Model- Based System of Systems Engineering with UPDM

Matthew Hause
Artisan Software Tools
Eagle Tower Suite 701
Cheltenham, Glos, GL50 1TN
England
Matthew.Hause@ArtisanSoftwareTools.com

Abstract. The Unified Profile for DoDAF and MODAF, (UPDM) initiative was started by members of INCOSE and the OMG. UPDM provides a consistent, standardized means to describe DoDAF 1.5 and MODAF 1.2 architectures in SysML/UML-based tools as well as a standard for interchange. The concepts found in the Systems Modeling Language (SysML) such as parametrics, blocks, complex ports, enhanced activity modeling, and cross-cutting constructs improve the state of the art for systems engineers and architects. The formal meta-model basis of UPDM also provides a basis for trade-off analysis, model execution, requirements traceability, and the transition to systems development and implementation. Finally, the interconnections between views can help combat stovepipe development and improve communication. This paper looks at the current direction of UPDM, how it is improving the state of the art for system architects, and enables interchange of architectural information. We will also examine some of the latest concepts found in DoDAF 2.0 and how the UPDM Group is addressing these.

INTRODUCTION

What is a Military Architectural Framework? Arguably, the two most widely used military frameworks are the Department of Defense (DoD) Architecture Framework (DoDAF) in the USA and the Ministry of Defence (MOD) Architecture Framework (MODAF) in the UK. Military Architectural Frameworks such as DoDAF define a standard way to organize an enterprise architecture (EA) or systems architecture into complementary and consistent views. DoDAF contains four basic views: the overarching All Views (AV), Operational View (OV), Systems View (SV), and the Technical Standards View (TV). Each view is aimed at different stakeholders, and it is possible to create cross-references between the views. Although they were originally created for military systems, they are commonly used by the private, public and voluntary sectors around the world, to model complex organizations such as humanitarian relief organizations and public services such as the Federal Emergency Management Agency (FEMA). Their goal is to improve planning, organization, procurement and management of these complex organizations. All major DoD weapons and information technology system procurements are required to document their enterprise architectures using DoDAF.

Since the introduction of DoDAF, military architectural frameworks have been extended, resulting in several different versions. A short list includes MODAF (UK), NAF (NATO), AGATE (France), DNDAF (Canada), MDAF (Italy), and ADOAF (Australia). Each one adds to, redefines and/or clarifies the concepts, views, viewpoints and concerns contained within Military Architectural Frameworks, with the intention of improving procurement, planning, and implementation of military systems. However, supporting multiple and sometimes divergent frameworks leads to problems for industry, military organizations and tool vendors alike. In this age of globalization, mil-aero companies provide systems across the world to multiple governments. Often they must be specified in the local Architectural Framework creating extra overheads. Incompatible frameworks cause interoperability problems between governments because models cannot be exchanged. Interchange, even between modeling tools supporting the same framework, is difficult, if not impossible due to the different underlying implementations. Finally, having to support several constantly changing framework formats means that modeling tool vendors have a support nightmare. Figure 1 shows the evolution and relationships between DoDAF, MODAF, and NAF.

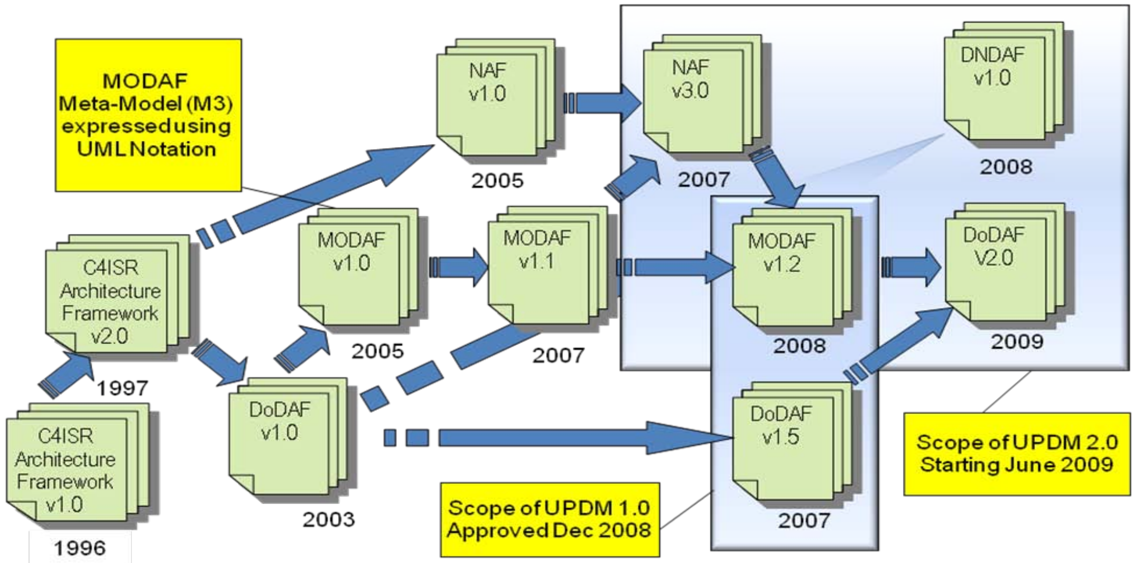
MODAF Meta-Model (M3) expressed using UML Notation
NAF v1.0
2005
NAF v3.0
2007
DNDAF v1.0
2008
C4ISR Architecture Framework v2.0
1997
MODAF v1.0
2005
MODAF v1.1
2007
MODAF v1.2
2008
DoDAF V2.0
2009
C4ISR Architecture Framework v1.0
1996
DoDAF v1.0
2003
DoDAF v1.5
2007
Scope of UPDM 1.0 Approved Dec 2008
Scope of UPDM 2.0 Starting June 2009

Figure 1: Evolution of Military Architectural Frameworks (Details Omitted)

The Unified Modeling Language (UML) and the recently created Systems Modeling Language (SysML) can be used as an underlying mechanism for all of these frameworks. This makes it feasible to work towards a standardized UML/SysML profile for these Military Architectural Frameworks. UML is a visual modeling language for software and can be extended to include new concepts using what is called a Profile. This provides a means to create and extend elements found in UML. SysML is an example of a UML Profile. SysML includes new concepts such as enhanced interface and flow specifications, system concepts, parametrics, integrated requirements and others. UML is currently widely used by architectural modelers and is referenced by many of the frameworks themselves. For example, DoDAF v1.5 Volume II provides guidance on using UML and the MODAF Meta-Model (M3) is expressed using UML Notation. (DoD 2003, DoD 2007a, DoD 2007b, DoD 2007c, HMSO 2002, and MOD 2008)

The UPDM Group. In March 2008, the UPDM Group was re-formed by members of INCOSE and the OMG to create the Unified Profile for DoDAF and MODAF (UPDM) using UML/SysML. (The previous submission was rejected by the DoD and MOD and was voted down by the OMG.) Members of the UPDM Group were tool vendors Adaptive, Artisan Software Tools, EmbeddedPlus, IBM, Mega, No Magic, Sparx, Visumpoint, members of industry ASMG, BAE Systems, Generic AB, General Dynamics, Lockheed Martin, Mitre, Northrop Grumman, Raytheon, Rolls Royce, Thales, Selex SI and representatives from the DoD, MOD, NATO, and Swedish armed forces. Members of the DoDAF 2.0 taskforce were heavily involved to ensure that

DoDAF 2.0 and UPDM converged as much as possible. Finally, members of the Canadian DND also participated. The DoD and MOD have officially issued a definitive statement of support for UPDM and this is available at www.omg.org and www.UPDM.com . Artisan, IBM and No Magic are co-Chairs of the UPDM Group. Through coordinated teamwork many of the challenges have already been overcome resulting in a specification that has been accepted by the OMG and is fully endorsed by both the DoD and MOD (OMG 2005, OMG 2009a). The finalized specification was accepted by the OMG in June 2009, and was officially issued by the OMG during the September 2009 meeting in San Antonio. Tools supporting UPDM are available now and projects are already using them. At the same meeting, a request for proposal for UPDM 2.0 was issued (OMG, 2009c). This will be discussed later in this paper. For further information on the UPDM development project and its use of Model-Based Systems Engineering see (Hause, 2009).

The goals of UPDM are to significantly enhance the quality, productivity, and effectiveness associated with enterprise and system of systems architecture modeling, promote architecture model reuse and maintainability, improve tool interoperability and communications between stakeholders, and reduce training impacts due to different tool implementations and semantics. Using the UML XML Metadata Interchange (XMI) interchange format, virtually all UML tools will be able to exchange models. The Model Interchange Working Group (MIWG) was recently formed by the OMG. The goal of the MIWG is to improve the interoperability of MOF/XMI-based tools. Their initial focus is on model interchange among UML, SysML, and UPDM -capable tools. More information can be found at http://www.omgwiki.org/model-interchange/doku.php. Standardization of model data and UML/SysML mapping means that both tool vendors and industry can provide models in a single format. Customized views can still be created, but they are based on core UPDM rather than requiring bespoke development. Finally, the UML/SysML foundation will improve the integration between architectural framework modeling and system modeling to support post acquisition life-cycle design and implementation.

What is UPDM? It is important to stress that UPDM is not a new Architectural Framework. Instead, UPDM 1.0 provides a consistent, standardized means to describe DoDAF 1.5 and MODAF 1.2 architectures in UML-based tools as well as a standard for interchange. UPDM, like DoDAF and MODAF is also process agnostic and it is also not a methodology. The rest of the paper will provide a brief overview of the development of UPDM, views unfamiliar to DoDAF 1.5 modelers, examples of the abilities UPDM provides to its implemented frameworks, our future goals, and further information on DoDAF 2.0.

Development of UPDM

Model-based engineering is at the heart of the Architectural Framework approach to modeling. A model of the system is created using different views to denote different stakeholder interests, and to provide a means for evaluation and report generation as well as to simplify maintenance. In the desire to “Walk our Talk”, UPDM was also developed using a model-driven approach; see Hause, (2009).

The Domain Meta-Model. In terms of the UPDM work process, a Domain Metamodel (DMM) was created using UML Class models to represent the concepts in DoDAF and MODAF. The DMM was the requirements model for UPDM, and traceability links between the DMM and the UPDM profile model were created. Concepts common to both DoDAF and MODAF were captured in a Core package, with DoDAF and MODAF packages also being created for their specific elements. The DMM concepts were then mapped to corresponding stereotypes in the

Profile which was analyzed and re-factored to reflect language architecture, tool implementation and reuse considerations. The conformance levels were finalized including mapping to SysML. Next, the Profile diagrams, stereotype descriptions, and documentation were added. Finally, the specification and XMI document were generated from the Profile model. This model based approach allowed the team to concentrate on architecture issues rather than documentation production, and issue the specification in record time. Consistency was automatically maintained by the UML tool. With every stereotype linked to the DMM element, the UML tool also enabled requirements traceability to be maintained between the Profile and the DMM. Figure 2 shows the All Views example of the Domain Meta-Model.

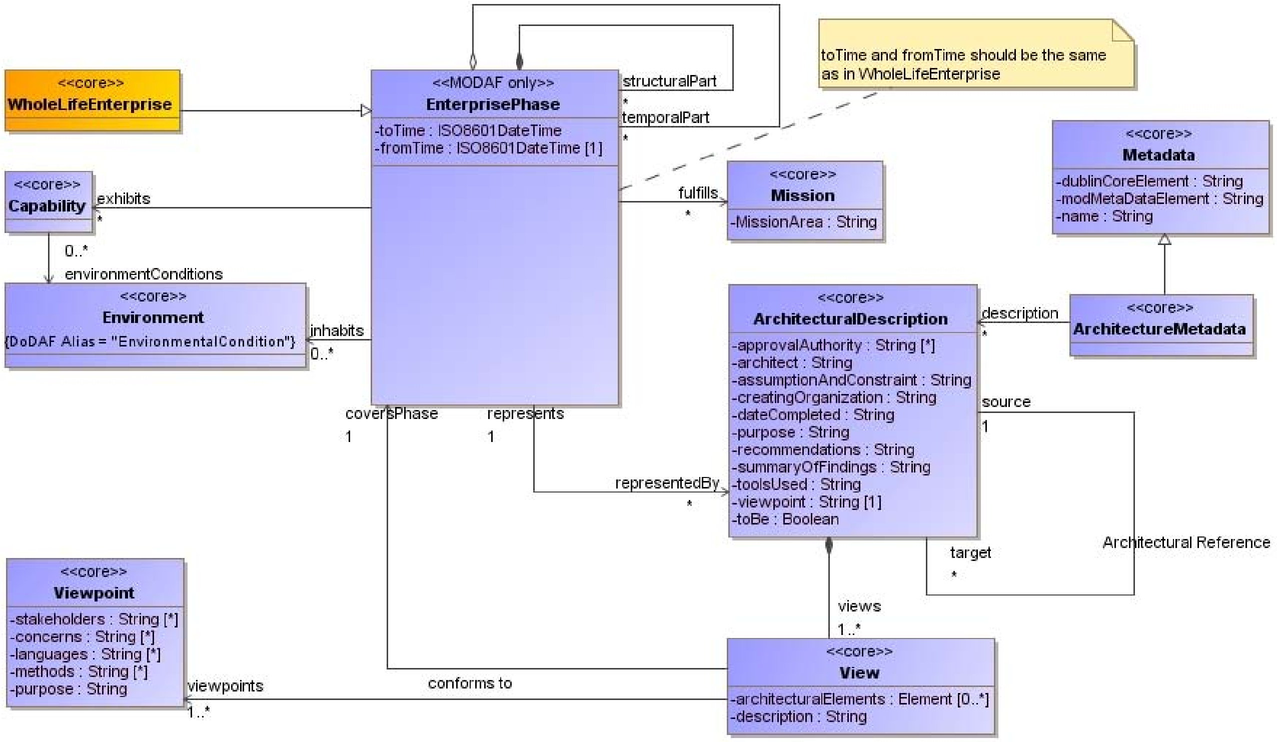

Figure 2: All Views Example of the Domain Meta-Model

UML is used to describe the Domain Meta-model. To fully understand the concepts, a basic understanding of UML is necessary. However, by “talking through” the diagram, the concepts can easily be understood. This example shows the Enterprise Phase of the model. The Enterprise Phase is made up of Structural and Physical Enterprise Phases, shown as the diamonds and connecting lines. The Whole Life Enterprise itself is a type of Enterprise Phase. Each Enterprise Phase fulfills zero or more Missions and exhibits zero or more Capabilities. The Enterprise Phase is represented by a set of Architectural Descriptions, containing one or more Views that conform to one or more Viewpoints. Time based Enterprise Phases provide the ability to model the “As Is” and desired architectures, as well as how the architecture will change over time.

The UPDM profile. As mentioned earlier, a UML profile provides a means of extending the UML. A UML profile is a coherent set of stereotypes, constraints, and tag definitions, defined for specific purposes (OMG 2007a). For example, if we group common requirements stereotypes together into a collection, we can create a requirements profile as was done in SysML. In addition, the appearance of the diagrammatic elements on the screen must be in a format familiar to modelers of military architectural frameworks. For example, on an OV-4 diagram, a role (called a Post in UPDM and is normally called a billet in DoD parlance) could be represented by a single stick man, whilst an organization could be shown as multiple overlapping stick men. However, modeling systems is more than just drawing pictures. It is also necessary to provide additional rules governing the creation of relationships between elements, interactions, multiplicity, cross diagram relationships, and the consequence of deleting elements that have relationships to others. This is called Domain Specific Modeling (DSM). Figure 3 below shows an example of the AV-1 stereotypes.

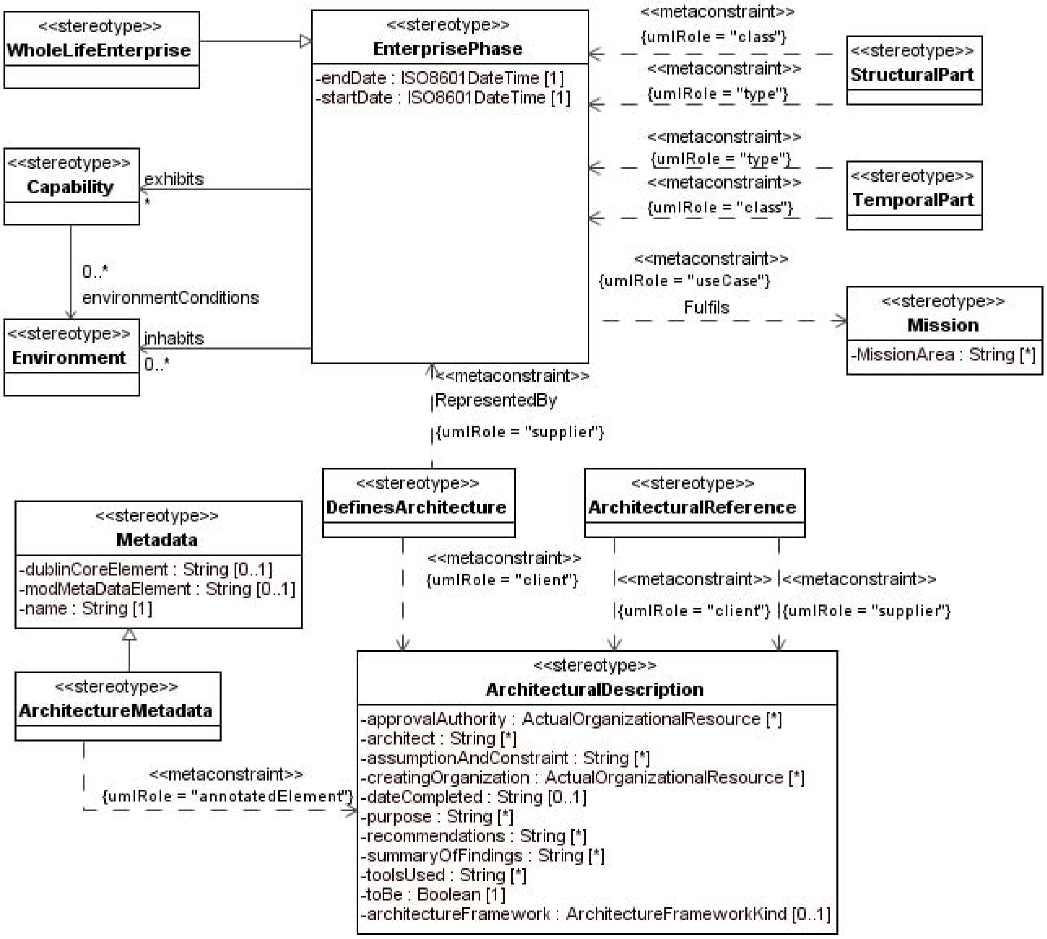

Figure 3: All Views Example of the UPDM Profile

This profile diagram expresses the same concepts as was shown in Figure 2. Walking through the diagram, the same description can be used. The Enterprise Phase is made up of Structural and Physical Enterprise Phases, shown as the diamonds and connecting lines. The Whole Life Enterprise itself is a type of Enterprise Phase. Each Enterprise Phase fulfills zero or more Missions and exhibits zero or more Capabilities. The Enterprise Phase is represented by a set of Architectural Descriptions, containing one or more Views that conform to one or more Viewpoints. However, this diagram expresses the concepts in a way that can be used by UML meta-modelers to implement a UPDM profile.

A model editor that ensures that a single diagram is consistent, correct and complete, but does not enforce these same rules across the entire model is worse than useless. As soon as the diagrams start to be updated, it will become extremely difficult to maintain consistency. In addition, quantitative analyses of elements in the model are not possible. In order for Domain Specific Modeling (DSM) to work effectively, the diagram editor must be built using an integrated model meta-model. Using a simple UML example, it should ensure that if class A is defined as a parent of class B in one diagram, it cannot be defined as a child of class B on another diagram. Whilst this simple example illustrates the point, the rules and relationships necessary for a correct AF model are far more onerous. For example, one could specify that for an AF model, an operational activity defined as a child activity, can only be resident on child operational nodes of the parent node to which the parent activity is assigned. Accordingly, consistency on a single diagram is not sufficient. Consistency must be maintained across the complete model. Finally, because of the interconnections between the different views, the model should reside in an integrated database rather than individual files representing the different views. The creation of an architectural framework involves many different architects working at different levels simultaneously. A file based editor that locks a set of file based views to a single user will quickly cause model lock-up. In addition, it is essential that the latest model information is available to all the architects or they will quickly become out of synch with one another.

Architectural Frameworks used in UPDM

The core views in DoDAF - All Views, Operational, Systems, and Technical - have been used successfully to define military architectures for some time now. However, system architects found that they did not go far enough. Although these views are aimed at getting the “big picture” and were sufficient for managing large projects, practitioners found that the viewpoints provided by DoDAF were not “big enough” to properly counter the issue of “stovepipe development”. This is where military procurements are developed in isolation from each other rather than in a coordinated manner resulting in the creation of incompatible and redundant systems, resulting in higher development costs, unnecessary expenditures, and inefficient military operating procedures. One example of this was a ground support helicopter that was deployed with a communication system that was incompatible with the ground troop’s radios. This meant that all communication had to be routed through the command base causing critical communication delays. They also found that DoDAF lacked the breadth necessary for effective program management, where the goal is to specify multiple projects in order to develop compatible capabilities.

MODAF kept compatibility with the core DoDAF viewpoints in order to facilitate interpretation of architectural information with the US. However, MODAF v1.0 added two new viewpoints. The new elements were the Strategic and Acquisition Viewpoints. These were added to better contribute to MOD processes and life-cycles, specifically the analysis of the strategic issues and dependencies across the entire portfolio of available military capabilities within a given time frame. In MODAF v1.2, Service views were added to support the development of Service Orientated Architectures (SOA). These were based on NAF 3.0. In the same way that the DoDAF views are integrated, MODAF views are as well. For example, the acquisition views specify when the capabilities defined within the strategic views will become available. Capabilities can be associated with capability configurations that define the systems, organizations and people necessary to achieve the capability. Detailing all of the new views is a task for a book, and not a paper. Consequently, this section will provide an overview of views unfamiliar to DoDAF 1.5

modelers. The example used in the UPDM specification was for a Search and Rescue (SAR) set of capabilities. Example SARs are mountain SAR, maritime SAR, battlefield SAR, etc. For further information, see the example model appendix in OMG (2009).

Many of these concepts were developed by the IDEAS group. IDEAS is the International Defence Enterprise Architecture Specification for exchange. The IDEAS Group was set up in 2005 to examine the issue of interoperability of Enterprise Architecture Data. The group consists of subject matter experts on the Australian, Canadian, UK and USA defense architecture frameworks. These experts have been working together in the IDEAS Group to define a common information structure for data exchange. For further information on IDEAS, see the IDEAS website - www.ideasgroup.org . The foundation model (published in April 09) is at http://www.ideasgroup.org/foundation/ .

Capability/Strategic View

A capability is the ability or capacity to achieve specific objectives. Examples include Search and Rescue, effects delivery, transportation, etc. The Strategic View provides a high level view of the enterprise capabilities and their relationships. This enables Capability Management, for example, capability introduction, integration, re-alignment and removal. A single Strategic View can be defined that will have a number of Architecture Descriptions. Each Architecture Description may then have multiple Operational, System, Technical Standards, and All Views. These provide a complete picture of the capability strategy. UPDM comprises six Strategic Views, which are detailed below.

The StV-1 Enterprise Vision defines Enterprise Goals and Visions relating to a time-based Enterprise Phase. It outlines the goals and vision for a capability area over a specified period of time, denoted by the Enterprise Phases. It also describes how high level goals and strategy are to be delivered in terms of capability. The StV-2 Capability Taxonomy defines capabilities for current and future enterprises in a hierarchy and the environmental conditions associated with the different capabilities. The StV-3 Capability Phasing view shows when capabilities will be available and/or de-commissioned over specific periods of time and how they relate to projects. StV-4 Capability Dependencies describes the capabilities in logical groups and the dependencies between the capabilities. StV-5 Capability to Organization Deployment Mapping shows how capabilities map to organizations and the capability configurations and resources that will achieve the capability. Finally, StV-6 Operational Activity to Capability Mapping shows which Operational Activities map to which capabilities.

Acquisition/Project View

The Acquisition View describes project details and dependencies between projects and capability integration. This helps to guide the acquisition and deployment processes. The AcV-1 Acquisition Clusters View enables the user to model the organizations and projects. It shows the dependencies between the actual organizations that own projects. The AcV-2 Program Timelines View defines project timelines and their relationship to Capability Configurations. It supports acquisition and deployment including the management of dependencies between projects and the integration of the Defense Lines of Development (DLOD) (called DOTMLPF in the US DoD) to achieve a successfully integrated military capability. Traditional DLODs are training, equipment, personnel, information, concepts & doctrine, organization, infrastructure, and logistics. Originally MODAF was developed to support a fixed number of DLODs but has been updated to support different combinations of DLODs (as they get updated and different organizations have different DLODs).

Service-Oriented View

The Service-Orientated View is a description of services needed to directly support the operational domain as described in the Operational View. A service is described as a unit of work through which a particular Resource provides a useful result to a consuming Resource. UPDM services may include standard web-based services, but also define effects deployment, logistics support, and even cooking meals for hungry soldiers. The resource provides the service, and the consuming resource makes use of it. The Services Views are the following:

- SOV-1 Service Taxonomy – this describes services in a generalization hierarchy, showing services that are types of other services.
- SOV-2 Service Interface Specification - describes the provided and required interfaces for services i.e. what they will do and what they need.
- SOV-3 Capability to Service Mapping - shows how services support capabilities.
- SOV-4a Service Constraints, SOV-4b Service State Model, and SOV-4c Service Interaction Specification describe service policies, state based behavior and interactions for a service in general.
- SOV-5 Service Functionality - describes functions and operations that the service will perform.

Leveraging SysML

OMG SysML is a visual modeling language that extends UML 2 in order to support the specification, analysis, design, verification and validation of complex systems that include components for hardware, software, data, personnel, procedures and facilities. SysML is intended to be used with different methodologies including structured analysis, object orientation and others. OMG SysML reuses a subset of UML 2 concepts and diagrams and augments them with some new diagrams and constructs appropriate for systems modeling. In particular, the language provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis models.

SysML Elements. The «block» is the basic unit of structure in SysML and can be used to represent hardware, software, facilities, personnel, data, or any other system element. The system structure is represented by block definition diagrams and internal block diagrams. A block definition diagram describes the system hierarchy and system/component classifications. The internal block diagram describes the internal structure of a system in terms of its parts, ports, and connectors. The package diagram is used to organize the model. The behavior diagrams include the use case diagram, activity diagram, sequence diagram, and state machine diagram. SysML includes a graphical construct to represent text based requirements and relate them to other model elements. The requirement diagram provides a bridge between the typical requirements management tools and the system models. The parametric diagram represents constraints on system property values such as performance, reliability, and mass properties, and serves as a means to integrate the specification and design models with engineering analysis models. SysML also includes an allocation relationship to represent various types of allocation, including allocation of functions to components, logical to physical components, software to hardware, and workflow.

Parametric Diagrams. Parametric diagrams are used to describe constraints on system properties to support engineering analysis. In order to support this type of modelling a ConstraintBlock has been introduced into OMG SysML. A ConstraintBlock defines a set of parameters and one or more constraints on the parameters. By default, these parameters are non-directional and so have no notion of causality. These ConstraintBlocks are used in a parametric diagram to constrain system properties. ConstraintBlocks may be used to express mathematical equations such as ‘F=m•a’ and ‘a = δv/δt’, or statistical values and utility functions such as might be used in trade studies. Based on the reusable concept of a block new ConstraintBlocks can be built by reusing more primitive ConstraintBlocks such as basic mathematical operators. SysML also defines a model of value types that can have units and dimensions and probability distributions. The value types are used to type properties of blocks. The Parametric Diagram is a specialized variant of an internal block diagram that restricts diagram elements to represent constraint blocks, their parameters and the block properties that they bind to. Both parameters and properties may be represented as small “pin-like” boxes to help make the diagrams more scalable. This section will show how the different elements of SysML can be leveraged to provide a mechanism for trade-off analysis, concentrating on the parametric diagram. Also see (Hause, Thom 2005).

UPDM provides a means of defining and using measurable quantities. These include:

- measurementTypes : MeasurementSet[*] - Types of measurements corresponding to the actual measurements.

- actualMeasurements : ActualMeasurementSet[*] - The actual measurements to which the element must conform.

Figure 4 shows an example of the measurement sets for the SAR model.

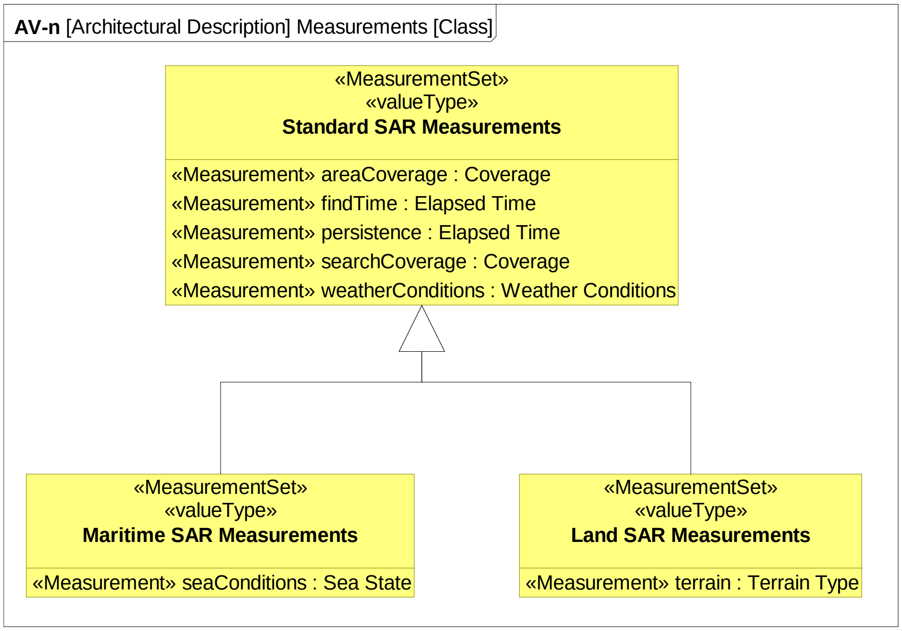
AV-n [Architectural Description] Measurements [Class]
«MeasurementSet»
«valueType»
Standard SAR Measurements
«Measurement» areaCoverage : Coverage
«Measurement» findTime : Elapsed Time
«Measurement» persistence : Elapsed Time
«Measurement» searchCoverage : Coverage
«Measurement» weatherConditions : Weather Conditions
«MeasurementSet»
«valueType»
Maritime SAR Measurements
«Measurement» seaConditions : Sea State
«MeasurementSet»
«valueType»
Land SAR Measurements
«Measurement» terrain : Terrain Type

Figure 4: Measurement Sets for the SAR

The Standard SAR Measurements describe metrics such as area coverage, find time, etc. The use of inheritance provides the ability to specialize the measurements as either Maritime or Land SAR. Figure 5 shows a SysML Block Definition Diagram (BDD) the Unit, Dimension, and Value Type Definitions used by the measurement sets.

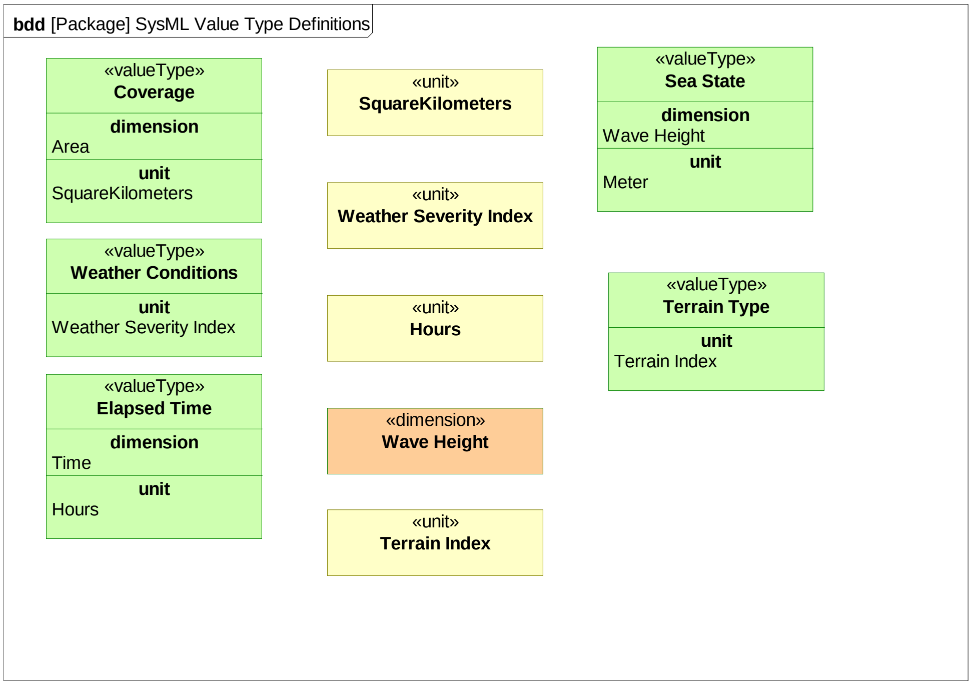

Figure 5: Unit, Dimension and Value Type Definitions for SAR

During any requirements development process it is essential to specify quantifiable values. It is not enough to specify that SAR should be implemented. It is also necessary to state required values for the area coverage, find time, etc in order to find the right solution, and to verify and validate that the solution was found. Figure 6 shows the actual and required values for maritime SAR for a search and rescue enterprise development.

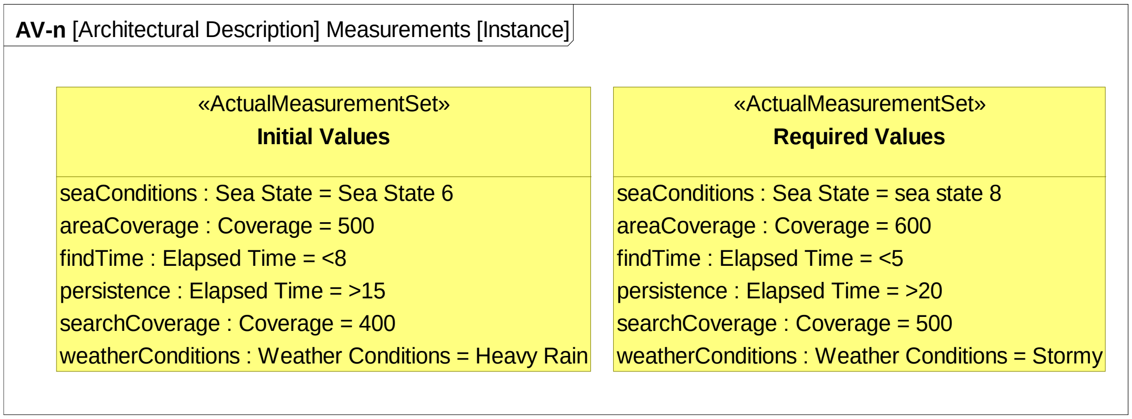
AV-n [Architectural Description] Measurements [Instance]
«ActualMeasurementSet»
Initial Values
seaConditions : Sea State = Sea State 6
areaCoverage : Coverage = 500
findTime : Elapsed Time = <8
persistence : Elapsed Time = >15
searchCoverage : Coverage = 400
weatherConditions : Weather Conditions = Heavy Rain
«ActualMeasurementSet»
Required Values
seaConditions : Sea State = sea state 8
areaCoverage : Coverage = 600
findTime : Elapsed Time = <5
persistence : Elapsed Time = >20
searchCoverage : Coverage = 500
weatherConditions : Weather Conditions = Stormy

Figure 6: Actual and Required Values

Additionally, it is useful to be able to make use of these metrics in the modeling stage to perform trade-off analysis. The primary tool for doing this in SysML is the parametric diagram. This provides a means of connecting the various measurements corresponding to a capability configuration in order to determine if it is fit for purpose. Figure 7 shows a parametric diagram linking the metrics to determine search performance.

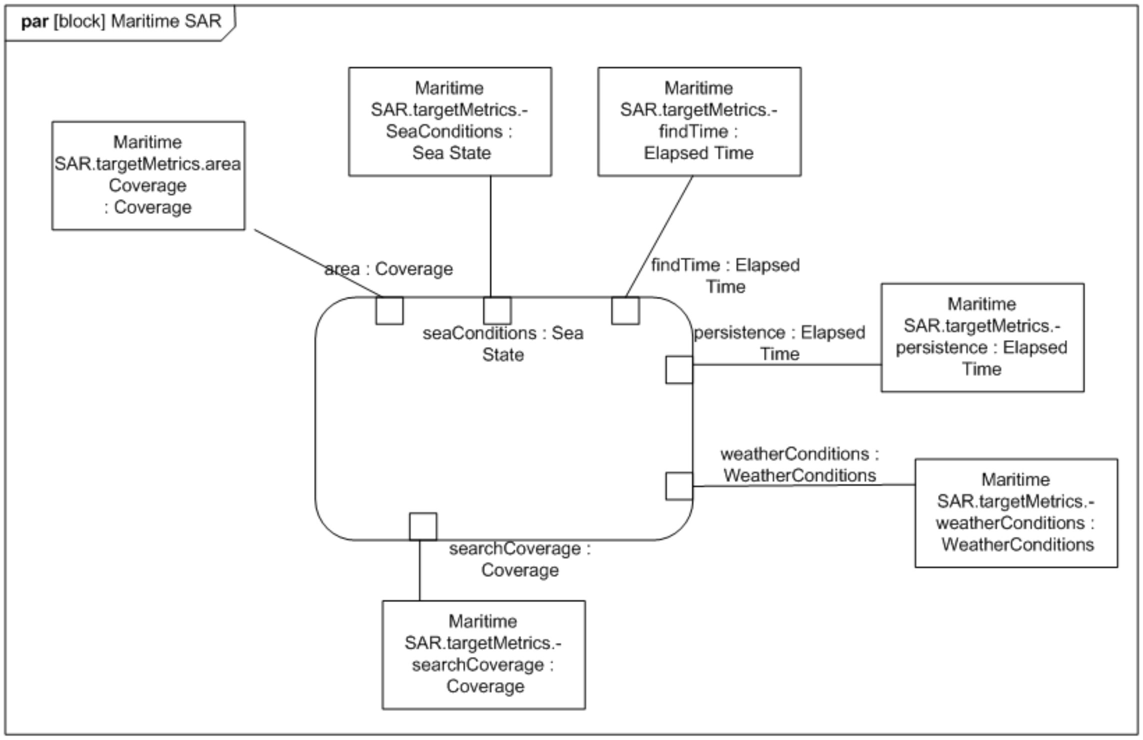
par [block] Maritime SAR
Maritime SAR.targetMetrics.area Coverage : Coverage
Maritime SAR.targetMetrics.-SeaConditions : Sea State
Maritime SAR.targetMetrics.-findTime : Elapsed Time
area : Coverage
findTime : Elapsed Time
seaConditions : Sea State
persistence : Elapsed Time
Maritime SAR.targetMetrics.-persistence : Elapsed Time
weatherConditions : WeatherConditions
Maritime SAR.targetMetrics.-weatherConditions : WeatherConditions
searchCoverage : Coverage
Maritime SAR.targetMetrics.-searchCoverage : Coverage

Figure 7: Parametric Diagram for the SAR

The parametric diagram can be interfaced to a problem solving tool such as Matlab or Modelica to determine if a particular configuration has complied with the metrics, or to determine what the optimum values for a configuration would be. This is an example of how Model Based Systems Engineering can be used to save time and money during systems specification and development. For further information on SysML see Hause (2006), OMG (2007b), Friedenthal (2008), Holt, (2008), and Korff (2008).

Compliance and Compatibility

One of the goals of UPDM was to reuse existing standards as much as possible. As there are defined and emerging standards for concepts such as services, views, viewpoints, models, etc, the group decided it was counter-productive to redefine them. As we stated earlier, integration with SysML was considered by many of the group to be key to a successful outcome. However, there was an equally strong point of view that a UML only solution was necessary. Consequently, two levels of compliance were defined for UPDM, namely L0, which uses UML only, and L1, which uses SysML. Tool vendors can choose which version to implement and interchange protocols between them have been defined.

Reuse of existing specifications

UPDM reuses UML/SysML wherever practical to satisfy the requirements of the RFP and leverage features from both UML and SysML to provide a robust modeling capability. Consequently, UPDM is intended to be relatively easy to implement for vendors who support

UML 2. The UPDM team was able to reuse UPMS. UPMS was formally adopted and reuse of the UPMS concepts was integrated into UPDM. UPMS was renamed to SoaML, (Service Oriented Architectures Modeling Language.) For more information see OMG (2009b).

Future direction of UPDM

The UPDM specification 1.0 was delivered to the OMG on the 25th of August 2008, and was voted on and accepted during the OMG September meeting. Comments from OMG members and the public in general were then sent to the OMG. These comments were addressed and the specification was voted on during the OMG December meeting. The Finalization Task Force then addressed problems found during the evaluation period. After this work was complete, it was voted on to be an adopted specification at the June 2009 meeting. Tool vendors, (including Artisan Software Tools), have already implemented UPDM in their tools. These are now being used on development projects and bids in both North America and Europe.

However, the work of the UPDM Group did not stop with DoDAF 1.5 and MODAF 1.2. A new initiative was launched as soon as it was complete to cover other Architectural Frameworks, as well as other areas of interest. DoDAF v2.0 was released in June, 2009 and a new version of UPDM is being created to maintain exchange compliance. The NATO Architectural Framework (NAF), which is very similar to MODAF v1.2, will also be addressed. In fact, it has been announced that the proposed version of NAF 3.1 will adopt all the MODAF 1.2 concepts. Finally, the Security Views in the Canadian DNDAF will be included. Other areas being considered are Human Factors views (Bruseberg, 2007), Business Motivational Modeling, and Business Process Modeling. The work of the UPDM Group will continue to address updates to architectural frameworks as and when they occur. This will ensure that system architects will continue to have standards based modeling tools at their disposal. It will also ensure that interchange between the frameworks will continue to be supported. For further information see (OMG, 2009c). For further information on UPDM, visit the UPDM website: www.UPDMG.com and the OMG website: www.OMG.org .

DoDAF 2.0

DoDAF 2.0 differs from previous versions of DoDAF and MODAF in that it has fully adopted the foundation elements found in IDEAS, which was discussed earlier. It is worth describing these here and providing an example of the DoDAF 2.0 DM2 taken from the DoDAF 2.0 specification. Figure 8 shows the top-level foundation elements. The domain concepts inherit several important properties. None of these foundation properties are unusual; they are all used in reasoning everyday:

- Individuals, things that exist in 3D space and time, i.e., have spatial-temporal extent.
- Types, sets of things.
- Tuples, ordered relations between things, e.g., ordered pairs in 2D analytic geometry, rows in relational database tables, and subject-verb-object triples in the DoDAF 2.0 Resource Description Framework.
- Whole-part; e.g., components of a service or system, parts of the data, materiel parts, subdivisions of an activity, and elements of a measure.

- Temporal whole-part; e.g., the states or phases of a performer, the increments of a capability or projects, the sequence of a process (activity).
- Super-subtype; e.g., a type of system or service, capability, materiel, organization, or condition.
- Interface; e.g., an overlap between two things.
- Three types of Things: Types (which are like sets), Tuples (ordered relationships), and Individuals (not persons, but Things that have spatial and temporal extent – spatio-temporal extent.)

Figure 8: IDEAS Top-Level Foundation Elements

Additionally, it is useful to show an example of how these concepts are used to describe elements in the DoDAF 2.0 DM2. Figure 9 shows the desired effect structure. It describes the effect that a capability is meant to exhibit. It is recommended that the reader prints out a large copy of figure 9 in order to help understanding. The key concepts are the capability, performer, activity and measure. A capability is the ability to produce a desired effect. This is done by performing activities. Performers perform activities. This will produce a desired effect on one or more resources, normally to a measureable quantity. Activities can also cause exchanges of data, energy and material to take place between performers. By expressing the concepts in this way, there is a direct connection between the capability, the effect the realization of the capability, and the performers and activities involves. In addition, this set of concepts is available at both the operational and systems level. This allows for the normal OV and SV sets of views as well as additional sets of abstractions. The UPDM team are mapping these concepts to the UPDM DMM to ensure backwards compatibility to UPDM 1.0 and to ensure that interchange between DoDAF and MODAF models will still be possible.

Figure 9: Desired Effect Structure

The paradigm shift to the ontological basis of the meta-model will have a dramatic effect on the modeling of architectural frameworks and will help to support a data-centric approach to modeling which will truly enable MBSE for architectures.

References

Bruseberg A & Lintern G (2007) “Human Factors Integration for MODAF: Needs and Solution Approaches”. In Proc. of INCOSE 2007, San Diego, CA, USA, 24 - 28 June 2007

DoD, 2003, Architecture Framework Version 1.0 Volume I: Definitions and Guidelines 30 August 2003

DoD, 2007a, Architecture Framework Version 1.5, Volume I: Definitions and Guidelines, 23 April 2007, DoD Architecture Framework Working Group

DoD 2007b, Architecture Framework Version 1.5, Volume II: Product Descriptions, 23 April 2007, DoD Architecture Framework Working Group

DoD 2007c, Architecture Framework Version 1.5, Volume III: Product Descriptions, 23 April 2007, DoD Architecture Framework Working Group

Friedenthal, S., Moore, A., Steiner, R. Practical Guide to SysML: The Systems Modeling Language, Morgan Kaufman September 2008

Hause, M.C. and Thom, F. (2005), MoDAF and SysML – A Winning Combination, May 2005, INCOSE UK Chapter – Spring Symposium 2005 Proceedings.

Hause, M.C., 2006, The Systems Modeling Language - SysML, Sept 2006, INCOSE EuSEC

Symposium 2006 Proceedings.

Hause, M.C., 2009, The UPDM RFC Development Project - An Exercise in Model-Based Virtual Team Development or “Practicing What We Preach”, 5th Annual Israeli National Conference on Systems Engineering proceedings, Herzlia, Israel

HMSO, 2002, The Strategic Defence Review White Paper (Cm 3999 HMSO) & SDR: A New Chapter(Cm 5566 HMSO). (www.hmso.gov.uk)

Holt, J., Simon Perry, S., SysML for Systems Engineering, IET Publications, 2008

Korff, A., Modellierung von eingebetteten Systemen mit UML und SysML, von Spektrum Akademischer Verlag Taschenbuch - 13. June 2008

MOD Architectural Framework, Version 1.2, 23 June 2008, Office of Public Sector Information, http://www.modaf.org.uk/

Object Management Group (OMG), 2005, Military Architecture Framework Request for Information, Available from www.omg.org. [Accessed April, 2005]

Object Management Group (OMG), 2007a. Unified Modeling Language: Superstructure version 2.1.1 with change bars ptc/2007-02-03. [online] Available from: http://www.omg.org [Accessed September 2007].

Object Management Group (OMG), 2007b, OMG Systems Modeling Language (OMG SysML™), V1.0, OMG Document Number: formal/2007-09-01, URL: http://www.omg.org/spec/SysML/1.0/PDF , Accessed November, 2007

Object Management Group (OMG), 2009a, Unified Profile for the Department of Defense Architecture Framework (DoDAF) and the Ministry of Defence Architecture Framework (MODAF), available at http://www.omg.org/spec/UPDM/

Object Management Group (OMG), 2009b, Service oriented architecture Modeling Language (SoaML), available at http://www.omg.org/spec/SoaML/

Object Management Group (OMG), 2009c, Unified Profile for DoDAF/MODAF (UPDM) 2.0 Request For Proposal), available at http://www.omg.org/cgi-bin/doc?c4i/2009-09-02

U.S. Department of Defense. 2003. Department of Defense Directive 5000.1: The Defense Acquisition System. Washington, DC: Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics.

BIOGRAPHY

Matthew Hause is Artisan’s Chief Consulting Engineer, the co-chair of the UPDM group and a member of the OMG SysML specification team. He has been developing real-time systems for over 30 years. He started out working in the Power Systems Industry, and has been involved in Process Control, Communications, SCADA, Distributed Control, military systems and many other areas of systems development. His roles have varied from project manager to developer. His role at Artisan includes mentoring, sales presentations, standards development and training courses. He has written a series of white papers on project management, Systems Engineering, architectural modeling and systems development with UML, SysML and Architectural Frameworks. He has been a regular presenter at INCOSE, the IEEE, BCS, the IET and other conferences. Matthew studied Electrical Engineering at the UNM and Computer Science at the University of Houston.