Architecting in the Fourth Dimension Temporal Aspects of DoDAF

Matthew Hause

Atego

Matthew.Hause@Atego.com

Lars-Olof Kihlstrom

Syntell AB

Lars-Olof.Kihlstrom@Syntell.se

Abstract. The capture of system structure, behavior, configuration, interaction, and compliance is common practice in architectures. These are largely static views showing a specific configuration or behavior. IEEE Std 610.12−1990 defines architecture as "the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution." (IEEE, 1990) Modeling this evolution or the temporal aspects in architecture frameworks such as the Department of Defense Architecture Framework (DoDAF) was previously problematic. With the release of DoDAF version 2.0, architectures can now take the fourth dimension (Time) into account. The challenge is to identify areas of architecture where time can be modeled and how to take best advantage of it. Also problematic is how to express these concepts without having to expose all the internal ontological relationships upon which DoDAF is built. The Unified Profile for DoDAF and MODAF (UPDM) delivers an implementation of DoDAF 2.0 that provides a clear and concise way of expressing these concepts without requiring the user to become an expert in the DoDAF 2.0 "internal wiring" and detailed ontological concepts. This paper will examine the temporal concepts defined in DoDAF 2.0 and show how time can be effectively integrated into a model to express essential temporal concepts.

Introduction

As William Shakespeare wrote in Julius Cesar, "Timing is everything." (Shakespeare, 1613) Cummings (1922) noted humorously that "Time is what keeps everything from happening at once." Philosophers have also mused on the concepts and flow of time for as long as human beings have roamed the planet. Time is no less important when building a military architecture framework. It is not sufficient to simply model the system configurations. It is necessary to show how a configuration will evolve over time, how the variations will differ, common components, additional and emergent behavior, how a systems behavior and capabilities change over time, etc. Some examples of the use of time are:

- Modeling a sequence of events
- Showing how a system changes over time
- Showing how the use of a system can change over time
- Capability modeling and how different systems support a capability over time
- Showing how a system supports multiple capabilities at different phases of its lifecycle
- Modeling system states to show time dependent behavior
- Time dependent activity sequences
- Modeling processing time, latency, transport time etc.

- Scheduling deployment of systems over time
- Personnel deployment and competency assessment
- Data management lifecycles
- Integrating system acquisition cost, deployment cost etc. to show total cost of ownership.
- Modeling product variants
- Showing cost vs. time vs. capability
- Etc.

A Brief History of DoDAF

Arguably, the most widely used military frameworks are the US Department of Defense (DoD) Architecture Framework (DoDAF), the British Ministry of Defence (MOD) Architecture Framework (MODAF) and the NATO Architecture Framework, (NAF). 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 was developed in the 1990s as the C4ISR architectural architecture framework. C4ISR v1.0 was released 7 June 1996, and was created in response to the passage of the Clinger-Cohen Act. It addressed the 1995 Deputy Secretary of Defense directive that a DoD-wide effort be undertaken to define and develop a better means and process for ensuring that C4ISR capabilities were interoperable and met the needs of the warfighter. C4ISR Architecture Framework v2.0 was released in December 1997.

DoDAF Versions. DoDAF v1.0 was released in August 2003. It broadened the applicability of architecture tenets and practices to all Mission Areas rather than just the C4ISR community. This document addressed usage, integrated architectures, DoD and Federal policies, value of architectures, architecture measures, DoD decision support processes, development techniques, and analytical techniques. The data format was expressed as CADM v1.01. (DoD, 2003) This was the start of the data-centric approach and placed emphasis on architecture data elements that comprise architecture products. DoDAF Version 1.5 was released in April 2007 as an update towards the integration of IDEAS concepts (discussed later). (DoD, 2007a, 2007b, 2007c) On May 28, 2009 DoDAF v2.0 was approved by the Department of Defense. (DoDAF, 2010)

DoDAF Views. DoDAF contains four basic views: the overarching All Views (AV), Operational View (OV), Systems View (SV), and the Technical Standards View (TV/StdV). 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 FEMA. The goal is to improve planning, organization, procurement and management of these complex organizations. All major DoD weapons and information technology system procurements are now required to document their enterprise architectures using DoDAF.

Evolution of MODAF/NAF. MODAF kept compatibility with the core DoDAF viewpoints in order to facilitate interpretation of architectural information with the US military. However, MODAF v1.0 added two new viewpoints. The new elements were the Strategic and Acquisition Viewpoints called the Capability and Project Views in DoDAF 2.0. These were added to better contribute to MOD processes and lifecycles, 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 and have been included in DoDAF 2.0. In the same way that the existing views are integrated, the new views are as well. For example, capabilities can be associated with systems that define the subsystems, organizations and people necessary to achieve required capabilities. The Project views specify when the systems supporting the capabilities will be available, thereby demonstrating when the capabilities will become available. (MOD, 2008)

The DM2. DoDAF has a meta-model underpinning the framework, defining the types of modeling elements that can be used in each view and the relationships between them. DoDAF versions 1.0 thru 1.5 used the CADM meta-model, which was defined in IDEF1X (then later in UML) with an XML Schema derived from the resulting relational database. From version 2.0, DoDAF has adopted the IDEAS Group foundation ontology as the basis for its new meta-model. This new meta-model is called "DM2"; an acronym for "DoDAF Meta-Model".

IDEAS

IDEAS is the International Defense Enterprise Architecture Specification for exchange. DoDAF version 2.0 is based on the IDEAS ontology foundation. (IDEAS, 2012) The current versions of NAF and MODAF are influenced by IDEAS to some degree but are still UML profiles. An update to MODAF called MODEM has however been prepared which is based entirely on the IDEAS Foundation ontology. The purpose of IDEAS is to develop a data exchange format for military Enterprise Architectures. This goal is to provide seamless sharing of architectures between the partner nations regardless of which modeling tool or repository they use. The initial scope for exchange is the architectural data required to support coalition operations planning:

- Systems – communications systems, networks, software applications, etc.
- Communications links between systems.
- Information specifications – the types of information (and their security classifications) that the communications architecture will handle.
- Platforms & facilities.
- System & operational functions (activities).
- People & organizations.
- Architecture meta-data – who owns it, who was the architect, name, version, description, etc.

Before going further, it would be best to explain a few of the main concepts in the IDEAS foundation key objects .At the base of the IDEAS ontology is the “Thing”. There are 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.)

Mereology is a collection of axiomatic first-order theories dealing with parts and their respective wholes. In contrast to set theory, which takes the set–member relationship as fundamental, the core notion of mereology is the part–whole relationship. Mereology is both an application of predicate logic and a branch of formal ontology. For further information see IDEAS (2012). The IDEAS foundation key objects are shown in Figure 1.

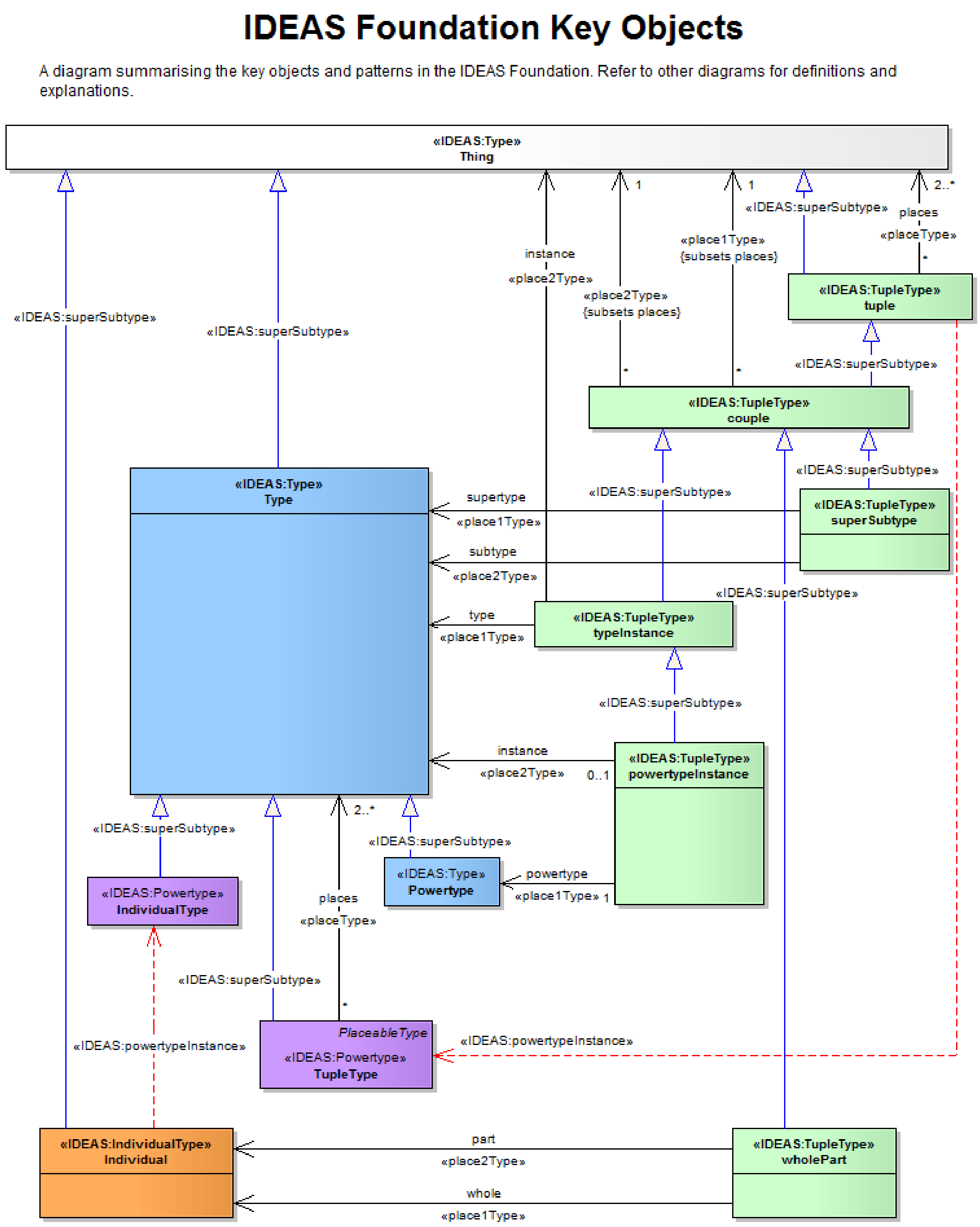

IDEAS Foundation Key Objects
A diagram summarising the key objects and patterns in the IDEAS Foundation. Refer to other diagrams for definitions and explanations.
«IDEAS:Type» Thing
«IDEAS:superSubtype»
instance
«place2Type»
«place2Type» {subsets places}
«place1Type» {subsets places}
places
«placeType»
2..*
1
«IDEAS:TupleType» tuple
«IDEAS:TupleType» couple
«IDEAS:Type» Type
supertype
«place1Type»
subtype
«place2Type»
«IDEAS:TupleType» superSubtype
type
«IDEAS:TupleType» typeInstance
«IDEAS:TupleType» powertypeInstance
0..1
«IDEAS:Type» Powertype
powertype
«IDEAS:Powertype» IndividualType
PlaceableType
«IDEAS:Powertype» TupleType
«IDEAS:powertypeInstance»
«IDEAS:IndividualType» Individual
part
whole
«IDEAS:TupleType» wholePart

Figure 1. IDEAS Foundation Key Objects. (IDEAS, 2012)

Foundation Objects. None of these foundation properties found in Figure 1 are unusual; they are all used in everyday reasoning:

- 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 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.

Higher level Objects. These can then be used together to model more complex concepts. Some of these are found as higher order concepts in IDEAS. Others are concepts found in DoDAF, MODAF and NAF.

- BeforeAfter (IDEAS foundation element)
- BeforeAfterType (IDEAS foundation element)
- TemporalWholePart (IDEAS foundation element)
- TemporalWholePartType (IDEAS foundation element)
- Desired Effect (DM2 element)
- Work Streams
- Project activity sequence
- State modeling
- Milestones
- Etc.

There are a number of items above that are pure IDEAS elements. The fact that they are allowed for direct use in DM2 actually represents a problem since this places an awful lot of responsibility on the modeler. DesiredEffect is a relationship between a kind of PerformerCapableOfResponsibility and a kind of resource. The time issue here is somewhat indirect. It is assumed that the connection is to a future state of the resource but the DM2 model does not make this explicit. Although Milestones are part of MODAF, they are not part of the DM2 vocabulary. The closest that DM2 gets to this is by assuming that a kind of project contains a kind of activity that is assumed to be a milestone by the modeler. In order to fully demonstrate this, an example of several of the elements above are contained in Figure 2 We will then go on to further explain the concepts using some models and portions of models taken from the DM2 meta-model.

A Project Based Example Using the DM2

A simple project schedule diagram has been created to help explain some of these concepts. Figure 2, apart from being unreadable looks extremely complicated. However, it is actually almost completely an explicit DoDAF 2 PV-2 model. It contains almost all of the DoDAF defined necessary elements for a PV-2 model. (There are some missing elements mostly associated with measurements of various kinds). It also contains some elements defined as optional for the view and some that a modeler is actually not allowed to use: notably Individual and IndividualType. These are used here since to exclude them would make it difficult to see where different elements point to. Figures 3, 4, and 5 detail Figure 2 in a more legible format.

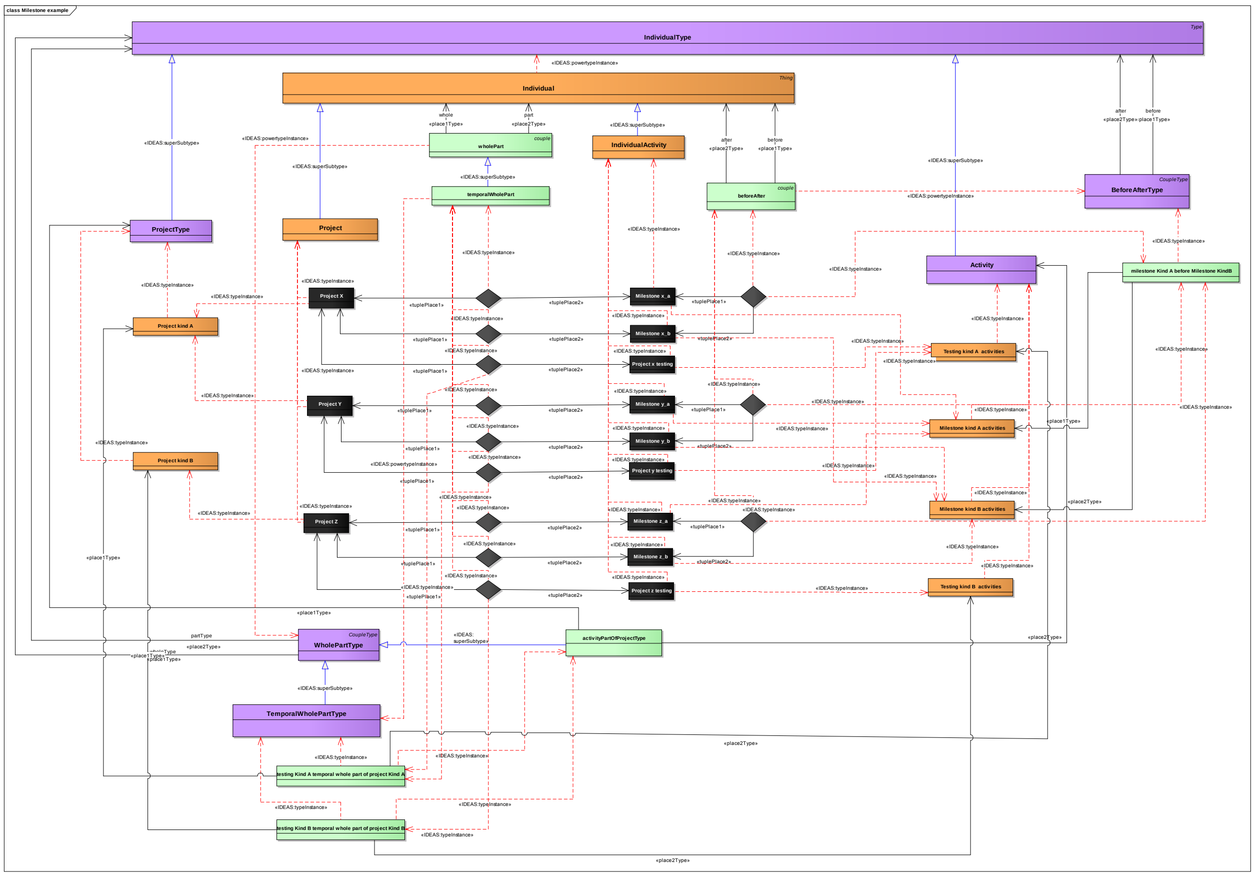


Figure 2. An Example of Temporal Usage Within DM2

Figure 2 actually demonstrates a fairly large number of the temporal aspects of DoDAF 2 and it may therefore be of interest to look at them in slightly more detail. A set of Individual projects are contained in the example model and a set of example activities. Since milestones are not a part of the DoDAF vocabulary, activities have been chosen instead and there are a few different individual milestones as well as a completely different type of activity (testing) associated with each individual project. Figure 3 shows project X with three different individual activities. Two of these are milestones and one is a testing activity. All three activities are temporal parts of the X project and before after is used to indicate that milestone a is before milestone b. Note that there is no indication of the time interval in between.

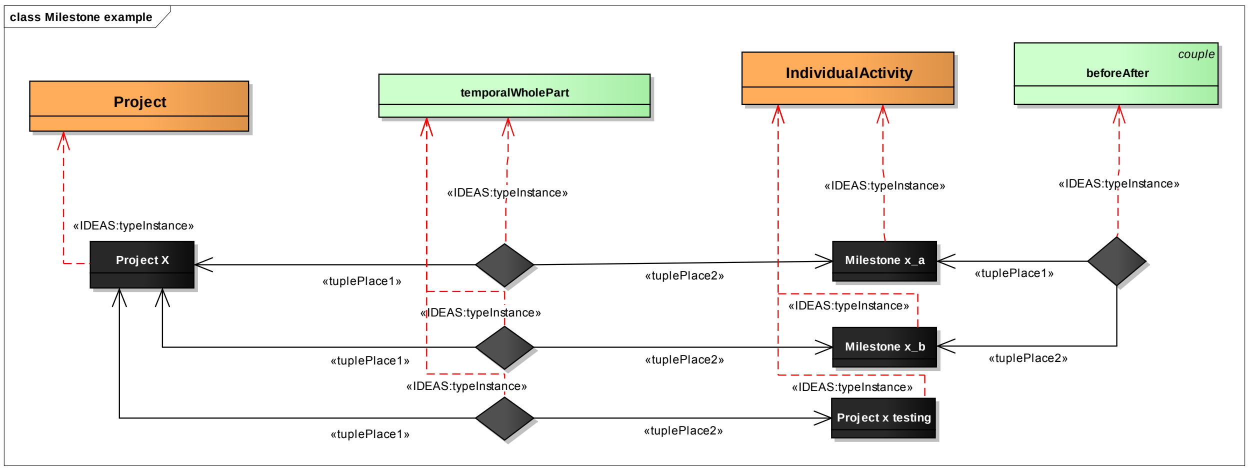


Figure 3. Temporal parts and before after

Activity in DoDAF 2 is the set of all subsets of the set of all individual activities and therefore the four sets defined here are instances of the Activity subset. Testing Kind A activities contain

Project x testing and Project y testing. Testing Kind B activities contain Project z testing. Milestone Kind A activities contain Milestone x_a, Milestone y_a and Milestone z_a. Milestone Kind B activities contain: Milestone x_b, Milestone y_b and Milestone z_b. Since all instances within Milestone Kind A activities occur (i.e. end) before all instances within Milestone Kind B activities an instance of BeforeAfterType can be created in the form of the element milestone Kind A before Milestone Kind B. This element contains all of the before after relationships defined in the example.

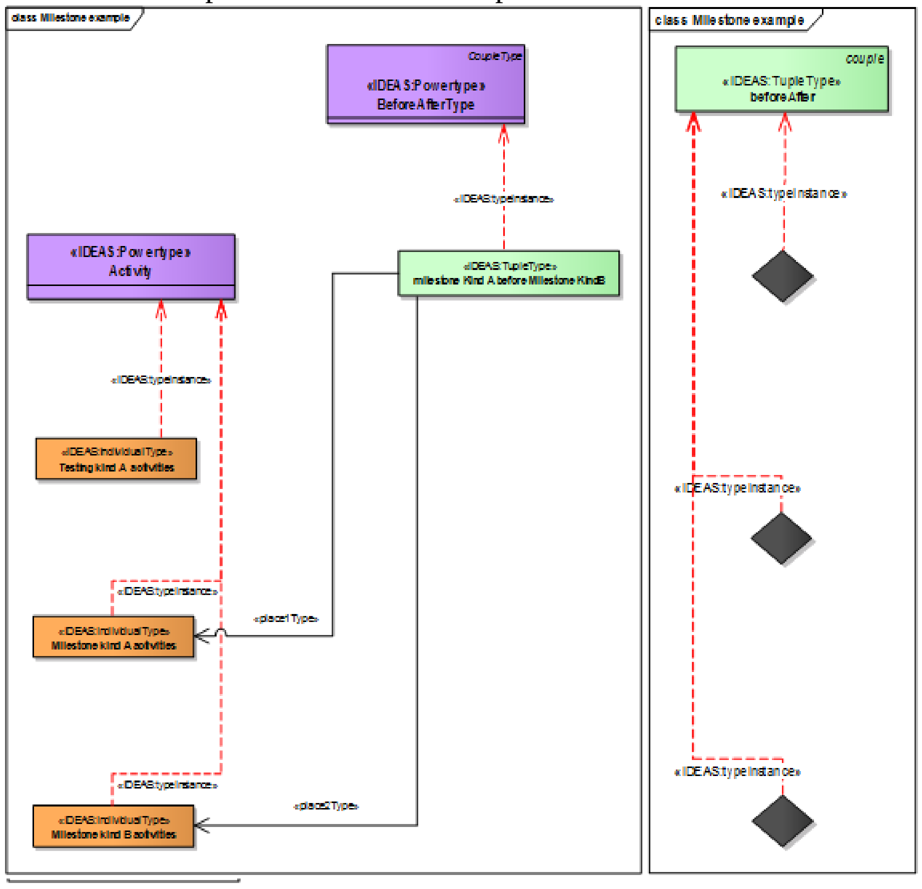


Figure 4. BeforeAfterType and Activities

Temporal Whole Part Type. As was shown previously, the testing activities can be combined into two distinct subsets that are instances of Activity (since it contains all possible subsets). This also means that instances of TemporalWholePartType can be created that contain the relationships that deal with temporal whole parts for testing Kind A and testing kind B. These in turn are instances of the DM2 element activityPartOfProjectType.

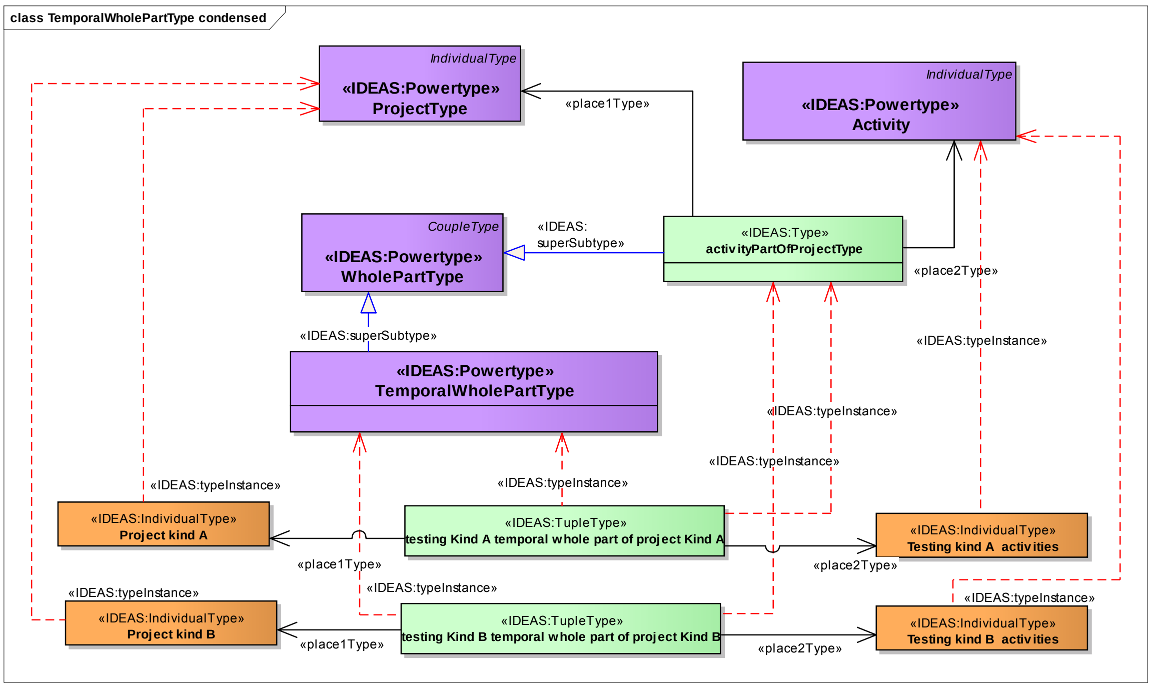


Figure 5. TemporalWholePartType

The Big Picture in Great Detail. Figure 2 contains the information required to express the concepts related to 3 simple projects each with some milestones and activities. Figures 3-5 then go on to describe the "internal wiring" that is necessary in order to properly express these concepts in an unambiguous and precise way such that assertions can be made about the model and demonstrated to be either true or false. The intention of these descriptions was not meant to criticize DoDAF 2.0 or the DM2. It is however apparent that getting architects to model in this fashion is not desirable. The details of an ontologically correct model need to be hidden from normal users by means of a tool that presents an easier way for architects to express what they want to express. This is the foundation that supports the DM2 and it is essential for proper modeling. However, in order for architects to build models in a more understandable fashion, DoDAF 2.0 needs to be implemented in an architecture tool. This is the purpose of UPDM.

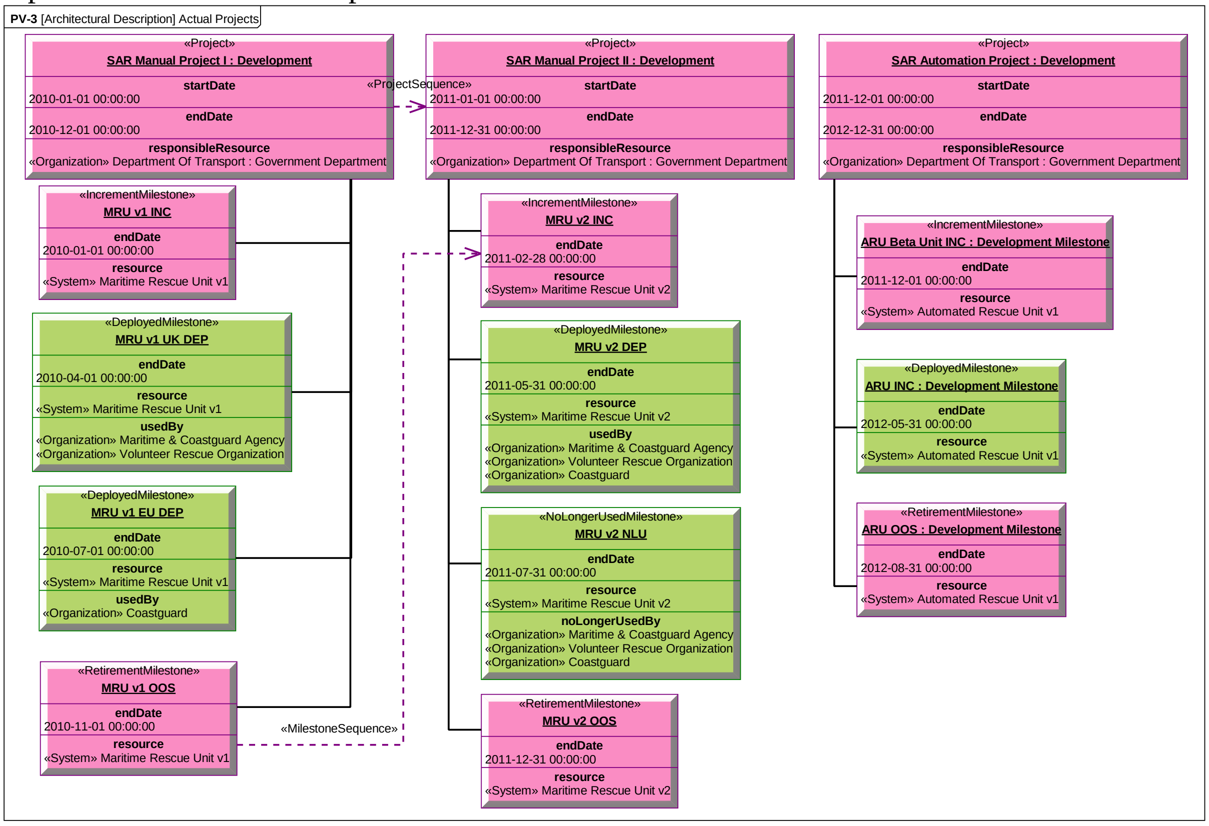
The Unified Profile for DoDAF and MODAF (UPDM)

The Unified Profile for DoDAF and MODAF, (UPDM) initiative was started by members of INCOSE, the OMG, the US Department of Defense, and the British Ministry of Defence. UPDM provides a consistent, standardized means to describe DoDAF and MODAF 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. It is important to stress that UPDM is not a new architecture framework. Instead, it provides a consistent, standardized means to describe DoDAF, MODAF and NAF architectures in UML-based tools as well as a standard for interchange (Hause, M.C., 2009), (OMG, 2005), (OMG, 2010), (OMG, 2009), (OMG, 2012), (DoDAF/DM2, 2010).

UPDM Examples

The following section contains several examples of the use of UPDM to express temporal aspects of architectures. The set of concepts listed in the introduction cannot all be described due to the limitations of space for this paper. It is also worth noting that that list is a short subset of all the concepts that are possible to express in UPDM. Consequently we will touch on a demonstrative subset.

Project Sequences. The PV-2 model shown in Figure 2 has been redrawn using UPDM in Figure 6. It still shows 3 projects each with a set of milestones as well as additional information regarding organization, fielded systems, milestone and project sequences. The same information is shown. However, the result is a far useable, clearer and comprehensible representation of the concepts.



PV-3 [Architectural Description] Actual Projects
«Project» SAR Manual Project I : Development
startDate 2010-01-01 00:00:00
endDate 2010-12-01 00:00:00
responsibleResource «Organization» Department Of Transport : Government Department
«ProjectSequence»
«Project» SAR Manual Project II : Development
startDate 2011-01-01 00:00:00
endDate 2011-12-31 00:00:00
responsibleResource «Organization» Department Of Transport : Government Department
«Project» SAR Automation Project : Development
startDate 2011-12-01 00:00:00
endDate 2012-12-31 00:00:00
responsibleResource «Organization» Department Of Transport : Government Department
«IncrementMilestone» MRU v1 INC
endDate 2010-01-01 00:00:00
resource «System» Maritime Rescue Unit v1
«DeployedMilestone» MRU v1 UK DEP
endDate 2010-04-01 00:00:00
resource «System» Maritime Rescue Unit v1
usedBy «Organization» Maritime & Coastguard Agency «Organization» Volunteer Rescue Organization
«DeployedMilestone» MRU v1 EU DEP
endDate 2010-07-01 00:00:00
resource «System» Maritime Rescue Unit v1
usedBy «Organization» Coastguard
«RetirementMilestone» MRU v1 OOS
endDate 2010-11-01 00:00:00
resource «System» Maritime Rescue Unit v1
«MilestoneSequence»
«IncrementMilestone» MRU v2 INC
endDate 2011-02-28 00:00:00
resource «System» Maritime Rescue Unit v2
«DeployedMilestone» MRU v2 DEP
endDate 2011-05-31 00:00:00
resource «System» Maritime Rescue Unit v2
usedBy «Organization» Maritime & Coastguard Agency «Organization» Volunteer Rescue Organization «Organization» Coastguard
«NoLongerUsedMilestone» MRU v2 NLU
endDate 2011-07-31 00:00:00
resource «System» Maritime Rescue Unit v2
noLongerUsedBy «Organization» Maritime & Coastguard Agency «Organization» Volunteer Rescue Organization «Organization» Coastguard
«RetirementMilestone» MRU v2 OOS
endDate 2011-12-31 00:00:00
resource «System» Maritime Rescue Unit v2
«IncrementMilestone» ARU Beta Unit INC : Development Milestone
endDate 2011-12-01 00:00:00
resource «System» Automated Rescue Unit v1
«DeployedMilestone» ARU INC : Development Milestone
endDate 2012-05-31 00:00:00
resource «System» Automated Rescue Unit v1
«RetirementMilestone» ARU OOS : Development Milestone
endDate 2012-08-31 00:00:00
resource «System» Automated Rescue Unit v1

Figure 6. Project View in UPDM

Systems Changing Over Time. The Greek philosopher Plutarch posed what is known as the Paradox of the Ship of Theseus. He posed the question to discuss the concept of whether a system was composed of its parts, or whether the configuration of the parts was what defined the system. In his story, all the parts of a system (Theseus' ship) were replaced with different parts. The question is then raised, is it the same ship or a different ship? Another example is Abe Lincoln's axe. Lincoln was well known for his ability with an axe, and axes associated with his life are held in various museums. Are they all "Abe Lincoln's Axe"? This may be an interesting intellectual exercise, but how does it apply to architecture? It is obvious that systems change over time for a variety of reasons. These include:

- System lifecycle of design, manufacture, deployment, maintenance, retirement
- Changes for mission-based configurations
- Changes due to maintenance

Taking the example of an aircraft, given all these changes is it the same aircraft because it has the same tail number, or is it a different aircraft and should we consider it so when creating an architecture? Figure 7 contains several SV-1 diagrams showing the evolution of an Intelligence Analysis (IA) system over time. The acronym IMP refers to information (in any medium or form), material, or persons that can be collected and analyzed to produce intelligence. Starting from left to right, the initial IA system contains the IMP with Data Cleansing as well as a human Intelligence Analyst and Internal Comms System as well as interfaces between them. The central system groups the additional Data Fusion into an Intelligence Management system along with Data Cleansing. The final version adds Real-time Threat Analysis and an Intelligence Coordinator.

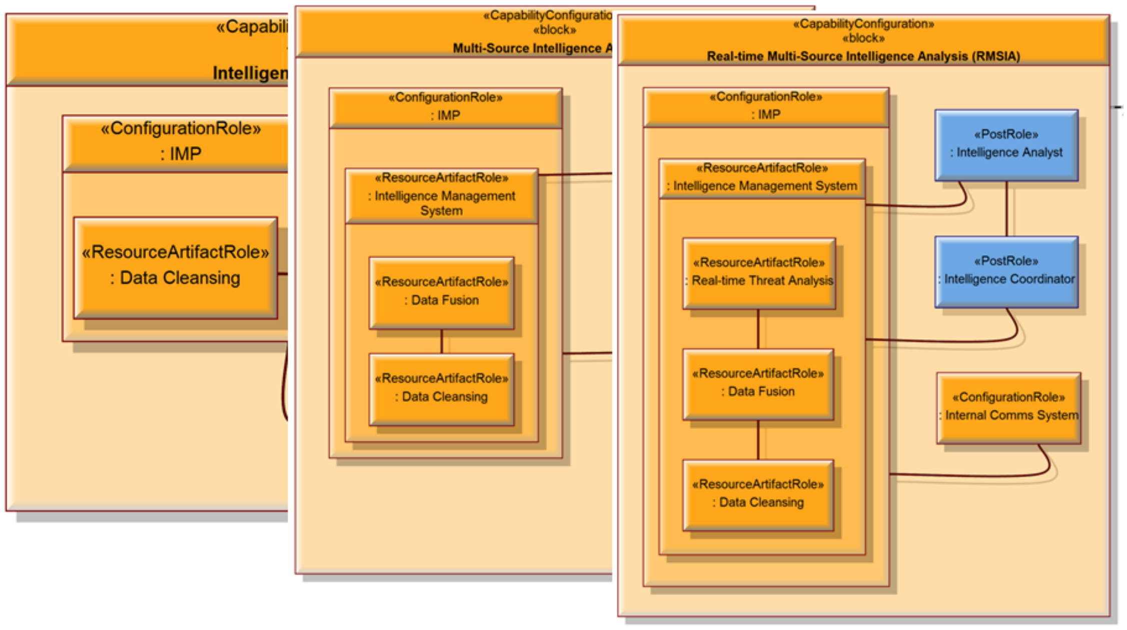
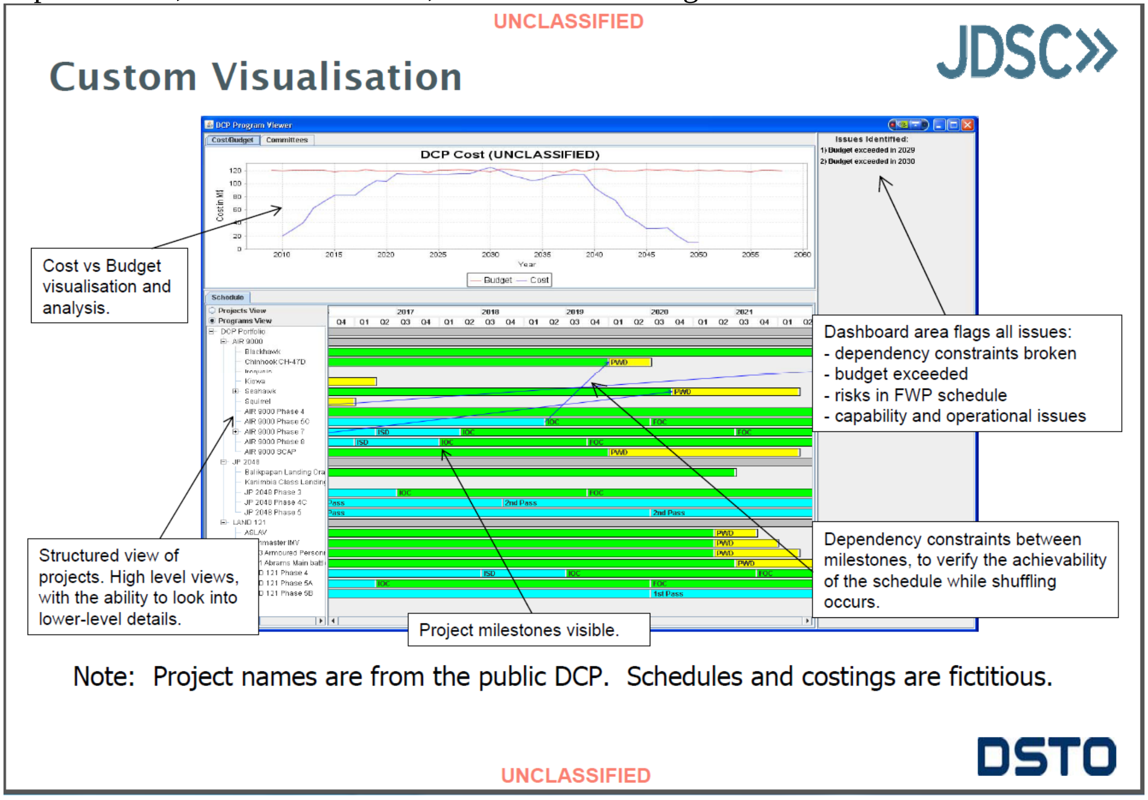

«CapabilityConfiguration»
«block»
Real-time Multi-Source Intelligence Analysis (RMSIA)
«ConfigurationRole» : IMP
«ResourceArtifactRole» : Intelligence Management System
«ResourceArtifactRole» : Real-time Threat Analysis
«ResourceArtifactRole» : Data Fusion
«ResourceArtifactRole» : Data Cleansing
«PostRole» : Intelligence Analyst
«PostRole» : Intelligence Coordinator
«ConfigurationRole» : Internal Comms System
Multi-Source Intelligence A
«Capabili
Intelligen

Figure 7. Intelligence Analysis System Configurations

Scheduling System Deployment. As mentioned earlier, the project view shown in Figure 6 also is used to show when systems are deployed over time. When linked to the capabilities, these can be used to show capability coverage as well as gaps using the CV-3 report. O'Shea et al (2012) expanded this view creating a Fit for Purpose view adding cost vs. budget, project dependencies, broken constraints, etc. as shown in Figure 8.


UNCLASSIFIED

Custom Visualisation

Cost vs Budget visualisation and analysis.

Dashboard area flags all issues:
- dependency constraints broken
- budget exceeded
- risks in FWP schedule
- capability and operational issues

Structured view of projects. High level views, with the ability to look into lower-level details.

Project milestones visible.

Dependency constraints between milestones, to verify the achievability of the schedule while shuffling occurs.

Note: Project names are from the public DCP. Schedules and costings are fictitious.

UNCLASSIFIED

Figure 8. Project Time vs. Cost vs. Capability

The work described in O'Shea (2012) applies a UPDM-based architecture development approach to capture capability development information with an emphasis on developing a fit-for-purpose visualization to support decision-making. This work includes the development of prototype visualization software to facilitate decision-support from DoDAF 2.0 architectural models. Users have the ability to shift projects on the timeline to determine the impact on overall budget as well as to ensure that costs for fiscal periods are not exceeded. Shifting the timeline will also affect equipment availability, thus impacting the ability to deliver essential capabilities. Finally, project approval cycles are built into the tool to ensure that sufficient project review time is available. Without the use of this tool, the work would largely be done by hand or by using multiple disconnected data sources. This saves project time, ensures capability, and saves ever decreasing tax dollars. Finally, the use of the UPDM repository will allow architects to further develop integrated architectures within the same repository rather than creating multiple disconnected models. Sections of the model can be extracted to form the basis of the development of architectures that support the required capabilities. This provides continuity throughout the development lifecycle.

Event and Interaction Sequences. The OV-6c can be used to describe operational activity sequence and timing that traces the actions in a scenario or critical sequence of events. The SV-10c provides a time-ordered examination of the system data elements exchanged between participating systems (external and internal), system functions, or human roles as a result of a particular scenario. Each event-trace diagram should have an accompanying description that defines the particular scenario or situation. Each SV-10c in the Systems and Services View may reflect system-specific aspects or refinements of critical sequences of events described in the Operational View. An example SV-10c is shown in Figure 9.

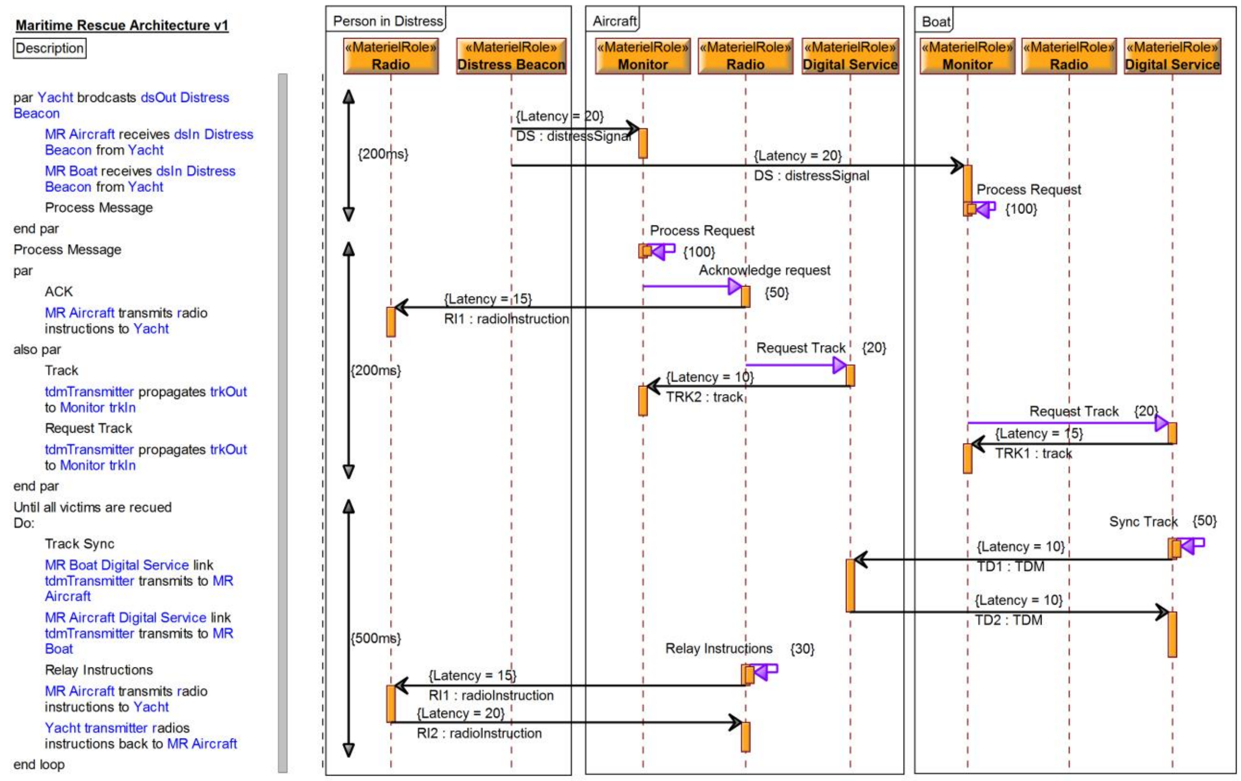


Figure 9. Event Sequence diagram

The diagram is owned by the system context. The elements shown are parts of this system. The interactions (horizontal flows) are those already defined in the in the SV-1 and SV-2. Time progresses from the top to the bottom of the diagram. Additional timing information has been added such as transmission latency and processing duration. Timing constraints are shown as vertical arrows on the left of the diagram. Static analysis can be done by collating the timing information on these diagrams into spreadsheets for numerical analysis. This provides architects with the ability to evaluate the performance of potential variant architectures. As the interactions are limited to those available in the configuration, consistency is built into the model. Simulation of behavioral portions of the model can be used to verify timing and behavior of the model for trade-off analysis and requirements specification. Having run the simulation, the timing information can be displayed on the sequence diagram, allowing the architect to evaluate alternate solutions.

State-Based Specification. The SV-10b state diagram is a graphical method of describing a system (or system function) response to various events by changing its state. The diagram basically represents the sets of events to which the systems in the architecture will respond (by taking an action to move to a new state) as a function of its current state. Each transition specifies an event and an action. Guard conditions including timing information can be added to these transitions to specify time-based transitions. The before-after concept described earlier is the underlying mechanism for the transitions. This is called a directed relationship in that it shows the transition from one state to another as shown in Figure 10.

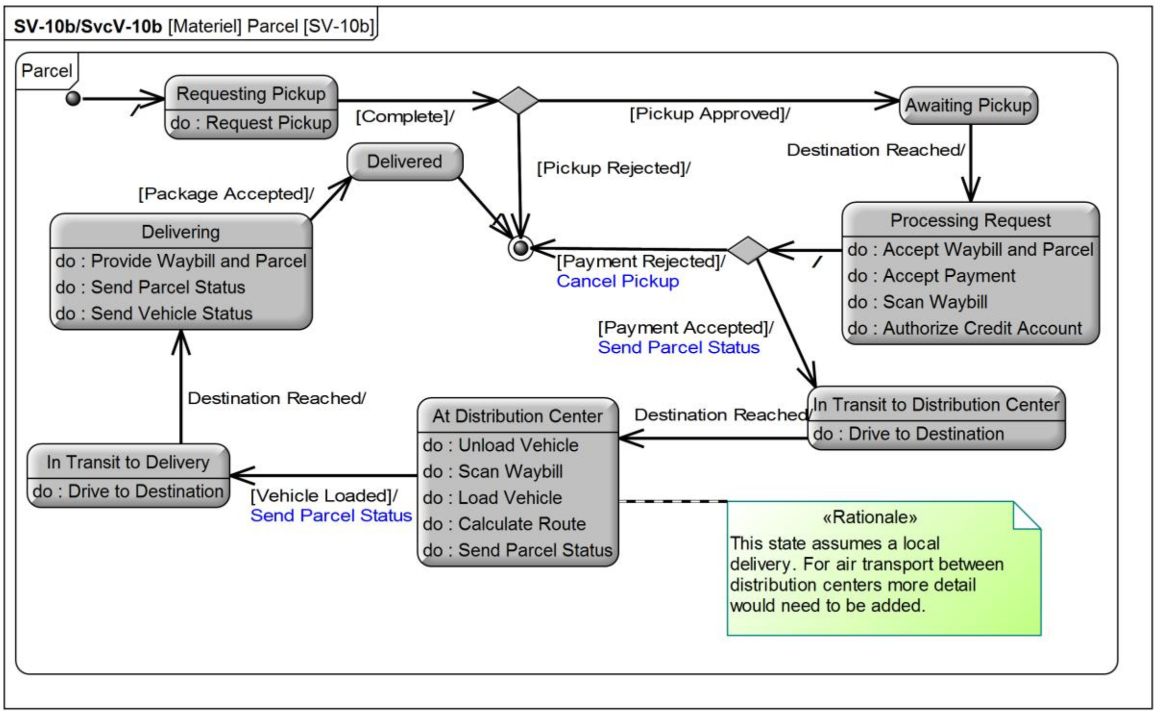


Figure 10. Case Management Lifecycle Using a State Transition Diagram

Another aspect of DoDAF 2.0 is the ability to show the interaction of more than just data. Energy, people, systems, organizations, etc. can be shown to be exchanged (or travel) between systems. This provides the ability to create architectures for logistics among other applications. The example shown in Figure 10 shows a simplified case management lifecycle for the delivery of a parcel. It shows the various states of the parcel, activities that can be performed on the parcel while in that state, valid transitions to other states, error conditions, and how the processing of the parcel changes in relation to its state. The SV-10b is normally used to show the state-based behavior of a system. This example was chosen as a means of demonstrating the additional capabilities now available in DoDAF 2.0 architectures.

The state machines shown above can be executed in UPDM tools. Timing constraints can be added to the transitions as well as embedded in the operations and activities of the state machine to demonstrate the behavior of the owning entity including time based behavior. By executing this state machine in conjunction with others in the model, the architect can perform behavioral and timing analysis of the architecture. This provides a means of performance based trade-off analysis.

Conclusion

Military architecture frameworks are powerful tools for enabling architects to define, design, plan, and implement enterprise architectures. The latest versions of frameworks such as DoDAF 2.0 are based on the ontological concepts in the IDEAS foundation objects. These concepts provide the detail necessary to express temporal concepts in precise and testable ways. Using tools implementing the UPDM standard, architects now have the tools to build the complex models needed to manage both government and industrial enterprises. UPDM tools provide the means to develop these architectures in a far more useable format. This paper has described examples of these concepts using UPDM. There are numerous other examples and it is hoped that these can be explored in future papers.

References

Cummings, Raymond King (1922). The Girl in the Golden Atom. U of Nebraska Press. p. 46. ISBN 978-0-8032-6457-1. Retrieved 9 April 2011. Chapter 5.

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

DoDAF/DM2 2.01.Version 2.01 as of 1 April 2010. Version Description Document for the DoD Architecture Framework (DoDAF) and DoDAF Meta Model (DM2), Version 2.01 can be downloaded from http://cio-nii.defense.gov/sites/dodaf20/products/DoDAF_DM2_VDD_v2-01.doc

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

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

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

IDEAS, 2012, Online http://www.ideasgroup.org

IEEE Std 610.12–1990, IEEE Standard Glossary of Software Engineering Terminology

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), 2010. Unified Modeling Language: Superstructure version 2.3 /2010-05-05. [online] Available from: http://www.omg.org/spec/UML/2.3/Superstructure/PDF/ [Accessed November 2012].

Object Management Group (OMG), 2012, OMG Systems Modeling Language (OMG SysML™), V1.3, OMG Document Number: formal/2012-06-01, URL: http://www.omg.org/spec/SysML/1.3/PDF, Accessed November, 2012

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

Object Management Group (OMG), 2012, Unified Profile for DoDAF/MODAF (UPDM) 2.0, available at http://www.omg.org/spec/UPDM/2.0/PDF, Accessed Nov 2012.

O'Shea, Kevin, Peter Pong and Gary Bulluss, 2012, Fit-for-Purpose Visualisation of Architecture to support Defence Capability Decision-Making Joint Operations Division, Defence Science and Technology Organisation, DSTO-TN-1098

Shakespeare, William, 1613, The Complete Works of William Shakespeare, Published by H. Pordes, 529 Finchley Road, London, NW3 7BH, England

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 Atego's Chief Consulting Engineer, the co-chair of the UPDM group and a member of the OMG SysML specification team. He has been developing multi-national complex systems for almost 35 years. He started out working in the power systems industry and has been involved in military command and control systems, process control, communications, SCADA, distributed control, and many other areas of technical and real-time systems. His roles have varied from project manager to developer. His role at Atego includes mentoring, sales presentations, standards development and training courses. He has written a series of white papers on architectural modeling, project management, systems engineering, model-based engineering, human factors, safety critical systems development, virtual team management, systems development, and software development with UML, SysML and Architectural Frameworks such as DoDAF and MODAF. He has been a regular presenter at INCOSE, the IEEE, BCS, the IET, the OMG, DoD Enterprise Architecture and many other conferences.

Lars-Olof Kihlstrom was tasked to aid the UK during the finalization work of the NATO architecture framework version 3 based on MODAF. Lars-Olof was asked to act as a modeling expert by the Swedish armed forces in the IDEAS group where the UK, USA, Canada and Australia is creating an ontology for architecture information that will enable the different architecture frameworks to exchange architecture information. Responsible for MODAF framework usage guidance description creation. Lars-Olof is a member of the architectural core of the UPDM 2.0 submission team responsible for development of the UPDM 2.0 submission. Presentations at the Enterprise Architecture conference in London concerning the use of SOA in NAF and the handling of the MODAF Learning Portal has also been held together with the Swedish Armed forces in 2008 and 2009. The Presentation concerning SOA has also been held at the DoD architecture conference in Orlando Florida. He has been project manager for the creation of MODEM where MODAF was re-engineered in order to base it completely on the IDEAS Foundation and remove its reliance on a UML profile. This work was conducted in a small expert team composed of Lars-Olof, Ian Bailey and Chris Partridge.