Kernel Meta Meta Model (KM3).
- Model Driven Architecture (MDA).
- NATO Architecture Framework (NAF).
- Open Knowledge Base Connectivity (OKBC).
- Web Ontology Language (OWL).
- Resource Description Framework (RDF).
- Rational Unified Process (RUP).
- Systems Modeling Language (SysML).
- Unified Modeling Language (UML).

Table F-2: Based Object Model (BOM).

ATTRIBUTE	COMMENTS
Organization:	Simulation Interoperability Standards Organization (SISO)
Definition:	The BOM is a component-based standard for describing a reusable piece part of a federation or an individual federate. Specifically, the BOM specification offers ontology for characterizing elements of a simulation and relationships among conceptual entities within a simulation environment as a language neutral interface. BOMs can be used to document the interface for one or more of the following piece part elements: 1) Object classes; 2) Interaction classes; 3) Patterns of interplay; 4) State machines; and 5) Events. BOMs provide developers and users a modular approach for defining and adding new capabilities to a federate or federation, and in quickly composing object models such as HLA FOMs and SOMs through BOM Assemblies.
Intended Use:	Base Object Models (BOMs) provide a key mechanism in facilitating interoperability, reuse, and composability. BOMs are specifically identified in the IEEE 1516.3 HLA Federation Development and Execution Process (FEDEP) as a potential facilitator for providing reusable model components used for the rapid construction and modification of federates and federations. The open standardization of BOM representations is considered essential for encouraging their development, distribution and use. The BOM concept is based on the assumption that piece-parts of simulations and federations can be extracted and reused as modeling building-blocks or components.

ATTRIBUTE	COMMENTS
Intended Use (cont'd):	The interplay within a simulation or federation can be captured and characterized in the form of reusable patterns. These patterns of simulation interplay are sequences of events between simulation elements. The implementation of the pattern using HLA object model constructs is also captured in the BOM.
Community of Usage:	Primary in Military Modeling and Simulation domain but even across government and non-government applications worldwide.
Significant Attributes:	The BOM framework as documented in the BOM specification and the BOM guidance document is intended to influence the following six capabilities within the M&S community: Interoperability – The application of Extensible Markup Language (XML) and XML Schemas prescribed for BOMs provides a mechanism for defining and validating context, and facilitates understanding of the data being exchanged. Furthermore, the flexibility offered by BOMs allows for greater application of simulation interoperability within other domains. Reusability – The Meta data cataloged within a BOM such as intent-of-use, integration history, behavioral information, and potential visual information will facilitate greater reuse of components. Composability – BOMs will facilitate the ability to rapidly compose simulations and simulation environments both statically (design time) and dynamically (at run-time). Adaptability – Mega-BOMs produced by BOM compositions can be used to represent the standard data exchange interface for systems and simulations. Aggregation – The application of BOMs can be used for supporting two types of aggregation: Pattern Aggregation and Entity Aggregation. Pattern Aggregations reflect the coupling of interface groupings that can be identified prior to an exercise. Entity Aggregations reflect the coupling of multiple entities into a single inclusive group, which can be accomplished during a Federation Execution (FEDEX). Multi-resolution Models – At the Federate Capability Level, BOMs can be used to represent the behavior states needed for modeling a conceptual entity of one or more patterns of interplay. Federates can choose from an assortment of BOM Component Implementations (BCIs) of varying resolutions which can be swapped out dynamically during an exercise, assuming the proper precautions are taken to ensure validity and consistency.
Relevance to Our Work:	See FOI-R—2363—SE (BOM and DCMF), BOM++, a Semantically Enriched BOM.
Relevance of Form for Our Product Specification:	High: • Its Meta data port (Model Identification) can be re-used. • BOM fulfils some of basic requirements on a Conceptual model such as; reusability composability and syntactic interoperability. • BOM structure and mechanism (like pattern of interplay and state Machine) as well as BOM Assembly can be source of inspiration.
References:	http://www.boms.info/. "Simulation Interoperability Standards Organization (SISO) Base Object Model (BOM) Template Specification SISO-STD-003-2006", 31 March 2006 Copyright © 2004 – 2006 by the Simulation Interoperability Standards Organization, Inc., P.O. Box 781238, Orlando, FL 32878-1238, USA.

ATTRIBUTE	COMMENTS
References (cont'd):	Simulation Interoperability Standards Organization (SISO) Guide for Base Object Model (BOM) Use and Implementation SISO-STD-003.0 DRAFT-V0.12, 26 October 2005, Prepared by: SISO Base Object Model Product Development Group Keywords: Automation, Behavior, BOM, Components, Composability, Conceptual Model, FEDEP, Interoperability, Meta data, Patterns, Requirements, Reuse Copyright © 2004 – 2005 by the Simulation Interoperability Standards Organization, Inc., P.O. Box 781238, Orlando, FL 32878-1238, USA.

Table F-3: Conceptual Modeling Language (CML).

ATTRIBUTE	COMMENTS
Organization:	U.S. Army, OneSAF Program
Definition:	CML is the OneSAF conceptual modeling language.
Community of Usage:	CML is used within the OneSAF development enterprise as an implementation independent and computationally amenable tool to increase efficiency and security of transformation of information about the real world into simulation computational structures.
Significant Attributes:	CML consists of meta-class components and inter class relationships which, when instantiated, constitute a pictorial and text specification of the military mission space conceptual model. The existence of well-defined component class types, and intended interclass relationships provides a meta model, whereby the conceptual model practitioner may instantiate conceptual model artefacts that will be reduced to computational implementation in accordance with the pre-existing correlations between conceptual model components and computational component artefacts. CML class and relationships were expressly selected to facilitate representation of military mission-space scenarios. The use of CML has been observed to yield the following benefits: "1) improves KAKE and developer productivity; 2) provides a common frame of reference for all shareholders; 3) minimizes requirements creep by limiting KAKE to relevant issues; and 4) provides a sufficiently detailed description of the modeling solution to minimize misinterpretations and reinterpretations of the requirements."
Relevance to Our Work:	CML and its associated employment process are ostensibly highly relevant to the subject effort, having been specifically designed to serve for conceptual model specification for military models and simulations. The language further has been used and tailored/optimized to this purpose in context of the OneSAF simulation development environment. Specializations within CML related to capturing military mission space representation are specifically relevant to the current work. On the other hand, language features and associated process specializations that are peculiar to OneSAF simulation representation and/or implementation paradigm (such as simulation time advance mechanization, data modularization protocols, object class declarations and visibility, etc.) are likely not suitable for the present effort.

ATTRIBUTE	COMMENTS
Relevance to Our Work (cont'd):	(NOTE that notwithstanding the assertion that CMS is "implementation independent", several features – many implicit in the CML itself – seem to entail implementation-specific presumption.)
Relevance of Form for Our Product Specification:	Notational form for within CML is suggestive and relevant in many ways for military mission space representation. On the other hand, the notation is not either an industry standard nor does it evidently support translation to alternative notations. It is, in fact particularly specific to time management, data flow, and control flow specifically elected from a wider range of options for use in OneSAF simulation implementation. The notation as -is is not suitable for use in the documents resulting from this task. It may be appropriate and sufficient as a notation for implementation of the best-practice guidance being developed, if-and-only-if its intended application environment is sufficiently similar to that of OneSAF.
References:	"Conceptual Modeling in OneSAF Software Development", briefing provided by Greg Tackett in cooperation with the U. S. Army OneSAF Program Office.

Table F-4: Capability Maturity Model Integration (CMMI).

ATTRIBUTE	COMMENTS
Organization:	Carnegie Mellon, Software Engineering Institute (SEI)
Definition:	CMMI is not a process (M1 level). It is a process improvement approach (M2 level) that provides organizations with the essential elements of effective processes.
Intended Use:	It can be used to guide process improvement across a project, a division, or an entire organization. CMMI helps integrate traditionally separate organizational functions, set process improvement goals and priorities, provide guidance for quality processes, and provide a point of reference for appraising current processes. Two models exist. The CMMI for Development is a reference model that covers the development and maintenance activities applied to both products and services. Organizations from many industries, including aerospace, banking, computer hardware, software, defence, automobile manufacturing, and telecommunications, use CMMI for Development. This document is a reference model that covers the acquisition of needed capabilities. Capabilities are acquired in many industries, including aerospace, banking, computer hardware, software, defence, automobile manufacturing, and telecommunications. All these industries can use CMMI-ACQ.
Community of Usage:	It is difficult to quantify how many organizations have adopted CMMI because any organization can use CMMI for process improvement without having to register with the SEI or otherwise identify themselves to the public.

ATTRIBUTE	COMMENTS
Community of Usage (cont'd):	However, there have been over 55,000 people who have attended Introduction to CMMI training, and CMMI has been adopted in many industries (e.g., software, finance, and manufacturing) in countries around the world (e.g., United States, Australia, Japan, Brazil, and Russia).
Significant Attributes:	The CMMI model is composed of these main components: process areas, goals and practices. It also includes other components such as notes, examples, amplifications (domain specific notes and examples) and references. Some components are required, other are expected or informative. A process area is a cluster of related practices in an area that, when implemented collectively, satisfy a set of goals considered important for making improvement in that area. A specific goal describes the unique characteristics that must be present to satisfy the process area. Generic goals are called "generic" because the same goal statement applies to multiple process areas. A generic goal describes the characteristics that must be present to institutionalize the processes that implement a process area A specific practice is the description of an activity that is considered important in achieving the associated specific goal. The specific practices describe the activities that are expected to result in achievement of the specific goals of a process area. Generic practices are called "generic" because the same practice applies to multiple process areas. A generic practice is the description of an activity that is considered important in achieving the associated generic goal. A generic practice elaboration appears after a generic practice in a process area to provide guidance on how the generic practice should be applied uniquely to the process area. A sub-practice is a detailed description that provides guidance for interpreting and implementing a specific or generic practice The typical work-products are sample output from a specific practice. Levels are used in CMMI to describe an evolutionary path recommended for an organization that wants to improve the processes it uses to develop and maintain its products and services. CMMI supports two representations. The staged representation (maturity levels) offers a systematic way to approach process areas one stage at a time. Achieving each stage ensures that an adequate process infrastructure has been laid as a foundation for the next stage. In the continuous representation (capability levels), an organization can choose to improve different processes at different rates. A CMMI level can be officially certified following a rating activity of appraisals. The Standard CMMI Appraisal Method for Process Improvement (SCAMPI) provides benchmark quality ratings relative to CMMI models.
Relevance to Our Work:	Since the CMMI models have been designed in a generic manner to improve processes, some of its components (process areas, goals, practices, etc.) are directly applicable to our conceptual modeling process guidelines. The CMMI for Development is privileged because M&S is a developmental activity within the targeted organizations.

ATTRIBUTE	COMMENTS
Relevance to Our Work (cont'd):	The notion of maturity levels is also very useful to scale the Conceptual Model process specification to various organizational needs and to guide them toward our strategic vision (Level 5). This representation is privileged because it is easier to understand.
Relevance of Form for Our Product Specification:	The M2 level of language of the CMMI makes it appropriate to serve as an example for our Conceptual Model Process Spec.
References:	http://www.sei.CMu.edu/CMmi/, CMMI for Development, Version 1.2, CMMI Product Group, *Technical Report*, CMU/SEI-2006-TR-008, August 2006, 573 pp.

Table F-5: Defence Conceptual Modeling Framework (DCMF).

ATTRIBUTE	COMMENTS
Organization:	FOI – Swedish Defence Research Agency
Definition:	DCMF is a framework developed to understand and describe activities and processes in military operations. These descriptions will then be the foundation for developing conceptual models in a formal way. Those models are primarily used for simulation model development. The framework consists of a number of components which together support developers in the task of creating high quality conceptual models out of unstructured data. Examples of Framework's components are a specified modeling process, ontologies and specifications of the format of the models.
Intended Use:	The overall objectives for DCMF is to capture authorized knowledge of military operations; to manage, model and structure the obtained knowledge in an unambiguous way; and to preserve and maintain the structured knowledge for future use and reuse. The premier aim of doing so is to enable semantic and substantive interoperability of the future simulation models built on these descriptions. Another long-term goal with DCMF is reusability which will reduce costs and enhance the quality in the development of conceptual models. Our vision is that DCMF will evolve enough to become a standard for the development of simulation models within the Swedish Armed Forces.
Community of Usage:	FOI and Swedish Armed Forces.
Significant Attributes:	MSMs – Mission Space Models – which are the final result of the DCMF process, and the kernel of both DCMF and CMMS they are defined as simulation and implementation independent functional descriptions of the real-world processes, entities, and environmental factors associated with a particular set of missions.

ATTRIBUTE	COMMENTS
Significant Attributes (cont'd):	These descriptions would be able to serve as a frame of reference for simulation development by capturing the basic information about the important entities involved in any mission, and their key actions and interactions. It is to say these are for all stakeholders in the M&S process a common description of what is to be simulated and serve as a bridge between the military experts and the developers. The military experts own the mission process and are an authoritative source when validating the content of the conceptual models. MSMs also serve as a platform for communication among stakeholders working with these simulation models. Already within the Modeling and Simulation Master Plan (MSMP) developed by the US DoD the CMMS concept was presented as an essential requirement for interoperability and reusability of knowledge in the military domain. On top of that the DCMF initiative has tried to put some more concrete requirements. To summarize, the DCMF requirements for how the final conceptual models should be are as follows: • Well documented; • Readable and usable for a person as well as a machine; • Composable; • Traceable the whole way back to the original sources; and finally • Useable as a basis for simulation models.
Relevance to Our Work:	The framework provides: • A definition for a Conceptual Model; • A process for delivering high quality Conceptual Models; • A structure/template for the procedure Conceptual Models; • A formalism over Conceptual Models; • A list of potential stakeholder; and • A list of different roles and their interactions.
Relevance of Form for Our Product Specification:	At least a source of inspiration and an example of the way we can develop the Defence Conceptual Modeling Framework.
References:	FOI-R—1754--SE. FOI-R—2362--SE. 05F-SIW-038.

Table F-6: Discrete Event Systems (DEVS).

ATTRIBUTE	COMMENTS
Organization:	Arizona Center for Integrative Modeling and Simulation (ACIMS)

ATTRIBUTE	COMMENTS
Definition:	Discrete Event Systems (DEVS) formalism specification is a notation language (M1 level) specific to discrete event models.
Intended Use:	EVS is to be used as a standard notation for discrete event system Modeling.
Community of Usage:	Discrete event system modellers.
Significant Attributes:	The DEVS model components are: inputs, outputs, states and events.
Relevance to Our Work:	DEVS is not directly applicable to MSG-058 work because it specifies a Modeling language (M1 level) instead of providing guidance on how to specify a language. Furthermore, it is specific to the discrete event formalism, which makes it a Modeling language of a lower level of abstraction that is already well defined and outside the scope of the Task Group. However, DEVS could be stated as an example of well-known low-level conceptual modeling language.
Relevance of Form for Our Product Specification:	Due to its scoped target, the DEVS specification can be very formal. Its form is not appropriate to the MSG-058 product specification.
References:	ACIMS – www.acims.arizona.edu. DEVS Standardization Group – http://www.sce.carleton.ca/faculty/wainer/standard/.

Table F-7: Department of Defense Architecture Framework (DoDAF).

ATTRIBUTE	COMMENTS
Organization:	DoD U.S.
Definition:	Department of Defense Architecture Framework.
Intended Use:	The DoDAF defines a standard way to organize an Enterprise Architecture (EA) or systems architecture into complementary and consistent views. All major U.S. Government Department of Defense (DoD) weapons and information technology system procurements are required to develop and document an EA using the views prescribed in the DoDAF. While it is clearly aimed at military systems, DoDAF has broad applicability across the private, public and voluntary sectors around the world, and represents only one of a large number of systems architecture frameworks. It is especially suited to large systems with complex integration and interoperability challenges, and is apparently unique in its use of "operational views" detailing the external customer's operating domain in which the developing system will operate.
Community of Usage:	All major U.S. Government Department of Defense (DoD) procurements.

ATTRIBUTE	COMMENTS
Significant Attributes:	DoDAF views are organized into four basic view sets: • Overarching All View (AV); • Operational View (OV); • Systems View (SV); and • Technical Standards View (TV). It does not prescribe a modeling language (e.g., UML). April 23, 2007: Version 1.5 was released, 'Volume I: Definitions and Guidelines' (46 pages), 'Volume II: Product Descriptions' (284 pages), and 'Volume III: Architecture Data Description' (223 pages).
Relevance to Our Work:	Framework provides a potential definition for a conceptual model. It contains: • An overall, high-level scenario (OV-1) – includes scope, purpose; • Connectivity between and within systems (OV-2); • Information exchanged during system connectivity (OV-3); • Systems used by the organizations to perform the activities (OV-3); • Organizations performing the activities (OV-4); • Activities performed in the scenario (V-6); and • Data elements contained in information exchanges (OV-7).
Relevance of Form for Our Product Specification	The Task Group did not employ DoDAF schemas explicitly in its product specification on the grounds that while the DoDAF views, usage, and associated tools may well support practitioner's needs, the approach is inherently specific with respect to representational schemas required, and peculiar in its endorsement by one Member Nation represented on the Task Group. The degree to which practitioners following the best-practice guidance herein use DoDAF conventions is considered elective particularly with respect to the consequences of their efforts and the nature of military (or other) simulations' conceptual models.
References:	DoDAF Promulgation Memo Feb 9, 2004 – DODAF Policy Directive mandates use, all Architectures approved after 12/01/03 must be DODAF compliant. http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_I.pdf, DoDAF 1.5 Volume 1] – Provides definitions, guidelines, background material. http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_II.pdf, DoDAF 1.5 Volume 2] – Describes each architecture product. http://www.defenselink.mil/cio-nii/docs/DoDAF_Volume_III.pdf, DoDAF 1.5 Volume 3] – Provides the architecture data description. DoDAF 1.0 Deskbook – Provides supplementary "how to" information relating to architectures. The DODAF architecture documents were updated on April 23, 2007 to version 1.5. Currently the Deskbook, which is from February 9, 2004, has not been updated.

Table F-8: Generic Methodology for Verification and Validation (GM-V&V).

ATTRIBUTE	COMMENTS
Organization:	NMSG-073
Definition:	Standards: GM-VV to support acceptance of Models, Simulations, and Data.
Intended Use:	The GM-V&V aims to provide a common generic framework for making formal and well-balanced acceptance decisions on a specific usage of models, simulations (both legacy and new development) and data. Moreover, the GM-V&V comprises of methods, practices and techniques that capture the interplay between the allocation of V&V resources (costs, etc.), stakeholders' needs, and M&S usage risks in decision-making. The GM-V&V is based on a three pillar-view; product, process and organization. The objective of a GM-V&V based project is to make well-informed acceptance decisions on a specific usage of a model, simulations or data based upon a precise and formal argumentation. GM-V&V adopts a goal-driven approach to derive acceptance criteria from the stakeholders' purpose, and subsequently derives evidence criteria and associated tests from those acceptance criteria. This goal-driven approach is considered in a context of the M&S system intended use, development, use-risk, V&V cost-benefits and project constraints. The goal-based hierarchical derivation of criteria is on the one hand well suited for use in compliance with (an adapted) ISO/IEC 9126, and on the other hand clearly focuses on the special elements needed for measuring the quality of the Modeling part. The hierarchical derivation starts with the goal to show that the Conceptual Model provides utility for its use in the development of an end-product. From that goal other utility type of goals can be derived, which can also be further broken down into validity goals, related to the Modeling abstractions, and correctness goals, related to the implementation of needed Conceptual Model views. Some of the utility and correctness criteria are covered by ISO/IEC 9126, others must be found elsewhere such as in [Lindland], [Pace] or [Teeuw] The validity criteria will be highly domain and application dependent and must thus be derived for each Conceptual Model anew until domain/application specific referents are constructed. After the criteria have been defined test must be defined and comparison material made available in order to produce evidence. The tests are for conceptual models often executed in the form of literature comparisons, comparisons with existing data on the same domain or problem and often it will be the collection of expert opinion. After collection of evidence it is "summed up" as a V&V Claim Network to the level of the top goal. If the top claim is equal to the top goal, fitness for purpose of the Conceptual Model is shown. If this is not the case the culprit(s) can be found by tracing back from the top claim via the claim hierarchy to evidence that indicate failed criteria and up the goal hierarchy to find for which purposes the Conceptual Model can not be proven to be good.
Community of Usage:	Modeling & Simulation.
Significant Attributes:	Systematic guidance for V&V.

ATTRIBUTE	COMMENTS
Relevance to Our Work:	Below mappings are provide between the GM-V&V and the guidance in this report. Mapping of Conceptual Model processes/products/roles to GM-V&V is shown in the tables below. Mapping of Roles Defined in this Report to the Roles Defined in GM-V&V.

NMSG-058	GM-V&V
Initiator/Sponsor/Client	VV&A Sponsor
M&S Project Manager	
Custodian/Administrator	
Controller/VV&A Agent	VV&A Project Manager, Acceptance Leader, V&V Leader, V&V Implementers
Consumers / Conceptual Model Users	VV&A Problem Owner

Mapping between GM-V&V Processes Steps and the Process Activities Described in this Report.

NMSG-058	GM-V&V
PP1 – Initiate Conceptual Model Development	VV&A Requirements Definition Process VV&A Requirements Analysis Process
PP2 – Define Requirements and Analyze Knowledge Needs for the Conceptual Model	Partial: V&V Design Process V&V Implementation Process V&V Integration Process
PP3 – Acquire Mission Space and Simulation Space Knowledge	Partial: V&V Design Process V&V Implementation Process V&V Integration Process
PP4 – Design the Conceptual Model	Partial: V&V Design Process V&V Implementation Process V&V Integration Process
PP5 – Build the Conceptual Model	Final: V&V Design Process V&V Implementation Process V&V Integration Process Acceptance Assessment Process VV&A Transition Process

ATTRIBUTE	COMMENTS
Relevance to Our Work (cont'd):	Mapping of GM-V&V "Worlds" to NMSG-058 "Spaces". NMSG-058 / GM-V&V Mission space / Real world Mission space / Problem world Simulation space / M&S world User space / Product world
Relevance of Form for Our Product Specification	N/A pending completion of SISO product standard.
References:	http://www.sisostds.org/StandardsActivities/DevelopmentGroups/GMVVPDGGenericMethodologyforVVAintheM.aspx.

Table F-9: Integration Definition for Function Modeling (IDEFØ).

ATTRIBUTE	COMMENTS
Organization:	NIST – National Institute of Standards and Technology. In December 1993, the Computer Systems Laboratory of the National Institute of Standards and Technology (NIST) released IDEFØ as a standard for Function Modeling in FIPS Publication 183.
Definition:	Process oriented function in cell (node), controls/inputs/outputs and mechanisms-on-arrow notation.
Intended Use:	IDEFØ models are often created as one of the first tasks of a system development effort. IDEFØ is useful in establishing the scope of an analysis, especially for a functional analysis. As a communication tool, IDEFØ enhances domain expert involvement and consensus decision-making through simplified graphical devices. As an analysis tool, IDEFØ assists the modeler in identifying what functions are performed, what is needed to perform those functions, what the current system does right, and what the current system does wrong.
Community of Usage:	Systems engineers needing function or process-oriented representational notation.
Significant Attributes:	IDEFØ is a method designed to model the decisions, actions, and activities of an organization or system. Effective IDEFØ models help to organize the analysis of a system and to promote good communication between the analyst and the customer. Thus, The "box and arrow" graphics of an IDEFØ diagram show the function as a box and the interfaces to or from the function as arrows entering or leaving the box.

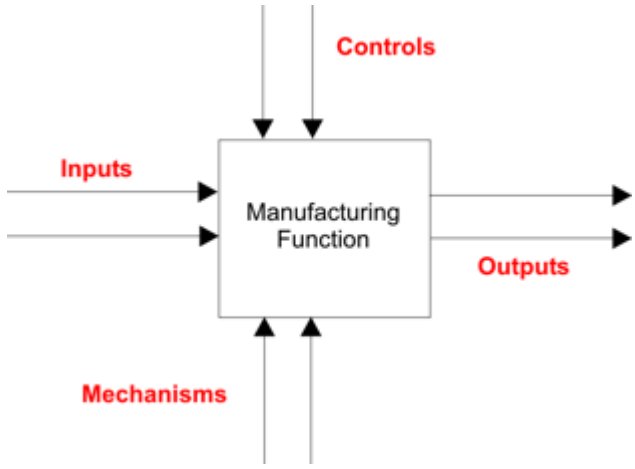
ATTRIBUTE	COMMENTS
Significant Attributes (cont'd):	To express functions, boxes operate simultaneously with other boxes, with the interface arrows "constraining" when and how operations are triggered and controlled. The basic syntax for an IDEFØ model is shown in the figure below.  **Figure F-1: IDEFØ Box and Arrow Graphics.**
Relevance to Our Work:	The fundamentals of function-on-node and control-on-arrow notation were widely and conveniently used by the Group in establishing and explaining the conceptual modeling process.
Relevance of Form for Our Product Specification:	While the semantics conveniently captured in the IDEFØ notation schema were carefully considered in establishing the resulting formal specification of best-practice process contained in Annex G; nevertheless, neither IDEFØ notation nor tools were used to produce that guidance in order not to intimate to practitioners that the notation per se was either required or preferentially recommended.
References:	http://www.idef.com/IDEF0.htm; http://www.idef.com/pdf/idef0.pdf.

Table F-10: Integration Definition for Function Modeling (IDEF5).

ATTRIBUTE	COMMENTS
Organization:	Armstrong Laboratory, AL/HRGA, Wright-Patterson Air Force Base, Ohio 45433
Definition:	Ontology Description Capture Method.
Intended Use:	The IDEF5 method provides a theoretically and empirically well-grounded method specifically designed to assist in creating, modifying, and maintaining ontologies. Standardized procedures, the ability to represent ontology information in an intuitive and natural form, and higher quality results enabled through IDEF5 application also serve to reduce the cost of these activities.

ATTRIBUTE	COMMENTS
Community of Usage:	Systems engineers / Ontology analysts.
Significant Attributes:	Supporting the ontology development process are IDEF5's ontology languages. There are two such languages: the IDEF5 schematic language and the IDEF5 elaboration language. The schematic language is a graphical language, specifically tailored to enable domain experts to express the most common forms of ontological information. The other language is the IDEF5 elaboration language, a structured textual language that allows detailed characterization of the elements in the ontology.
Relevance to Our Work:	IDEF5 focuses particularly on Ontology languages. On that account it is neither as intuitive nor as powerful as UML or other such notations in addressing the generation of ontological descriptions *per se*, nor in documenting them for persistent reference as is necessary in simulation conceptual modeling. This bias and relative dislocation from the purpose of the Task Group's scope of interest is reflected in both the IDEF concepts and in its diagrammatic –notational vocabulary. The standard was not strongly or formally employed in the Task Group's effort, though it might have served well enough for the subject deliberations if its familiarity were more prevalent among team members.
Relevance of Form for Our Product Specification:	The standard was not invoked explicitly in formulation of an expression of the best-practice guidance despite its ontological intentions. Whether its use in execution of the best-practice proffered, is left as with all other standards to the discretion of the practitioner in context of the prevailing conceptual modeling enterprise environment.
References:	http://www.idef.com/IDEF5.htm; http://www.idef.com/pdf/Idef5.pdf.

Table F-11: Joint Command Control Communication Information Exchange Data Model (JC3IDEM).

ATTRIBUTE	COMMENTS
Organization:	The NATO Data Administration Group (NDAG) cooperates with the (Multi-national Interoperability Programme) MIP's Data Modeling Working Group (DMWG) in building of JC3IEDM.
Definition:	JC3IEDM, or Joint Command, Control and Consultation Information Exchange Data Model is an evolution of the Command and Control Information Exchange Data Model (C2IEDM) standard that includes joint operational concepts, just as the Land Command and Control Information Exchange Data Model (LC2IEDM) was extended to become C2IEDM.
Intended Use:	The overall goal is to specify the minimum set of data that needs to be exchanged in coalition or multi-national operations.

ATTRIBUTE	COMMENTS
Intended Use (cont'd):	Each NATO Nation, agency or community of interest is free to expand its own data dictionary to accommodate its additional information exchange requirements with the understanding that the added specifications will be valid only for the participating NATO Nation, agency or community of interest. Any addition that is deemed to be of general interest may be submitted as a change proposal within the configuration control process to be considered for inclusion in the next version of the specification. The main source of information and the basis for the ontology design and development is the MIP (Multi-lateral Interoperability Programme) proposed standard JC3IEDM. The MIP aims to provide an assured capability for interoperability of information to support joint military operations. Interoperability is not envisioned merely at a data level but also at strategic, operational and tactical level to allow proper planning and functioning of joint operations.
Community of Usage:	NATO North Atlantic Treaty Organisation.
Significant Attributes:	JC3IEDM is intended to represent the core of the data identified for exchange across multiple functional areas and multiple views of the requirements. Toward that end, it lays down a common approach to describing the information to be exchanged in a Command and Control (C2) environment. • The structure should be sufficiently generic to accommodate joint, land, sea, and air environmental concerns. • The data model describes all objects of interest in the sphere of operations, e.g., organizations, persons, equipment, facilities, geographic features, weather phenomena, and military control measures such as boundaries. • Objects of interest may be generic in terms of a class or a type and specific in terms of an individually identified item. All object items must be classified as being of some type (e.g., a specific tank that is identified by serial number WS62105B is an item of type "Challenger" that is a heavy UK main battle tank). • An object must have the capability to perform a function or to achieve an end. Thus, a description of capability is needed to give meaning to the value of objects in the sphere of operations. • It should be possible to assign a location to any item in the sphere of operations. In addition, various geometric shapes need to be represented in order to allow commanders to plan, direct, and monitor operations. Examples include boundaries, corridors, restricted areas, minefields, and any other control measures needed by commanders and their staffs. • Several aspects of status of items need to be maintained. • The model must permit a description of the composition of a type object in terms of other type objects. Such concepts include tables of organizations, equipment, or personnel. • The model must reflect information about what is held, owned or possessed in terms of types by a specific object item.

ATTRIBUTE	COMMENTS
Significant Attributes (cont'd):	• There is a need to record relationships between pairs of items. Key among these is the specification of unit task organizations and orders of battle. • The model must support the specification of current, past, and future role of objects as part of plans, orders, and events. • The same data structure should be used to record information for all objects, regardless of their hostility status. • Provision must be made for the identification of sources of information, the effective and reporting times, and an indication of the validity of the data. The JC3IEDM is described from three different perspectives: • Conceptual Data Model: A top, high level, Conceptual Data Model of generalized concepts such as Actions, Organizations, Materiel, Personnel, Features, Facilities, Locations, intended for top officers, senior commanders, etc., who do not need to know the specific technical details of the model, but is sufficient to be aware of the different concepts and their relationships. • Logical Data Model: Middle, Logical Data Model which is more detailed, is based upon breaking down the high level concepts into specific information that is regularly used. For example, a tank is an armored fighting vehicle that is a piece of equipment that is a piece of materiel. It also makes implicit knowledge explicit, like following the human reasoning patterns that a tank is a piece of armored fighting equipment and allows command and control systems to generalize by recognizing, for instance, that tanks are equipment. A logical data model specifies the way data is structured with an entity-attribute-relationship diagram and supporting documentation. At this level, technical implementation specific details are still obscured from view. This level is useful for middle level system analysts and domain experts. • Physical Data Model: The third and lower most, Physical Data Model provides the detailed specifications that are necessary to generate a physical schema that defines the structure of a database. Mainly intended for the information system developer. The physical data model can be seen as a traditional database schema model, which illustrates the different logical concepts (tables), their attributes (fields) and the relationships.
Relevance to Our Work:	• To describe a Conceptual Model, one needs the same concepts of Objects or Entities of interests around which the operation is focused. • Each entity has both static characteristics as well as dynamic properties, as is represented by the situation concepts in the JC3IEDM. • Relevant if the main focus in our conceptual models is that of activities, or Actions as proposed in the JC3IEDM as well. • And to co relate pieces of information, to provide the context and other vital information, a group of information packages is required as well.
Relevance of Form for Our Product Specification:	It depends on our chosen Conceptual models structure, content or process will make use of.

ATTRIBUTE	COMMENTS
References:	http://www.mip-site.org. http://encyclopedia.thefreedictionary.com/JC3IEDM. http://en.wikipedia.org/wiki/JC3IEDM. FOI-R—175--SE.

Table F-12: Kernel Meta Meta Model (KM3).

ATTRIBUTE	COMMENTS
Organization:	FOI – Swedish Defence Research Agency
Definition:	The Knowledge Meta Model (KM3), developed by the Swedish Defence Research Institute (FOI), has its place as a tool and a language to construct well-formed conceptual models that are to be used successfully in simulation model and Development.
Intended Use:	The intention when producing the KM3 was not to construct a grand "unified model description language". It rather represents one possibility to capture system structures and behavior in an object-oriented and rule-based way. The KM3 is all of the following (in no particular order): • The KM3 is a specification. It is a specification consisting of object-oriented concepts, primarily aimed at capturing different dependencies in and between activities. In this setting this means that the KM3 is a specification for the creation of generic and reusable conceptual models of objects and processes of (military) interest. • The KM3 is a tool. It is a tool for structuring knowledge about objects and processes as conceptual models. The main objective of KM3 is to produce generic templates of knowledge (MSMs, in the above list). • The KM3 is a language. It is a common language to for different stakeholders involved in the modeling process, to enable them to construct conceptual models. The KM3 is mainly used as a specification for construction of generic models, which in turn are used to model knowledge at an instance level. KM3 is, in this respect, a model for how to make models. A model produced using the KM3 is a well-formed, well-understood conceptual model which in turn can be instantiated to be used as a simulation model.
Community of Usage:	FOI and Swedish Armed Forces.
Significant Attributes:	• Is an *activity centric* structure. • Covers *static* and *dynamic* aspects of objects in the same model. • Covers *relations* between objects. • Captures *rules* of behaviour.

ATTRIBUTE	COMMENTS
Relevance to Our Work:	• A general and flexible structure and a potential candidate for Conceptual Model Template. • The interpreted information is subsequently transformed into a common format, generalized and stored as a reusable model. This common format is described by a Knowledge Meta Model (KM3). • The reusable conceptual model, now called a Mission Space Model (MSM), can be instantiated with real-world data and serve as a basic structure when performing simulations.
Relevance of Form for Our Product Specification:	A source of inspiration and example of the way we can specify our Conceptual Models.
References:	FOI-R—1754--SE. 05F-SIW-040.

Table F-13: Model Driven Architecture (MDA).

ATTRIBUTE	COMMENTS
Organization:	Object Management Group (OMG)
Definition:	Model Driven Architecture (MDA) is a software development process (M1 level, because it specifies the Modeling languages). It is the base architecture for OMG's software development standards, including MOF, UML, OWM and XMI.
Intended Usage:	MDA separates software business and application logic from underlying platform technology. By leveraging OMG's universally accepted MOF and UML standards, the MDA allows creation of software applications that are portable across, and interoperate naturally across, a broad spectrum of systems from embedded, to desktop, to server, to mainframe, and across the Internet. In MDA, attention focuses first on the application's business functionality and behavior, allowing stakeholders' investment to concentrate on the aspects that critically affect core business processes. Technical aspects, also critical but secondary to business functions, are well handled by automated or semi-automated development tools. MDA is always ready to deal with yesterday's, today's and tomorrow's challenges and makes it easier to integrate applications and facilities across middleware boundaries. Domain facilities defined in the MDA by OMG's Domain Task Forces provide much wider interoperability by always being available on a domain's preferred platform, and on multiple platforms whenever there is a need.
Community of Usage:	OMG Task Forces organized around industries including Finance, Manufacturing, Biotechnology, Space technology, and others use the MDA to standardize facilities in their domains.

ATTRIBUTE	COMMENTS
Significant Attributes:	The MDA is structured around Computation Independent Model (CIM), Platform Independent Model (PIM), Platform Specific Model (PSM) and implementations: • The CIM represents the requirements of the system in the form of a domain model. It is also a source of vocabulary for the other models. The transformations from and to this model are mainly for requirements traceability. • The PIM describes a system without anchoring it to technology-specific functional interfaces. • The PSM implements in an abstract fashion the specification of the PIM using technology-specific functionalities. • The Implementation is the operational system. The different models are linked through automatic conversions. This improves traceability, consistency, uniformity and productivity. The MDA is implemented using OMG's software development standards. The MOF is OMG's foundation specification for modeling languages; MOF compliance allows UML structural and behavioral models, and CWM data models, to be transmitted via XMI, stored in MOF-compliant repositories, and transformed and manipulated by MOF-compliant tools and code generators. Patterns play a critical role in most MDA-based development projects. Successful transformation from PIM to PSM, and from PSM to code, requires that the PIM contain enough detail to completely guide the software tools through the process. By incorporating this detail through the use of patterns, instead of inserting it by hand, we gain multiple benefits: architects do less work, the resulting PIM reflects the collective wisdom of many contributors, and the tools can work the pattern (parameterized as necessary in our UML models) through the transformations, ultimately pulling implementation code from a library written by experts and inserting it into the application.
Relevance to Our Work:	The MDA process could be applied directly to M&S conceptual modeling, i.e., that a new MOF-based conceptual model language adapted to military M&S could be developed or an existing one could be reused (e.g., UML). However, since the MSG-058 mandate is to provide a guidance (M2 level), proposing the MDA as-is would be too prescriptive. MDA could be stated as an example of a development process implementation leading to a high level of maturity: focus on functionality and behavior while technical aspects are automated or semi-automated, high maintainability and interoperability through shared domain facilities. The notion of patterns is also desirable to apply in the MSG-058 work-product.
Relevance of Form for Our Product Specification:	The MDA philosophy (different abstraction levels linked with automatic transformations) is applicable to the MSG-058 product specification. It has influenced our strategic goal because it greatly improves the level of maturity of a process.
References:	MDA – http://www.omg.org/mda/. MDA Guide Version 1.0.1, OMG, omg/2003-06-01, June 2003, 62 pp.

ATTRIBUTE	COMMENTS
References (cont'd):	MOF – http://www.omg.org/mof/. Meta Object Facility (MOF) Specification Version 1.4, OMG, April 2002, 358 pp. UML – http://www.uml.org/. Unified Modeling Language Specification Version 1.4.2, OMG formal/05-04-01, ISO/IEC 19501:2005(E), January 2005, 454 pp. OMG Unified Modeling Language (UML), Infrastructure, V2.1.2, formal/2007-11-04, November 2007, 224 pp. OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2, formal/2007-11-02, November 2007, 738 pp.

Table F-14: NATO Architecture Framework (NAF).

ATTRIBUTE	COMMENTS
Organization:	NATO, NC3A
Definition:	NATO Architecture Framework is a derivative framework based on DoDAF.
Intended Use:	• It provides guidance on describing communication and information systems (or C3 systems) through architectures and provides the rules, guidance, and templates for developing and presenting architecture descriptions to ensure a common denominator for understanding, comparing, and integrating architectures in NATO. The application of the NAF is designed to enable architectures to contribute most effectively to acquiring and fielding cost-effective and interoperable military capabilities. • The ability to define views of architectural information in a more flexible way and support Stakeholders so that extensive analysis can be made to provide rationale for prioritization decision-making. • A standardized way of documenting NATO-wide business processes and to provide support to Capability-based planning. • Critical support for the achievement of NNEC and NATO transformation by facilitating the move from a system-oriented paradigm to a service-oriented paradigm, and by identifying mechanisms to handle the complexity of the relationships within the NATO federation of systems in a holistic manner. • A NAF Meta-Model (NMM) and repository to enable stakeholders and users to extract bespoke architecture information and make necessary analyses to support development, interoperability, acquisition or technical considerations. • A complementary tool to NATO and National programme management, contributing to reduction in cost overruns, risk reduction, and more efficient use of common funded budgets.

ATTRIBUTE	COMMENTS
Intended Use (cont'd):	• A coherent mechanism to identify capability gaps and promote interoperability across NATO, including for the critical deployed force and NRF scenarios.
Community of Usage:	NATO, PfP.
Significant Attributes:	The NAF is still in development. NAF introduces a number of new "Service Views" to support Service Orientated Architecture. It is likely that DoDAF and NAF may converge into a single or very closely related architecture sometime in the near future.
Relevance to Our Work:	See DoDAF.
Relevance of Form for Our Product Specification	See DoDAF.
References:	http://194.7.80.153/website/book.asp?menuid=15&vs=0&page=volume1%2Fch03s02.html.

Table F-15: Open Knowledge Base Connectivity (OKBC).

ATTRIBUTE	COMMENTS
Organization:	The development of OKBC is being overseen by a working group. Richard Fikes is the working group chair and the following six institutions are the voting members: • Cycorp; • Information Sciences Institute; • Knowledge Systems Laboratory, Stanford; • Science Applications International Corporation (SAIC); • SRI International; and • Teknowledge.
Definition:	OKBC is an application programming interface for accessing knowledge bases stored in Knowledge Representation Systems (KRSs). OKBC is being developed under the sponsorship of DARPA's High Performance Knowledge Base program (HPKB), where it is being used as an initial protocol for the integration of various technology components.

ATTRIBUTE	COMMENTS
Intended Use:	[OKBC] … provides a set of operations for a generic interface to underlying KRSs ... OKBC is complementary to language specifications developed to support knowledge sharing. KIF, the Knowledge Interchange Format, provides a declarative language for describing knowledge. As a pure specification language, KIF does not include commands for knowledge base query or manipulation. Furthermore, KIF is far more expressive than most KRSs. OKBC focuses on operations that are efficiently supported by most KRSs (e.g., operations on frames, slots, facets — inheritance and slot constraint checking). OKBC is intended to be well impedance-matched to the sorts of operations typically performed by applications that view or manipulate object-oriented KRSs.
Community of Usage:	Knowledge management specialists.
Significant Attributes:	OKBC specifies a knowledge model of KRSs (with KBs, classes, individuals, slots, and facets). It also specifies a set of operations based on this model (e.g., find a frame matching a name, enumerate the slots of a frame, delete a frame). An application uses these operations to access and modify knowledge stored in an OKBC-compliant KRS.
Significant Attributes:	The current implementation of OKBC is object-oriented: methods in the appropriate object-oriented programming language for an application are used to implement OKBC operations. We refer to the set of methods that implement the protocol for a particular KRS as a back end. Many OKBC operations have default implementations written in terms of other OKBC operations; therefore, the programmer need define only a core sub-set of all OKBC operations in order to implement a compliant back end. These OKBC operations are called mandatory, and they comprise the OKBC kernel. The default implementations can be overridden within a given back end to improve efficiency. The design objectives for OKBC are as follows. Simplicity: It is important to have a relatively simple specification that can be implemented quickly, even if that means sacrificing theoretical considerations or support for idiosyncrasies of a particular KRS. Generality: The protocol should apply to many KRSs, and support all the most common KRS features. For example, it should support all the knowledge access and modification functionality that will be required by a graphical KB editor. The protocol should not require numerous changes to a KRS for which the protocol is implemented. That is, the protocol should not legislate the behavior of an underlying KRS, but should serve as an interface between an existing KRS and an application. Performance: Inserting the protocol between an application and a KRS should not introduce a significant performance cost. Consistency: The protocol should exhibit consistent behavior across different KRSs. That is, a given sequence of operations within the protocol should yield semantically equivalent results over a range of KRSs.
Relevance to Our Work:	OKBC is of potential use in executing any of the various knowledge collection and management activities cited within the best-practice process.

ATTRIBUTE	COMMENTS
Relevance of Form for Our Product Specification:	While OKBC was not used explicitly in the best-practice process specification, it is understood that utility may be found by the practitioner in executing the best-practice process and in documenting the conceptual model product itself.
References:	http://www.ai.sri.com/~okbc/. http://www.ai.sri.com/~okbc/spec.html.

Table F-16: Web Ontology Language (OWL).

ATTRIBUTE	COMMENTS
Organization:	W3C – Worldwide Web Consortium
Definition:	"OWL is the W3C's recommended ontology Language for representing information on the Semantic Web. It is typically used to define an ontology for a particular domain. An OWL ontology is a set of axioms describing classes, properties, and the relationships between them." – Lacey 2005.
Intended Use:	"The purpose of OWL is to provide a standard language for Semantic Web information representation." – Lacey. The W3C OWL 2 Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things." OWL is a computational logic-based language such that knowledge expressed in OWL can be reasoned with by computer programs either to verify the consistency of that knowledge or to make implicit knowledge explicit. OWL documents, known as ontologies, can be published in the World Wide Web and may refer to or be referred from other OWL ontologies" – http://www.w3.org/TR/2009/REC-owl2-primer-20091027/.
Community of Usage:	Data managers, web application developers, and database engineers who what to leverage the power of the Semantic Web by representing their information using the Web Ontology Language – OWL in order to enable Semantic Web applications and services to process and interpret content.
Significant Attributes:	"OWL 2 is a language for expressing ontologies." "OWL 2 is not a programming language: OWL 2 is declarative, i.e., it describes a state of affairs in a logical way" "OWL 2 is not a schema language for syntax conformance." "OWL 2 is not a database framework." "OWL 2 is a knowledge representation language, designed to formulate, exchange and reason with knowledge about a domain of interest". OWL assumes basic knowledge modeling practice and includes provision for representation of information related to classes, properties, and individuals. For survey of features see: http://www.w3.org/TR/2009/REC-owl2-primer-20091027/#ref-owl-2-quick-reference. The currently most widely used OWL editor is Protégé, a free open-source editing framework developed at Stanford University.

ATTRIBUTE	COMMENTS
Significant Attributes (cont'd):	By virtue of its open plug-in structure, it allows for the easy integration of special-purpose ontology editing components. Other editors include TopQuadrant's commercial TopBraid Composer and the open-source systems SWOOP and NeOn-Toolkit. There are several reasons for OWL DL which differs somewhat in terms of coverage of the supported reasoning features. For some of these, OWL 2 conformance is currently planned and the corresponding implementations are in progress. The Test Suite Status document lists to which extent some of the reasons mentioned below comply with the test cases. For reasoning within OWL DL, the most prominent systems are Fact++ by the University of Manchester, Hermit by Oxford University Computing Laboratory, Pellet by Clark & Parsia, LLC, and RacerPro by Racer Systems. In addition to those general-purpose reasons aiming at supporting all of OWL DL, there are reasoning systems tailored to the tractable profiles of OWL. CEL by Dresden University of Technology supports OWL EL. QuOnto by Sapienza University di Roma supports OWL QL. ORACLE 11 g supports OWL RL. The open-source OWL API plays a rather prominent role as the currently most important development tool around OWL.
Relevance to Our Work:	The OWL standard and knowledge/concept specification schema is designed to address the description of ontological domains. In the present work, such relevant domains are the military mission space and simulation executive domains. Therefore OWL is a language whose relevance to expressing the semantic content of military model and simulation conceptual models is direct and obvious. Further, toe prospect of using Semantic Web interface and communications infrastructure in association with cloud computing techniques is highly suggestive for the present effort. Relationship of OWL to XML is also prospectively significant for consideration of machine readable military model and simulation conceptual models.
Relevance of Form for Our Product Specification:	The notational schema manifest in OWL for ontology specification and semantic elaboration is effectively similar to that of UML and is manifestly suitable to the needs for notational specification of military simulation conceptual models.
References:	OWL: Representing Information Using the Web Ontology Language, Lee W. Lacey, Trafford, Victoria, BC, Canada, 2005; and http://www.w3.org/TR/#tr_OWL_Web_Ontology_Language.

Table F-17: Resource Description Framework (RDF).

ATTRIBUTE	COMMENTS
Organization:	World-Wide-Web Consortium (W3C)
Definition:	"RDF is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed."

ATTRIBUTE	COMMENTS
Intended Use:	Semantic description of data exchanged between independent users. It is not human readable.
Community of Usage:	Anyone who need to describe web content such that independent applications can discover and retrieve it.
Significant Attributes:	"RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a "triple"). Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications. This linking structure forms a directed, labeled graph, where the edges represent the named link between two resources, represented by the graph nodes. This *graph view* is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations." RDF is written in XML. It is designed to be read by computers. Its basic building block is an object-attribute-value triple, called a statement.
Relevance to Our Work:	May be used for representing Meta data about conceptual models.
Relevance of Form for Our Product Specification:	The Task Group did not use RDF standard specifications or standards in framing the best-practice guidance for military model conceptual modeling on the assumption that our work-product would be a text document and that a practitioner would leverage such relatively fundamental standards in accordance with their own enterprise practices in due course of developing and managing simulation conceptual models.
References:	http://www.w3.org/RDF/. http://www.w3.org/standards/techs/rdf. http://www.w3.org/2001/sw/.

Table F-18: Rational Unified Process (RUP).

ATTRIBUTE	COMMENTS
Organization:	IBM
Definition:	"IBM Rational Unified Process® (RUP®) is a comprehensive process framework that provides industry-tested practices for software and systems delivery and implementation and for effective project management."
Intended Use:	**"**RUP promotes iterative development and organizes the development of software and systems into four phases, each consisting of one or more executable iterations of the software at that stage of development."
Community of Usage:	Software developers in enterprise environments and project teams.
Significant Attributes:	Systematic, tailorable process for software development. Extensive support tools and training/coaching available.

ATTRIBUTE	COMMENTS
Relevance to Our Work:	Significant software development bias but includes software system conceptual modeling within scope.
Relevance of Form for Our Product Specification	N/A.
References:	http://www-01.ibm.com/software/awdtools/rup/. http://www.ibm.com/developerworks/rational/library/content/03July/1000/1251/1251 bestpractices_TP026B.pdf. "Rational Unified Process, Best-Practices for Software Development Teams, Rational Software – White Paper", TP026B, Rev 11/01.

Table F-19: Systems Modeling Language (SysML).

ATTRIBUTE	COMMENTS
Organization:	The UML for Systems Engineering RFP was developed jointly by the OMG and the International Council On Systems Engineering (INCOSE) and issued by the OMG in March 2003
Definition:	Systems Modeling Language.
Intended Use:	The SysML is a Domain-Specific Modeling language for systems engineering. It supports the specification, analysis, design, verification and validation of a broad range of systems and systems-of-systems. SysML was originally developed by an open source specification project, and includes an open source license for distribution and use. SysML is defined as an extension of a sub-set of the Unified Modeling Language (UML) using UML's profile mechanism. SysML offers system engineers several noteworthy improvements over UML, which tends to be software-centric. These improvements include the following: SysML's semantics are more flexible and expressive. SysML reduces UML's software-centric restrictions and adds two new diagram types, requirement and parametric diagrams. The former can be used for requirements management; the latter can be used for performance analysis and quantitative analysis. As a result of these enhancements, SysML is able to model a wide range of systems, which may include hardware, software, information, processes, personnel, and facilities. SysML is a smaller language that is easier to learn and apply. Since SysML removes many of UML's software-centric constructs, the overall language is smaller as measured both in diagram types and total constructs. SysML allocation tables support common kinds of allocations. Whereas UML provides only limited support for tabular notations, SysML furnishes flexible allocation tables that will support requirements allocation, functional allocation, and structural allocation. This capability facilitates automated Verification and Validation (V&V) and gap analysis.

ATTRIBUTE	COMMENTS
Intended Use (cont'd):	SysML model management constructs support models, views, and viewpoints. These constructs extend UML's capabilities and are architecturally aligned with IEEE-Std-1471-2000 (IEEE Recommended Practice for Architectural Description of Software Intensive Systems). SysML uses seven of UML 2.0's thirteen diagrams, and adds two diagrams (requirements and parametric diagrams) for a total of nine diagram types. SysML also supports allocation tables, a tabular format that can be dynamically derived from SysML allocation relationships.
Community of Usage:	Systems Engineering.
Significant Attributes:	The Object Management Group announced the adoption of the OMG SysML™ on July 6, 2006 and the availability of OMG SysML™ v1.0 in September 2007. With SysML you can use Requirement diagrams to efficiently capture functional, performance and interface requirements, whereas with UML you are subject to the limitations of Use Case diagrams to define high-level functional requirements. Likewise, with SysML you can use Parametric diagrams to precisely define performance and mechanical constraints such as maximum acceleration, curb weight, air conditioning capacity, and interior cabin noise management. UML provides no straightforward mechanism to capture this essential performance and mechanical information.
Relevance to Our Work:	SysML can be used as a conceptual model language format.
Relevance of Form for Our Product Specification	Relatively little – See UML.
References:	http://www.omgsysml.org/.

Table F-20: Unified Modeling Language (UML).

ATTRIBUTE	COMMENTS
Organization:	OMG – Object Modeling Group
Definition:	Modeling notation and syntactic denotation specification created for software development, but evolved into practical systematic, comprehensive and formal notational method for specification of any system.
Intended Use:	"... enabling object visual modeling tool interoperability."
Community of Usage:	"The objective of UML is to provide system architects, software engineers, and software systems as well as for modeling business and similar processes with tools for analysis, design, and implementation of software-based systems as well as for modeling business and similar processes."

ATTRIBUTE	COMMENTS
Significant Attributes:	"The modeling concepts of UML are grouped into language units. A language unit consists of a collection of tightly coupled modeling concepts that provide users with the power to represent aspects of the system under study according to a particular paradigm or formalism. For example, the State Machines language unit enables modelers to specify discrete event-driven behavior using a variant of the well-known state charts formalism, while the Activities language unit provides for modeling behavior based on a workflow-like paradigm. A model contains three major categories of elements: Classifiers, events, and behaviors. Each major category models individuals in an incarnation of the system being modeled. A classifier describes a set of objects; an object is an individual thing with a state and relationships to other objects. An event describes a set of possible occurrences; an occurrence is something that happens that has some consequence within the system. A behavior describes a set of possible executions; an execution is the performance of an algorithm according to a set of rules. Models do not contain objects, occurrences, and executions, because those things are the subject of models, not their content. Classes, events, and behaviors model sets of objects, occurrences, and executions with similar properties. Value specifications, occurrence specifications, and execution specifications model individual objects, occurrences, and executions within a particular context. The distinction between objects and models of objects, for example, may appear subtle, but it is important. Objects (and occurrences and executions) are the domain of a model and, as such, are always complete, precise, and concrete. Models of objects (such as value specifications) can be incomplete, imprecise, and abstract according to their purpose in the model."
Relevance to Our Work:	Highly relevant to the present work in multiple ways. First, UML is a suitable notational schema in which to manifest, represent, and document a simulation conceptual model itself. The notation is powerful and versatile, and it has extensive mechanisms for specialization to alternative representation domains. As far as is known, UML is sufficient (though not necessary) to specify any simulation conceptual model envisioned by the present study. In addition, UML might be used to document the best-practice process and intended conceptual model work-product specification conceived by the MSG-058 Task Group.
Relevance of Form for Our Product Specification:	While not used extensively within the formal simulation conceptual model process and product specifications of Annexes G and H below or in the accompanying text of Chapters 4, 5 and 6 of the document, UML could have been used as the primary meta-language to provide such specifications. As a practical matter, the Task Group felt is desirable not to adopt pre-emptively any single notational schema lest the practitioner find himself constrained artificially to that choice. Nevertheless, UML is a candidate for future (alternative) best-practice standards specifications.
References:	OMG Unified Modeling Language TM (OMG UML), UML Superstructure Specification, v2.3, May 2010. OMG Unified Modeling Language TM (OMG UML), UML Infrastructure Specification, v2.3, May 2010. http://www.uml.org/. http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML.