The UPDM RFC Development Project
An Exercise in Model-Based Virtual Team Development
or "Practicing What We Preach"

Matthew Hause
Artisan Software Tools
Chief Consulting Engineer, Co-Chair UPDM Group
Suite 701 Tower Eagle
Cheltenham, GL50 1TN, England
MatthewH@Artisansw.com

Abstract. Model Based Systems Engineering (MBSE) in a distributed environment using virtual teams presents many challenges. This paper will discuss the development projects for creating specifications for a Unified Profile for DoDAF and MODAF (UPDM) versions 1.0 and 2.0, which were created under the Object Management Group (OMG). The UPDM 1.0 project was completed in 5 months and involved people from 21 organizations in 11 different time zones. For UPDM 1.0, three face-to-face meetings were held with sub-groups, but the majority of the work was performed virtually using MBSE techniques. In this paper we will discuss the challenges we encountered, the lessons learned, and how we have changed our working practices for UPDM 2.0.

INTRODUCTION

Background. In the systems development industry, tool vendors, academia, and industry have been advocating a model-based approach to development for many years. We have also preached the benefits of virtual teams and claimed that it was no longer necessary for project teams to be co-located (physically present in the same room or building). In March 2008, a team was formed to define a specification for a Unified Profile for DoDAF and MODAF (UPDM). From the beginning we decided that we were going to live up to our principles by creating a model of our requirements, and generate the UPDM profile, specification and the XMI from the model. Discussions and interchange of information were to be based on the model information and all information was to reside in the model. The military organizations involved were the US DoD, DISA, Mitre, the UK MOD, the Canadian DND, NATO, and Generic AB for the Swedish Military. The support of the military organizations was key to the success of the project and ongoing support for the specification and tools implementing the specification. UML Tool Vendors were Adaptive, Artisan Software (Co-Chair), EmbeddedPlus, NoMagic (Co-Chair), Sparx Systems, and Visumpoint. IBM and Mega officially joined the team after the specification was complete and are participating in UPDM 2.0. Industry partners providing domain expertise were BAE Systems, Lockheed Martin, Selex SI, Raytheon, and Rolls Royce. Thales, Northrop Grumman, and General Dynamics joined the team for UPDM 2.0. Advisors were Decisive Analytics, Silver Bullet, Model Futures, and ASMG Ltd. The team was a distributed multinational team with representatives from the US, UK, France, Sweden, Lithuania, Australia, Canada, and Thailand in 11 different time zones. Before speaking about how the project was conducted, let's briefly look at the rationale for the project. For more information on the UPDM specification itself see Hause, (2009), and OMG, (2009a).

The Need for UPDM. So what is UPDM and why is it necessary? A plethora of military architectural frameworks, such as DoDAF (USA), MODAF (UK), NAF (NATO), DNDAF (Canada), MDAF (Italy), AGATE (France), and ADOAF (Australia) is emerging. 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. Figure 1 shows the development and relationships between a subset of these, namely, DoDAF, MODAF, and NAF.

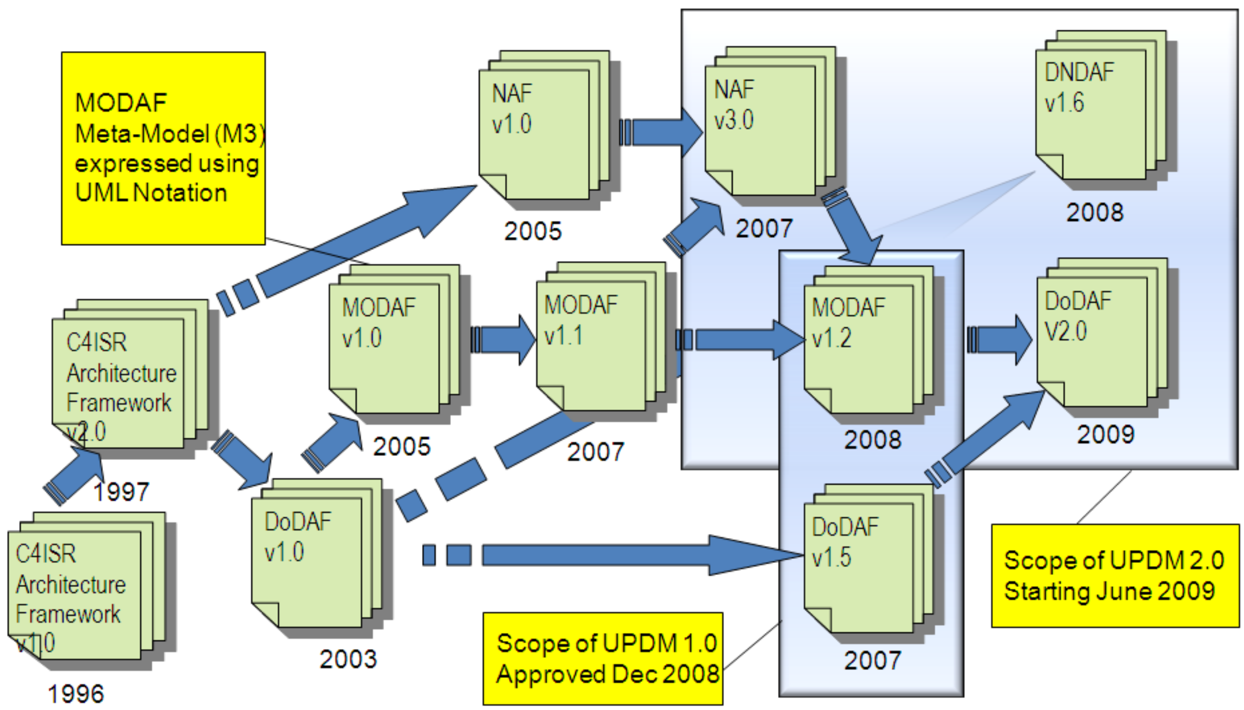

Figure 1. DoDAF, MODAF, and NAF Timeline and Derivation

These frameworks have been developing and growing in complexity over the last several years. Consequently, supporting multiple and sometimes divergent frameworks leads to problems for industry, military organizations and tools 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 architecture 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.

Arguably, the two most widely used 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 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 now required to document their enterprise architectures using DoDAF.

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 called the Capability and Project Views in DoDAF 2.0. 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 and have been included in DoDAF 2.0. 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.

SysML and UML. The Unified Modeling Language (UML) and Systems Modeling Language (SysML) are used as an underlying mechanism for the 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. For further information on SysML see Hause (2006), OMG (2007b), Friedenthal (2008), Holt, (2008), and Korff (2008). 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). It is important to stress that UPDM is not a new architecture framework. Instead, it provides a consistent, standardized means to describe DoDAF and MODAF architectures in UML-based tools as well as a standard for interchange.

Reuse of existing specifications. UPDM reuses UML/SysML wherever practical to satisfy the requirements of the RFP (Request For Proposal) 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 intended to reuse the UML Profile and Metamodel for Services (UPMS). However, since UPMS had not been formally adopted at the time of this specification, a separate service profile in UPDM was developed that used similar concepts. UPMS was recently renamed to SoaML, (Service Oriented Architectures Modeling Language.) As part of the finalization effort of UPDM 1.0, SoaML was officially integrated into

UPDM and UPDM imports common concepts and strategies. This was particularly difficult because the specific, official mechanism for a profile to reuse, import, or merge another profile has not fully been documented or agreed upon by the OMG. A new group has been formed by the OMG to investigate this particular problem as the different standards teams are starting to run into this more often. To find out more about UPDM go to www.UPDM.com. To find out more about the OMG, go to www.omg.org.

The rest of the paper will not look at UPDM itself, but will deal with the organization of the teams and sub-teams, how we worked and didn't work together, lessons learned and what we will change in the future.

PROJECT KICK-OFF

The UPDM RFC (Request for Comment) project was initiated in March 2008 carrying on from the previous UPDM effort. The previous UPDM effort was initiated in response to the OMG Request for Proposal (RFP) for a UML Profile for DoDAF and MODAF, itself, a response to the UPDM Request For Information (OMG, 2005). Figure 2 shows a timeline of the project including the various teams involved prior to the submission in March 2007.

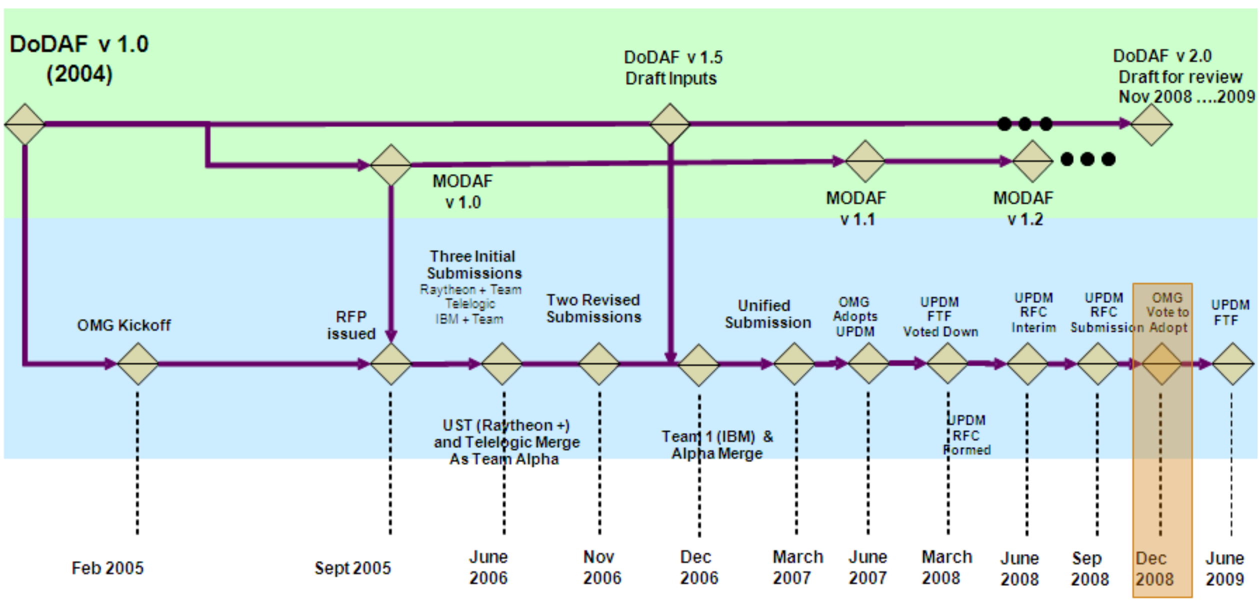

Figure 2. UPDM Timeline

It was rejected by the OMG, DoD, and MOD as it failed to meet the needs of the defence community. There were other reasons as well, but it would not be helpful or useful to go into them at this stage. Instead, it would be more useful to state the guiding principles of the 2008 effort. These are detailed in the following sections.

Model-Based Development of the Specification. OMG processes are still largely document driven. To some extent, this is understandable as it is a standards organization. Their deliverables are not systems, but instead are specifications for standards. The standards are normally implemented by tool vendors who build tools that implement the standard such as UML, SysML, CORBA, etc. However, like industry specifications, OMG documents are full of embedded diagrams, explanatory text referencing the diagrams, and appendices, examples, etc, summarizing and elaborating on information contained within the diagrams. Creation and modification of the specification involves many people. Consequently, it is extremely challenging to produce specifications like this using a document centric approach. For these reasons, the group decided on a model centric approach where the specification and XMI files would be generated from the model. For more information on the OMG standards processes, see OMG (2008b) and the OMG website.

Open, Collaborative Process. Projects can have problems sharing information for many reasons. These can involve the technology or tools used, security concerns, political reasons, and all the other well known problems that can occur collaborating with a distributed team. We had all of these problems, as well as the inter-organizational communications aspects involving sharing documents with people in organizations that have high security requirements. Issues arose during the previous UPDM effort as not all parties had access to the latest specification and insufficient time was provided for reviews. Consequently, we all worked hard to ensure that the process was open and collaborative. In addition, in the previous effort there was a lack of stakeholder involvement until much later in the development process, leading to conflicts, misunderstandings, and a specification that satisfied old requirements. The group therefore sought to include all stakeholders in decision making, and maintained an open membership, even to those who were not OMG members. This dispensation was granted to us by the OMG as the application domain had unique requirements. It is often said that success motivates a team onto other successes. Failure can be an even greater motivator in that people are determined not to repeat the same mistakes. Of course, one always has to avoid what Brooks called the "second system syndrome". In computing, the second-system effect or sometimes the second-system syndrome refers to the tendency, when following on from a relatively small, elegant, and successful system, to design the successor as an elephantine, feature-laden monstrosity (Brooks, 1995). Given the tight timescales and strict set of requirements, this was unlikely to be the case.

All Member Inputs Considered. The keywords to this principle were Discuss, Debate, Decide, Prioritize, and Defer. Discussion took place via email and during web meetings, after which a decision would be made and presented to the group. A prioritized issues list was maintained which reflected the history of each issue as well as any resolution. Old issues could be re-examined in light of new evidence. Finally, issues that could not immediately be resolved were deferred until more information was gathered. These were revisited on a regular basis to ensure that a large list of problems were not being saved up until the end..

80-20 Rule. This is a variation on the Pareto Principle that states that for many events; roughly 80% of the effects come from 20% of the causes (Pareto, 1971). This manifested itself in two areas: DoDAF/MODAF mapping and issue resolution. A perfect mapping between DoDAF and MODAF could never be achieved, especially given our challenging timescales. As a start, we created a mapping between the two to get a first cut, followed by the definition of a Domain Meta-Model (DMM) to capture the key concepts. In other words, we created a model of what we were going to model. This allowed us to agree on the key concepts and formed a requirements specification for the UPDM profile itself. Mappings were then created using a best-fit approach to avoid getting hung up on the minutiae. This allowed us to create a profile that would satisfy the vast majority of the UPDM modelers. Given the extensible nature of UML based models, those requiring additional work could be done by tools vendors as and when needed. This also provided a means for fulfilling another principle, which was to "Keep it Simple".

Re-Use Rather than Re-Define. Given the complexity of the architectural frameworks involved, (MODAF 1.2 itself has over 40 views), we needed to be certain not to add additional complexity by inventing new concepts, or even worse by re-inventing the wheel. The specification itself contained concepts from MODAF 1.2/M3, DoDAF 1.5/2.0, NAF 3.0, UML 2, SysML 1, BMM, UPMS (now SOAML), BPMN, etc. Explicit integrations were not made with all of these however. Also, definitions and descriptions of the concepts always referred to the original definitions and terms to avoid inventing new terms, buzzwords, or slang which would be unfamiliar to domain experts. The purpose of UPDM is to provide a standardized way to express DoDAF and MODAF concepts using UML and SysML, and not to create a new framework.

PROJECT EXECUTION

Project Timescales and Goals. Specifications of this size and complexity normally will take 1-2 years to complete. For example, the SysML specification took 3 years to complete starting from the release of the initial RFP. We set ourselves the goal of completing this in 5 months for the initial release. This was for several reasons. The previous UPDM effort was to define a profile for DoDAF 1.0 and MODAF 1.0. Since the start of that project, MODAF had progressed to version 1.2 and DoDAF to version 1.5. As it was unlikely that anyone would start a new project using the outdated versions, the specification was largely irrelevant. Also, the release of DoDAF 2.0 was projected for the end of 2008 (it was eventually released in May 2009), so the new UPDM specification needed to be completed before it also became irrelevant. As there was no standard way of representing DoDAF/MODAF concepts using UML/SysML, each tool vendor had created a unique implementation, thus rendering interchange impossible. Finally, once DoDAF 2.0 was released, tool vendors would be eager to support it. In order to avoid the previously mentioned interchange problems, we needed to complete the definition of UPDM 1.0 before starting on UPDM 2.0 and support for DoDAF 2.0.

The RFC Process. One technical aspect of OMG procedures and rules was that requirements for a specification cannot be changed once the specification is in its finalization phase. (This differs from other types of projects where it almost seems that requirements changes are mandatory, and the closer to project end date the better!) Using normal procedures, we would issue a new Request for Proposal (RFP), which under OMG rules would take at least two meeting cycles (or 6 months) followed by the development of the specification. As we had such tight timescales, the RFP process was not going to work. Instead, we opted for the RFC or Request for Comment process. This is normally used by external organizations or companies that have already created a standard that is in use and would like to see it adopted by a standards body. Under this process, the specification is submitted, after which it is released for a 60 day review period. The OMG members can then vote to adopt. If they do vote to adopt, the specification is finalized by a Finalization Task Force (FTF). For various reasons we were unable to make use of OMG facilities to generate the specification and to manage the project. Consequently, we had to use our own initiative to use freeware tools or to make tools used by the team members available to the whole team. As mentioned earlier, this is atypical of OMG standards projects. Having access to the proper tools makes all the difference, and the OMG has developed these over many years. Wikis are now widely used, as they provide a means of sharing documents, information and procedures. However, integrated CM is not provided.

MBSE. Even though MBSE is a well known concept it is worth repeating its definition, as defined by INCOSE. “Model-based Systems Engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing through-out development and later lifecycle phases." (INCOSE, 2007). Put simply, modeling is at the heart of all aspects of the development effort, covering the complete lifecycle, and has a direct effect on any generated artifacts. For the UPDM project, this was achieved by creating models of the requirements (the Domain Meta-Model), the design (the profile itself), the implementation (to be implemented by the tool vendors) and the proof of concept (the example model.) Links between the DMM and the UPDM profile were maintained in the model and traceability tables were generated to ensure compliance. The final specification that was submitted to the OMG was also generated from the model. This allowed us to continue working until two weeks before the due date of the specification thus giving us more time to do the real engineering work. More on this later.

Virtual Teams and their issues. Technology has developed such that it allows for effective remote collaboration on system development in a team-based environment. Two avionics examples of systems developed in such a way are the Airbus 380, and the Eurofighter, with development split between several European countries. Virtual Teams are groups that are formed for executing a specific, normally long-term project. This requires groups to share information and development artifacts, and to communicate both synchronously and asynchronously on a variety of subjects, as well as the developing social relationships normally found in teams. By necessity, standards bodies operate using virtual teams. To achieve industry consensus, it is necessary for a variety of companies and organizations to work together to produce the standard. Studies have found that virtual teams have the same dynamics, issues, interactions, and social lifecycles as co-located teams. Results applicable to industry include team management style, group development, the decision making process, the use of communication media available, the amount of communication per team, decision-making patterns throughout the system development process, and most importantly effective use of systems engineering in a development project (Hause & Hause, 2008). Awareness of these issues meant that many of the problems found in virtual teams were avoided. For further information see Hause & Hause (2008), the studies referenced there, and other recent INCOSE publications on the subject. Research in academic collaboration using Wikis is being done at the most notably at the Open University of Great Britain, the Open University of Israel and others. See Tal & Tal (2006a, 2006b) and Bruns and Humphreys (2005). The research on Wikis is cited as they are widely used in the development of standards, particularly at the OMG.

Project Organization. The project team was subdivided into several groups. The architecture group consisted of members from four of the modeling tool companies. They took the lead in creating the DMM, translating it into the UPDM profile, maintaining the issues list, and ensuring consistency. During development it was felt by some of the group that this information was not shared widely enough with the rest of the team and that decisions made and their rationale were not sufficiently transparent. Use of the OMG Wiki to share information during the development of UPDM 1.1 and UPDM 2.0 has allowed us to resolve that issue. Project management, process definition, and compliance were done by one of the chairs, with the other taking responsibility for generating the specification. The example model was created by another group with one of the defense industry members acting as domain expert. Traceability between MODAF, DoDAF, NAF and the DMM was done on an ongoing basis by another group. MOD and DoD staff acted as domain experts for the group and ensured that UPDM was fit for purpose. The rest of the team raised and helped resolve issues, and documented the DMM and UPDM profile elements.

Tools Used. Microsoft Word was used as the format for generated documents. Although a proprietary Microsoft standard, it is nonetheless a de facto industry standard. This caused us some problems later on as the OMG requires documents to be delivered in Framemaker or Open Office format. Conversion from Word is not as seamless as is advertised and errors were introduced into the final document by the conversion process. The main tool used was a UML modeling tool. This was used to define the DMM, the UPDM profile, identify issues, document the main concepts, produce the specification, and generate the XMI. Issues were also identified and flagged in the model using the UML/SysML comment element. These were then tracked and resolved as required. There were some problems with sharing the model. UML tools are normally either used via a local network, or more recently via terminal servers. Local networks were not an option due to the distributed nature of the group. Due to security restrictions and resource constraints, no member of the group was able to provide terminal server access. As the UML modeling tool did not have merge capabilities, this meant that we had to operate as if the UML modeling tool were a single user modeling tool. This caused several problems. Edits to the model had to be done on a transaction or token passing basis, which slowed things down. The latest model had to be passed from person to person to implement any changes. There was also the inevitable confusion over which was the latest version. For simplicity, the decision was made to limit the changes to a single person, or gatekeeper to the model. This caused demand on the modeler, and meant that he had the potential to become a bottleneck for the changes, especially when his employer put other project pressures on him. However, his heroic (long hours) working practices meant this did not take place. Regardless, it is not practical to rely on this for time critical projects.

Configuration Management Repository. To resolve the management problems with the model updates, an attempt was made to use an open source configuration management tool, hosted by one of the team members. The user interface for the tool made it somewhat difficult to learn, but the product was useable. Unfortunately, because some of the team members worked for large defense contractors, there were unable to access the artifacts in the database. These involved security restrictions, restrictions on using open source software, and others. Consequently, use of the Configuration Management tool had to be abandoned. Eventually, the latest versions of all the project artifacts were stored on a Java site to avoid continuously emailing large files. This still had the disadvantage that it was difficult to determine if the latest file on the website was in fact the latest file and to avoid simultaneous edits.

Generation of the final document. For a standards project, the standard specification document is the end goal of the project. Templates were developed to generate the specification from the model. Unfortunately, this took far longer than expected leading to several marathon evenings prior to the due date. The final specification was delivered on time, but at the cost of several sleepless nights by team members. The lesson was learned that we needed to start the process earlier to work out any issues. Additionally, generation of the specification should be prototyped to ensure that it will conform to the correct format.

Generation of the OCL.. The Object Constraint Language (OCL) is a formal language used to describe expressions on UML models. These expressions typically specify invariant conditions that must hold for the system being modeled or queries over objects described in a model. Note that when the OCL expressions are evaluated, they do not have side effects (i.e., their evaluation cannot alter the state of the corresponding executing system). OCL expressions can be used to specify operations / actions that, when executed, do alter the state of the system. UML modelers can use OCL to specify application-specific constraints in their models (OMG, 2009b). OCL is a text language as opposed to a graphical language. Thus all the problems associated with keeping text consistent occur when using OCL regarding consistent naming, consistent relationships, and general maintenance. Additionally, OCL is difficult to understand and visualize what exactly is meant by the constraint. To solve this problem, the team created a graphical notation using UML dependencies and relationships. The OCL text was then generated from the graphical elements and inserted in the appropriate location in the document and in the correct syntax.

Project Meetings. Virtual team meetings were held weekly at a set day and time to allow people to schedule these ahead of time. Those at the far ends of the time zone spectrum were the most inconvenienced with people attending at 06:00 and 18:00 respectively. A variety of online meeting tools were used such as WebEx, Net meeting, and others. Audio communication was done via conference call as all team members did not have access to free tools such as Skype. Agendas were posted ahead of time, minutes taken, and published as soon as possible after the meeting. Action items were documented and tracked for completion. Three face-to-face meetings were held: one after the project had progressed for a month, one at the mid-point of the project during the normal OMG meeting cycle, and another two weeks prior to the due date of the specification. The main thrust of the meetings was to sort out technical issues with the DMM, UPDM profile, and compatibility with DoDAF and MODAF. In addition, the DoDAF 2.0 experts were able to attend to discuss convergence. As DoDAF 2.0 was still under development, it was deemed useful to work towards a common goal rather than to work independently and try to sort out issues when it was too late. This meant that we were spending time addressing issues that would not be relevant until UPDM 2.0, but we decided that it was worth the investment to reduce future problems. MOD members of the MODAF team also attended to ensure MODAF compliance, and provided us with pre-release versions of MODAF 1.2. Side meetings were held with the MOD personnel to gain their approval.

List of issues and its maintenance. The issues list was a multi-column table containing a unique ID, description of the problem, suggested resolution, current status, person who raised the issue, person responsible for fixing it, etc. We initially used Excel, but found that Microsoft Word tables worked better due to problems with maximum allowable sizes in Excel cells, and lack of a spell checker in earlier versions. The issues list was maintained by a member of the architecture group, who took responsibility for ensuring it was up to date and that issues identified as fixed had been implemented correctly. This was essential to make sure that issues were corrected as requested by the person who found the problem. Again, this was exacerbated by the virtual nature of the team as it was not always possible to communicate synchronously.

Running a Project with Volunteers. From a project management point of view, running any project with volunteers is a nightmare. Chiefly among these is lack of management control over deliverables and work being done. As all members still had their day jobs to do, work was often done out of hours during the person's free time. Obviously, this cannot be relied upon as goodwill will only go so far. However, as the group was highly motivated, people put in the necessary hours, performed tasks as requested, and most deliverables were delivered on time.

LESSONS LEARNED

The project continues to be a challenge and is certainly one of the most rewarding that I have ever worked on. The following were some of the lessons that were learned during its execution. This represents the ideal as opposed to what may be immediately possible.

- MBSE works!

Because of the use of MBSE working practices, we were able to concentrate on completing the project rather than fighting the documents. It also allowed us to continue developing right up to the end of the project with a large degree of confidence that the specification would be complete, consistent, and coherent. Had we chosen a document-centric approach, we would have written the specification by hand, drawn the diagrams in Visio, PowerPoint or other graphic/non data oriented media, inserted the diagrams into the specification and then written descriptions of the elements. If the diagrams were to have changed, updating the diagrams and relationships would all have had to be done by hand and relied more on people’s memories than good engineering. In addition, generating the specification automatically meant that we were able to ensure a consistent and correct style and that the model matched the specification which matched the XMI, which matched the profile specification. Finally, MBSE techniques ensured that traceability from the DMM to MODAF, DoDAF and NAF were maintained and that reports were generated automatically.

- Virtual communication requires more time.

Some of the problems we encountered were due to lack of time. Often this is because of the increased time required for virtual communications and that much of the communication was asynchronous. It is not possible to walk down the hall and talk to the person or people directly. Additionally, as previously stated, even if they are contacted, their priorities may not be your priorities. This needs to be factored into any project timescales.

- Ensure that project information is accessible.

Access to current and correct information is the lifeblood to any project. Much time can be wasted by editing the wrong document, or making project decisions based on faulty information. For virtual teams across many organizations, this is not only essential, but infinitely more complex.

- Ensure that the model is centralized and distributed.

This does sound like a contradiction, but it is possible. Mostly it involves making MBSE a reality. A centralized repository for the information as opposed to a file-based system ensures that people are able to access the model as a whole rather than snippets. In addition, because of the pervasive nature of the model elements and the need for cross-references, much of the model is required for most operations. Keeping as much information in the model simplifies traceability considerably and helps ensure completeness, correctness, and consistency. It also provides a means for impact analysis.

- Provide Versioning, Variants, and Backups.

If possible this should be done on a whole model basis. Again, this ensures that impact and traceability can be assessed against the whole model. If done on a section by section basis, it becomes more likely that version skew will take place.

- If possible, use dynamic references.

Models are made up of both diagrams and descriptive text. The description of one element will often contain references to others. When the names of elements change, as they always do, this becomes a maintenance nightmare. The references in the descriptive fields then have to be searched and changed individually. Using dynamic or rich text references means that a reference to the element becomes embedded in the description field. These are automatically updated when the names are changed. It also provides a means of tracking these cross-references.

- Maintain the project schedule and ensure it is “trackable”.

A project schedule that is trackable is one where the project schedule tasks and deliverables correspond to what people are actually doing on the project. This may sound obvious, but I have been unpleasantly surprised by too many project schedules to assume that this is always the case. Regular and short-term deliverables are essential to letting you know when you are falling behind. Finally, contingency planning needs to be done to investigate what to do when things go wrong. (Note: things always go wrong.)

- Keep communications open and regular.

Regular meetings and email are essential to ensure that everyone is kept up to date. Team building and socializing are just as important as technical discussions. It helps to build a sense of trust between the members of the groups and minimizes unnecessary conflicts. More concretely, it is unlikely you will work until 02:00 to ensure a deliverable is complete on time, if you don’t like the people in your group. As stated earlier, virtual teams have the same need for socialization as collocated teams (Hause and hause, 2008).

- Be familiar with the project and process standards.

Prior to starting the project, we reviewed the OMG and ISO standards to ensure that we would be in compliance. This was especially important pertaining to timescales. One example was the total time required for the FTF phase. The final specification needs to be delivered four weeks prior to the OMG submission meeting, issues need to be resolved 4 weeks prior to that and the process must take a minimum of 10 weeks. Added up, this means that the FTF process needs to run for at least 2 meeting cycles. Missing these finer points can be quite disastrous.

- Prototype the deliverables throughout the development lifecycle.

For full lifecycle projects this includes document generation templates, code generation templates, test rigs, system integration, project management tracking, disaster recovery, and so on. You do not want to find problems with your deliverables when you are mid-project.

- Ensure Ongoing Stakeholder Involvement

One of the main reasons for the success of the UPDM Group has been the ongoing support and participation of the US and UK defense organizations. They have provided manpower for review, discussion, explanations and general support for the work of the group. This ongoing review process meant that the specification was deemed fit for purpose by the defense organizations when it was released, and industry acceptance was ensured. As such, UPDM is the only implementation of DoDAF and MODAF that have been formally reviewed by the DoD and MOD to ensure that it implements DoDAF and MODAF correctly. This gives added confidence to those adopting UPDM that it will meet with the approval of their end users.

- If possible, start small.

Tom Gilb has a saying, “If you don’t know what you are doing, don’t do it in a big way.” Learn on a small project and make mistakes when they are not too costly (Gilb, 2010).

As with all projects and processes, the ongoing UPDM project is still far from perfect. However, striving to improve is what makes us all good engineers, and makes the job worthwhile.

Postscript and Further Work

It is worth noting that the UPDM specification passed through all the votes during the September and December 2008 OMG meetings. It has since undergone its finalization phase and is now an approved specification. Consequently, the project has been a success by any measure.

UPDM 1.0 Finalization Task Force (FTF). As mentioned earlier, OMG specifications go through a finalization phase to iron out any bugs and to allow for additional comments prior to official release. During this phase, issues can be submitted by the public and need to be addressed by the FTF team. Official ballots are held during the finalization phase to allow members of the team to review any issues and vote on the proposed resolution. This phase of the submission process is far more formal and accurate records need to be kept of all issues, their resolutions, the result of the vote taken, and the impact on the written specification. Changes done to the model were relatively easy to track; however, as the specification had been converted to Framemaker and reformatted, it was impossible to perform an automated difference on the specification. This meant the changes to the specification had to be done by hand, ensuring that change bars were generated in order for people evaluating the specification to evaluate the true impact of the changes. Nonetheless, some automation was possible. The official ballots were automatically generated from the issues list using mail merge. In addition, the use of the Wiki at this stage allowed documents to be easily accessed by the team members. This was especially important as some of the ballots were several megabytes in size making it difficult to email.

UPDM 2.0. The work of the UPDM Group did not stop with DoDAF 1.5 and MODAF 1.2. DoDAF v2.0 was released in 2009 and UPDM 2.0 is being developed to maintain exchange compliance. The NATO Architectural Framework (NAF), which is very similar to MODAF v1.2, will also be addressed. 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. For further information on UPDM, see OMG (2008a) and visit the UPDM website: www.UPDMG.com and the OMG website: www.OMG.org. We have been further improving our means of working during UPDM phase 2. MODAF 1.2 and DoDAF 1.5 were similar in that both shared a common means of expressing the meta-model and were still very similar in their concepts. DoDAF 2.0 has significantly diverged from MODAF 1.2 in that it has fully adopted the concepts of the IDEAS group foundation elements. IDEAS is the International Defence Enterprise Architecture Specification for exchange. The purpose of the project is to develop a data exchange format for military Enterprise Architectures and to allow seamless sharing of architectures between the partner nations regardless of which modeling tool or repository they use. It was developed by an international group of computer scientists, engineers, mathematicians, and philosophers under defense sponsorship. 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 comms 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.,

It has been developed by the IDEAS Group, which is a consortium of Australian, Canadian, Swedish, UK and USA defence ministries. For further information on the IDEAS Group see http://www.ideasgroup.org or http://en.wikipedia.org/wiki/IDEAS_Group

In addition, DoDAF 2.0 has a formal review process resulting in modifications to the DoDAF 2.0 specification and meta-model on a regular basis. To help the UPDM Group keep track of the changes to the specification and their impact on the UPDM meta-model, requirements management tools are being used across the different models. This is especially complex because the DoDAF 2.0 DM2 is modeled in yet another modeling tool. This requires traceability links to be analyzed on a regular basis and impact reports to be generated. The work is ongoing and is expected to finish in September 2010.

During the past year, UPDM tools have become available and are being evaluated by military and non-military projects throughout Europe and North America. The UPDM Group will continue to report on its progress and use of the standard and its use of MBSE practices to ensure success.

References

Brooks, Frederick P., 1995, The Mythical Man-Month, Essays on Software Engineering Anniversary Edition, Addison-Wesley Publishing company,

Bruns, A. , Humphreys, S. (2005), Wikis in Teaching and Assessment: The M/Cyclopedia Project, Retrieved November, 8, 2005: http://snurb.info/files/Wikis%20in%20Teaching%20and%20Assessment.pdf

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

Gilb, Tom, Planguage Concept Glossary, available online at http://www.gilb.com/tiki-page.php?pageName=Competitive%20Engineering%20Glossary (Accessed March, 2010). A complete list of his numerous publications is available at this site.

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., Hause, M.L., 2008, System Development With Virtual Teams In Collaborative Development Environments, In Proceedings of INCOSE 2008, Utrecht, Holland, 16 - 19 June 2008

Hause, M.C., 2009, An Overview of UPDM – a Unified Profile for DoDAF/MODAF, 5ème Conférence Annuelle d'Ingénierie Système AFIS 2009, NOVOTEL ATRIA Charenton, PARIS, 23-25 September 2009

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

International Council on Systems Engineering (INCOSE), Systems Engineering Vision 2020, Version 2.03, TP-2004-004-02, September 2007

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.

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].

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

Object Management Group (OMG), 2008a, Unified Profile for the Department of Defense Architecture Framework (DoDAF) and the Ministry of Defence Architecture Framework (MODAF) OMG Document: c4i/2008-08-02.

Object Management Group (OMG), 2008b, The OMG Hitchhiker's Guide A Handbook for the OMG Technology Adoption Process, Version 7.7, OMG Document: omg/2008-08-01

Object Management Group (OMG), 2009a, Unified Profile for the Department of Defense Architecture Framework (DoDAF) and the Ministry of Defence Architecture Framework (MODAF) OMG Document: c4i/2009-09-01

Object Management Group (OMG), 2009b, Object Constraint Specification, Available at http://www.omg.org/cgi-bin/doc?formal/2006-05-01

Pareto, Vilfredo; translated by Page, Alfred N. (1971), Translation of Manuale di economia politica ("Manual of political economy"), A.M. Kelley

Tal H., Tal E. (2006a) "Wiki – Tutor Assignment: A Collaborative Assignment in a Wiki environment", in: Eshet, Y., Kaspi, A. & Yair, Y. (eds.). The Working Individual in the Technological Age: A Collection of Articles Presented at the Chais Conference for Learning Technologies Studies, March 2006. 43-51 (Hebrew).

Tal-Elhasid , Meishar-Tal (2006b), “Wiki as an Online Collaborative Learning Environment at the Open University”, Mashabei Enosh (Human Resources), 6-225, 48-53 (Hebrew).

BIOGRAPHY

Matthew Hause, Chief Consultant at Artisan Software Tools and Co-Chair of the UPDM Group

Matthew Hause is Artisan’s Chief Consulting Engineer, is a member of the OMG SysML specification team, and the co-chair of the UPDM group. He has been developing complex 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. 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 University of New Mexico and Computer Science at the University of Houston, Texas. He has lived and worked in five different countries and speaks several languages with varying degrees of success. In his spare time he is a church organist.