

# Journal of Enterprise Architecture

February 2009 | Volume 5, Number 1

## Features

- Architect's Spotlight: *Alice Cave* 4
- Architect's Toolkit: *Review of No Magic® MagicDraw™ with DoDAF Plug-in*: Lawrence McCaskill and Ian Komorowski 6

## Articles

- An Information Model for Managed Application Landscape Evolution* 12  
Sabine Buckl, Alexander M. Ernst, Florian Matthes, Christian M. Schweda
- Process-Centric Enterprise Architecture* 27  
Paul Stamas
- A Contingency Approach to Enterprise Architecture Method Engineering* 36  
Christian Riege and Stephan Aier

## Case Study

- Using Metamodels to Improve Enterprise Architecture* 49  
Lyn Uzzle

A quarterly publication of the Association of Enterprise Architects  
*An International Forum for Enterprise Architecture*

## Review of No Magic® MagicDraw™ with DoDAF Plug-in

By Lawrence McCaskill and Ian Komorowski

### INTRODUCTION

In previous Architect's Toolbox articles we have described three basic categories of software tools that fall into the EA domain-precision (focusing on a particular subset of EA), all-in-one (can capture all of the elements needed for a complete EA), and repository management (focused on the analysis of EA data). No Magic's MagicDraw is an "all-in-one" tool specifically focused on a Unified Modeling Language (UML) implementation of the Department of Defense Architecture Framework (DoDAF). It supports DoDAF V1.5, and will likely support the DoDAF 2.0 and the UML Profile for DoDAF and MODAF (Ministry of Defence Architecture Framework (UPDM – see: <http://www.updm.com/index.htm>) soon after these standards are released.

The core MagicDraw enterprise tool is a business process, architecture, software and system modeling tool designed facilitate end-to-end analysis and design of Object Oriented (OO) systems and databases. Its target users include: Business Analysts, Software Analysts, Programmers, QA Engineers, and Documentation Writers. It provides full round-trip support for Java, C++, C#, CL (MSIL) and CORBA IDL programming languages, as well as database schema modeling, DDL generation and reverse engineering facilities. Using MagicDraw's Teamwork Server, multiple developers can work simultaneously on the same model. It interfaces with another No Magic tool called Cameo SUITE, which appears to provide a means of managing requirements via linking objects (documents, models, etc.), much in the way IBM® Telelogic® DOORS® or IBM Rational® RequisitePro® allow for requirements management.

The DoDAF plug-in supports all work products and views in the DoDAF specification (A complete description of the DoD Architecture Framework can be found here: <https://dars1.army.mil/IER/index.jsp>). It is

an extension of MagicDraw's SysML plug-in, and thus, in addition to the DoDAF products, allows creation of the entire list of System Modeling Language (SysML) diagrams, as well as the ability to use the Business Process Modeling Notation (BPMN), which DoDAF practitioners often use in lieu of an Event Trace diagram in the creation of the DoDAF OV-6c Operational Event/Trace work product.

The product evaluated was the downloadable evaluation copy of No Magic MagicDraw Enterprise V16.0 with SysML and DoDAF plug-ins (the DoDAF plug-in requires installation of the SysML plug-in). This review will focus on MagicDraw from the perspective of being able to develop DoDAF architectures in support of the Joint Capabilities Integration and Development System (JCIDS) architectures, which the DoD uses to show how individual platforms (airplanes, ships, ground vehicles) fit into the overall scheme of things from an information technology perspective.

### INSTALLATION AND SETUP

Installation for the MagicDraw Enterprise tool and the SysML and DoDAF plug-ins required about 380MB disk space. The MagicDraw web site recommends a Pentium™ 4 1.4GHz and 1GB RAM for NT/2000, and 2GB RAM for Windows Vista. Other operating systems supported included Linux and Mac OS X. We used Lenovo T61p Thinkpads with 3GB RAM for the evaluation.

Installation was relatively straightforward: install the core MagicDraw tool, then unzip the SysML and DoDAF plug-ins into the "MagicDraw UML" directory (under Program Files). Unlock keys for the core MagicDraw product as well as the SysML and DoDAF plug-ins had to be placed on the hard drive (I put them in the "MagicDraw UML" directory. The e-mails automatically sent to the registered user upon downloading the plug-ins and containing the keys explained the

process reasonably well. The software looked for these keys upon loading, and once pointed to the file locations never asked for them again. One issue with the free demo download – it only allowed for 20 classes to be created, which were all used by the DoDAF plug-in upon startup to stereotype different classes as different diagram/object types in DoDAF. A call and an e-mail to No Magic solved the problem via updated unlock keys. We suspect that this may be “on purpose” to force engagement with the No Magic sales staff.

## INSTALLATION AND SETUP

Installation for the MagicDraw Enterprise tool and the SysML and DoDAF plug-ins required about 380MB disk space. The MagicDraw web site recommends a Pentium™ 4 1.4GHz and 1GB RAM for NT/2000, and 2GB RAM for Windows Vista. Other operating systems supported included Linux and Mac OS X. We used Lenovo T61p Thinkpads with 3GB RAM for the evaluation.

Installation was relatively straightforward: install the core MagicDraw tool, then unzip the SysML and DoDAF plug-ins into the “MagicDraw UML” directory (under Program Files). Unlock keys for the core MagicDraw product as well as the SysML and DoDAF plug-ins had to be placed on the hard drive (I put them in the “MagicDraw UML” directory. The e-mails automatically sent to the registered user upon downloading the plug-ins and containing the keys explained the process reasonably well. The software looked for these keys upon loading, and once pointed to the file locations never asked for them again. One issue with the free demo download – it only allowed for 20 classes to be created, which were all used by the DoDAF plug-in upon startup to stereotype different classes as different diagram/object types in DoDAF. A call and an e-mail to No Magic solved the problem via updated unlock keys. We suspect that this may be “on purpose” to force engagement with the No Magic sales staff.

## OVERALL ARCHITECTURE

MagicDraw is a thick client tool and the tool we reviewed creates a file for each project, within which it stores artifacts. With multi-user versions, we’d expect that it would allow for

creation of these project files in shared spaces, but that was untested. The multi-user version employs record-locking to allow for multiple users to simultaneously access different artifacts within the tool.

MagicDraw has an additional cost add-on called RConverter that will allow one to import IBM Rational Rose diagrams into MagicDraw directly. It is important to note that the importing of comma separated variable artifacts is also a for-cost add-on; most tools include this at no cost.

## Customizability

Magic Draw enables customization via what it calls “Domain Specific Language” – a model-driven approach based on UML profiling. This allows for the ability to adapt domain specific profiles to create custom diagrams, custom specification dialogs, custom real-time semantic rules, and other customizations saved in profiles within the project.

## Capability and Use

From a “pointing and clicking” perspective, we were pleasantly surprised at the ease with which one can create and navigate between artifacts within the tool, and how quickly one could learn to use the tool. No tool is perfect, but this one is better than many that we’ve used. From a DoDAF perspective, it’s not particularly scalable, but we believe this to be a limitation of UML vs. a limitation of this particular tool; this is elaborated upon below. A more detailed discussion of our user experience follows.

With our unfamiliarity with the tool, we downloaded and read the “DoDAF Plug-in User’s Manual” provided by No Magic at: [http://www.magicdraw.com/main.php?ts=navig&cmd\\_show=1&menu=dodaf](http://www.magicdraw.com/main.php?ts=navig&cmd_show=1&menu=dodaf).

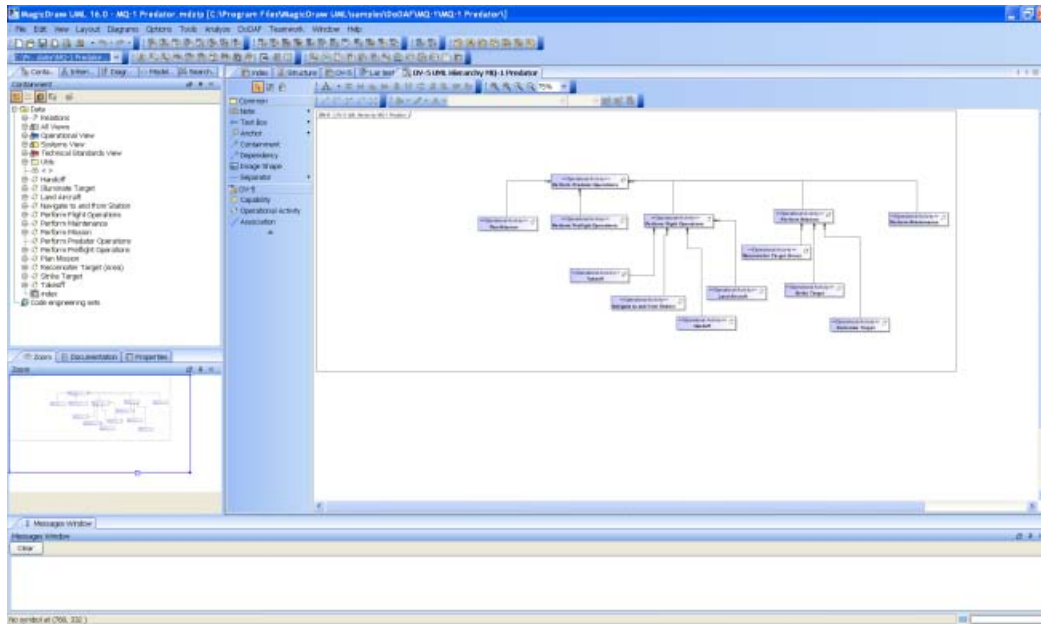
It’s actually very informative, and helps one understand how they have chosen to implement DoDAF in the UML/SysML notation using the tool. Also useful are No Magic’s online tutorials at: <http://school.nomagic.com/mdumltips>.

Upon initially opening the tool, MagicDraw presents one with an initial navigation pane (“Welcome screen”), within which you can Manage Projects, see “Latest News” from No Magic and use the “What’s New” link to view release notes for the version loaded. There is also a “Resources and Plug-ins” link and a “Samples” link. It’s a pretty clean opening page,

but those working on “one and only one” project can bypass it and get directly to work through settings under Options=>Environment.

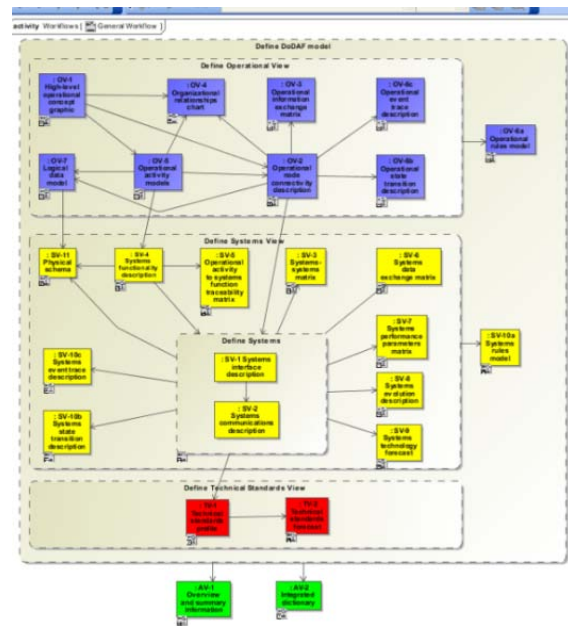
When a user opens a new or saved project (including samples provided by No Magic), the tool presents itself in a fairly standard fashion:

there is a navigation pane on the left, an outline view tool under it (shows you “where you are” in the diagram and other details about the object selected in the browser, etc.), tool icons across the top, a editing/work area to the right, and a messages area below.



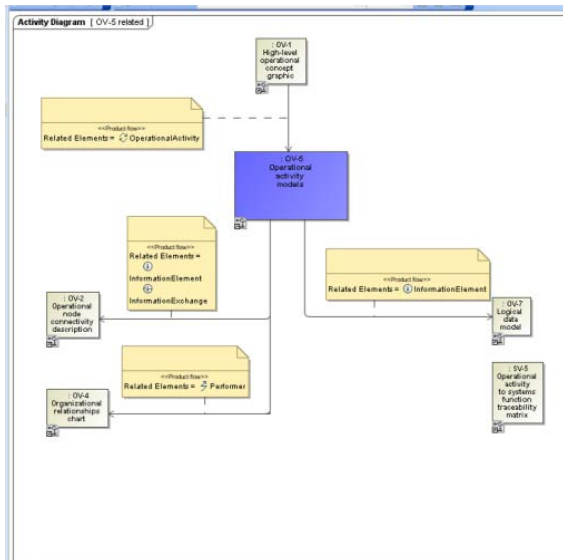
Something that’s not intuitively obvious, but nonetheless helpful: in the browser on the left, if the user navigates to the “Index” object under “Data” and double-clicks it, one is presented with a Content diagram depicting two objects: a “Structure” object, and a “General Workflow” object.

Double clicking on the “General Workflow” object presents one with “a means” (and a “not bad means” at that...) of working through the set of products within DoDAF. Using the “fit for purpose” mantra being socialized within the draft DoDAF 2.0 documents, one only creates products necessary for the purpose for which they’re creating the architecture, but this provides a good roadmap for product creation. If you’re going to be making regular use of the tool, printing it, and pinning it to your wall isn’t a bad idea.



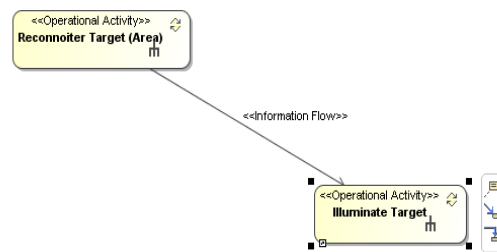
Double clicking on any of the elements within this diagram takes you to the metamodel view within MagicDraw of the product (in this case,

the OV-5 Operational Activity Model work product):

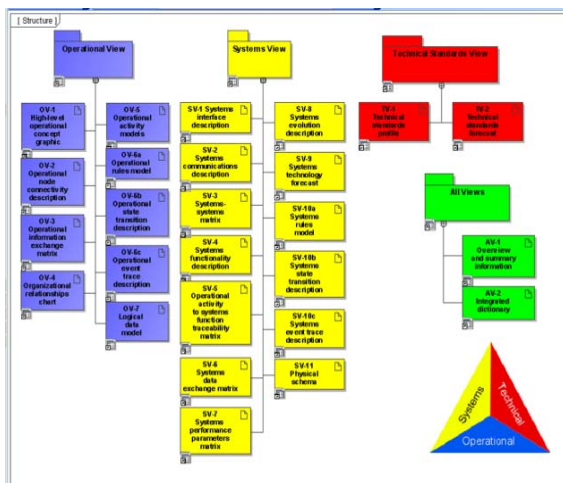


however, even being intimately familiar with DoDAF, we found these useful.

Another useful feature: while creating the artifacts (in this case, an OV-5 Activity Diagram - Note that this is the MagicDraw term, due to the use of a particular UML diagram, as opposed to the work product name (Operational Activity Model) in the DoDAF specification. ), when one selects an object (in this case, and Operational Activity), icons appear to the right that are context-sensitive (i.e., the only objects that appear are ones that the object selected can connect to within the notation). This allows for quick creation of model artifacts.



Returning to the Index Diagram, and double clicking the “Structure” object presents one with a hierarchical view of the DoDAF.



Reporting within MagicDraw appears to be very robust, and all of the DoDAF matrix-based products (OV-3, SV-6) are created via report. The only potential problem with the OV-3/SV-6 reports is they come out in RTF (vs. CSV or MS Excel format) with multiple columns' information put into single columns; this could be problematic when trying to combine data created by MagicDraw with other tool vendors' OV-3/SV-6 solutions.

Quirks (all tools have them):

- The auto save function “takes over” the machine when it begins executing; not a good thing when one is presenting multiple artifacts to a customer in different formats (PowerPoint, etc.).
- One must be careful when creating diagrams that the correct container is highlighted; otherwise, it dumps the artifact under the top-most “Data” container.

This is useful. Something even more useful is found by double-clicking one of the products. One is presented with a description of the artifact from DoDAF and applicable diagram types within MagicDraw that can be applied to the product. These containers allow for one to drag-and-drop artifacts into them from the browser, and present the artifacts as a shortcut icon to the diagram. For those familiar with DoDAF, they may want to directly create the artifacts without using these road mapping tools;

## CONCLUSION

In the endgame, MagicDraw is relatively easy to learn, and appears to have implemented DoDAF in UML as called for in DoDAF 1.5. Specifics on the DoDAF implementation follow

Something that was relatively unique (and we thought appropriate...) is MagicDraw's implementation of the OV-5; it implements the OV-5 hierarchy in a Class Diagram, and uses the UML Activity Diagram to implement Activity Flow at the leaf (i.e., lowest) levels in the Class Diagram. However, it wasn't intuitively obvious how to make these associations in the tool beyond naming the artifacts the same thing (i.e., there was no way to drag-and-drop activities created in the node tree format [called "Operational Activity Hierarchy" within Magic Draw] from the browser). Within the DoDAF Sample provided, they tried "really hard" to make the UML Activity Diagram look like an IDEF0 diagram; however, UML Activity diagrams are more akin to Business Process Modeling Notation (BPMN) or IDEF3 process flows. These notations are designed to show process in the form of ordered activities, and (assuming swim lanes are added) who accomplishes the activities. The only major problem with this is depicting information exchange between the activities. However, we believe this to be more a problem with UML than MagicDraw's DoDAF implementation using it – more below.

Information Exchanges: two of the major products in DoDAF, the OV-3 and SV-6, are primarily concerned with information/data passing between Organizations, People/Systems, and the Activities they're accomplishing. There are a couple of potential problems here, MagicDraw has provided "a means" (not necessarily the best means, but "a means") of working around it:

- OV-2 within MagicDraw is accomplished via using a Class Diagram, with Classes stereotyped as Operational Nodes, and Associations stereotyped as Needlines. Adding Information Exchanges to the Needlines is a completely manual process, and does not use any information from MagicDraw's OV-5 Activity Flow Diagrams or OV-6c Sequence Diagrams. The good news is that the OV-3 reports on information in the OV-2 diagram, but more automation here would have been a plus

- SV-6 is a report on the SV-1, which is created similarly to the above; System Data Exchanges are manually created within the diagram.
- However, these "problems" are endemic to the UML notation; we're already hearing a collective gasp and anticipating hate mail from the UML zealots out there. That being said, before composing this retort, consider the following: UML is designed to help software architects conceptualize the problem space using the concepts of abstraction and polymorphism, and is not designed to report on information exchanges across boundaries (it's good at what it's designed to do... it's just not good at looking at information exchange between Classes/Objects...). However, the Unified Profile for MoDAF and DoDAF (UPDM), which creates over 100 stereotypes to model MoDAF/DoDAF in UML/SysML, is an attempt by the community to alleviate these and other problems implementing MoDAF and DoDAF using UML (see <http://www.omg.org/cgi-bin/doc?c4i/2008-08-13> for the UPDM specification).

It may look like we're hammering MagicDraw here – we're not... most tools have warts, and that's what we've documented. We've also documented several areas where No Magic has done an extremely good job of implementing a tool for creation of UML/SysML artifacts for DoDAF – overall they've done a good job - it's a good tool.

### Quick Scorecard: (0-4)

4	Overall Ease of Use
3	Coverage of EA Modeling Needs
4	Adherence to Model Standards
3	Automation of Model Build/Edit
4	Appearance and Readability of Models
4	Use of Model Data Internally
2	Use of Model Data Externally (interfaces)
3	Reports Included
3	Reporting Writing Capability
3	Stability
2	Use on a Large EA Program
3	Customizability
<b>38</b>	<b>Total (out of 48)</b>

0=no applicability; 1=Poor; 2=Fair; 3=Good; 4=Excellent

## ABOUT NO MAGIC™

Victoria S. Girdziunas and Paul T. Ducanson founded No Magic in July of 1995 with the vision that there is no magic to develop better software. No Magic, Inc. is a Wyoming Corporation, with software development facilities are located in the EU (Kaunas, Lithuania) and Thailand (Bangkok). No Magic's Corporate and Sales headquarters are located in Plano, Texas. MagicDraw v1.0 was released in 1998, and its current release at the time this review was written is v16.0. The product has received several awards, including the Jolt Productivity Award (2007, 2005) and the Java Developer's Journal Reader's Choice Award (2006, 2003).

No Magic is a member of OMG (Object Management Group, a standards organization developing UML, CORBA, MDD and other standards), CodeGear and Sun Technical Partner.

### Corporate Headquarters:

7304 Alma Drive, Suite 600  
Plano, TX 75025  
Phone: +1-214-291-9100  
Fax: +1-214-291-9099  
E-mail: sales@nomagic.com  
www.nomagic.com

## TOOL COST

For a single-seat, undiscounted license of MagicDraw Enterprise with SysML and DoDAF plug-ins (DoDAF plug-in requires installation of the SysML plug-in...), plus maintenance for a year, the cost is on the order of \$4500. There will likely be discounts available for government, multi-user licenses, etc., but expect to pay on the order of \$4-5k per seat.

The cost for the "Applying DoDAF with MagicDraw" course is \$2000. However, before attending, we strongly recommend a tool-neutral course in DoDAF; these are offered by a number of organizations, including:

- WBB (<http://www.wbbinc.com/architecture-training.html>)

- AFCEA (<http://www.afcea.org/>)
- FEAC Institute (<http://www.feac institute.org/>)

## AUTHOR BIOGRAPHIES

Lawrence McCaskill is Chief Enterprise Architect at WBB Consulting®, a technical and management consulting firm ([www.wbbinc.com](http://www.wbbinc.com)). At WBB, he has overseen or created over 40 architectures supporting the Joint Capabilities Integration and Development System (JCIDS). Of note are the E-2C/D, F-22A, MV/CV-22 Osprey, Joint Light Tactical Vehicle, and the MQ-1 Predator/MQ-9 Reaper, and several Net-Enabled Weapons architectures. In prior architecture-related positions, he developed architectures depicting net-centric operations and Global Information Grid (GIG) 2.0 Architecture. He was Chief Architect at Headquarters Air Force Special Operations Command and part of the initial cadre that developed the Activity Based Modeling methodology. Mr. McCaskill teaches the WBB tool-neutral DoD Architecture Framework course and currently participates in the DoDAF Working Group. He holds a Bachelor's degree in Computer Engineering from the University of Miami, FL, and a holds a Master's degree in Software Engineering from the University of West Florida. He has also authored/co-authored three papers for the International Command and Control Research and Technology Symposium.

Ian Komorowski is the Manager of Federal Architecture Requirements at WBB Consulting®, a technical management and consulting firm ([www.wbbinc.com](http://www.wbbinc.com)). Ian has a wide array of experience in developing architectures for commercial and government customers. Ian has also developed and taught a variety of classes on enterprise architecture among which include: tool neutral DoD Architecture Framework, Telelogic System Architect DoDAF ABM, and Department of Health and Human Services EA Modeling (based on a custom HHS framework implemented in Metis) classes. He can be reached at 703-448-6081 x265 or [ikomorowski@wbbinc.com](mailto:ikomorowski@wbbinc.com).