

# Journal of Enterprise Architecture

November 2009 | Volume 5, Number 4

## Features

Architect's Toolkit: *Review of Artisan Studio™ with DODAF Profile* 4  
Lawrence McCaskill

Book Review: *Waltzing with the Elephant* 14  
Mark Toomey

## Articles

*Service Oriented Enterprise Architecture,  
A Conceptual Design Through Data Architecture* 16  
Tyson Brooks

*The Service Oriented Architecture Method  
For Federated Architectures* 27  
Thomas Mowbray, Grant Chanachote, Bruce Fenchel, and Mark Odell

*Metaphor as an Inference from Sign* 35  
Barie Fez-Barrington

## Case Study

*Skills Analysis for Enterprise Architects: Implications for  
University Education and Curriculum Design* 44  
Keith Frampton and Shuk Ying Ho

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

## Review of Artisan Studio® with DoDAF Profile

By Lawrence McCaskill

### Introduction / Tool Overview

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)
- Repository Management (focused on the analysis of EA data)

Artisan Studio® is an "all-in-one" tool specifically focused on a Object Management Group's (OMG) Unified Modeling Language (UML) and Systems Modeling Language (SysML) implementation of the Department of Defense Architecture Framework (DoDAF). It supports DoDAF V1.5, and will soon support the DoDAF 2.0 and the UML Profile for DoDAF and MODAF (Ministry of Defence Architecture Framework) (UPDM, <http://www.updm.com/index.htm>), since Artisan is a high-profile member of the UPDM consortium.

According to the website: "...Artisan Studio® is an all-in-one integrated development tool suite which provides systems and software modeling and component based development targeted for technical systems. It is the tool of choice for complex mission-critical systems and software engineering. Artisan Studio provides comprehensive support for the leading industry standards, including OMG SysML, OMG UML and Architectural Frameworks. Studio delivers on the promise of an integrated collaborative development environment – allowing systems and software engineering teams to work as one – from concept through to delivery and maintenance..."

The version reviewed was 7.0.22 with patches a-f. It supports all products within the views (Operational, Systems, Technical Standards, and All View) in the DoDAF V1.5 specification (<https://dars1.army.mil/IER/index.jsp>).

Additionally, it has the capability to integrate with IBM DOORS and MathWorks Simulink; these capabilities were not reviewed. They have both a node-locked and a server-based licensing scheme; the node-locked (i.e., single, detached user) version was used during this review.

The review is of the DoDAF profile, and hence, I'm reviewing it from a DoDAF practitioner's perspective, specifically from the perspective of describing a platform or system, as it relates to the larger "enterprise." Despite all the talk of "Enterprise Architecture," in the DoD, this type of architecture creation (the platform as the "center of the world") has become the most prevalent type of architecture because the Joint Capabilities Integration and Development System (JCIDS – the process by which the DoD procures nearly everything from aircraft, ships, and ground vehicles to sea- and shore-based systems in the DoD) requires them as part-and-parcel of the description of the capability to be procured by the DoD. The tutorials (described below) took this perspective as well. In fact, most tools are focused in this manner; the fact that most tools suffer from "severe myopia" and "don't integrate worth a darn" when one tries to integrate these types of architectures at the "enterprise" level could be the subject of another paper, but I'm not looking at this perspective here. So... ground rules are: I'm looking at how well this tool describes artifacts related to a platform as the "center of the world."

### Installation and Setup

I cannot vouch for the ease of setup for the server version, but installation of the laptop-only version of Studio was straightforward. All files, including the tutorial files, and example files took on the order of 315 MB. Minimums from the installation guide were listed as:

- 3 GHz P4 processor, a Core 2 Duo processor or a Core 2 Quad processor

- 1 GB recommended (2 GB on Windows Vista)
- SVGA graphics, 1024x768 minimum, 256 colors

It ran perfectly fine on my IBM ThinkPad T61p with 3GB RAM; I expect it will run similarly on any modern laptop equipped with a decent amount of memory and disk space (note to your network administrator: please give the architects your most capable machines...).

Also useful are Studio's online tutorials at: <http://www.artisansoftwaretools.com/download-center/product-guides/> - the tutorials I used during this review included the Artisan Studio Tutorial, the Artisan SysML Tutorial, with primary focus on the Artisan DoDAF/MODAF Tutorial. There was an additional tool Artisan calls the "Real-time Perspective Mentor," which was a content-rich tutorial environment:

- Lots of information across the gamut of systems and software engineering, including overviews of evolution of development methodologies (Software Development, Real-time Systems, Real-Time perspective), Modeling (Requirements Architecture, Solution Architecture, Supplementary Models), Process for project development (steps relating Systems Engineering to Software Engineering, including a BPMN-like "dive down" describing each step in the Systems/Software Engineering process), Development Issues (Reuse, Performance, Requirements Validation, and Round Trip Engineering), and Management (Process Management, Configuration Management, Project Management, Quality Management)
- Uses Internet Explorer to keep track of where you are – somewhat disconcerting in some places. Example: Management takes you to a "select one of 4" panel (my term, not theirs...), select Configuration Management (which has only one panel), and doesn't have a capability to go back to the parent to select the other 3 without navigating back by using the back button, or going to the Management tab again
- Has a wiki-ish feature that allows for extension of the material; a means of keeping the information relevant. However, the assumption is that there's a moderator in Artisan-land that can disapprove any change that isn't relevant.

Other than a "more than a few" grammar errors (easily solved via a technical editor - these were not a case of U.S. vs. British English; the tool has roots in both "Olde England" and the good old USA), the suite's tutorial and help facilities are very robust, and I found to be very helpful.

### Customizability

Studio enables extensive customization through Profiles, Stereotypes and Tag Definitions. These provide a controlled way of extending the properties and linking capabilities of Studio. A Profile is a coherent collection of Stereotypes and Tag Definitions that can be applied to a Model. After applying a profile to a model, its Stereotypes and Tag Definitions can be applied to items in that Model. These were not examined in detail; however, the DoDAF/MODAF Profile reviewed here was one instance of an extensive modification to the base tool for the purpose of accomplishing architectures complying with the DoDAF/MODAF specifications. Therefore, it is obvious that the behavior of the tool is extremely customizable. However, the user is repeatedly warned against modifying the DoDAF profile, so there's perhaps a double entendre here.

### Capability and Use

Each tool has its own peculiarities in paradigm of "pointing and clicking." Artisan is no different in this respect. In going into the tool, the recommended paradigm is to go into the "Artisan Models Neighborhood" tool and select the model you want to edit. The Models Neighborhood appears to be what one would expect to see after entering the tool, and hitting an "open model" dialog. It's essentially a pointer to the different project containers one has created using Studio (i.e., each project container contains one-or-more projects). That being said – one can enter directly into the tool and use the open paradigm.

The tool is conformant to DoDAF v1.5. However, Artisan is a major player in the UPDM (Unified Profile for MODAF and DoDAF – a profile extending the Unified Modeling Language [UML] and Systems Modeling Language [SysML] for use as a means of expressing MODAF/DoDAF diagrams using UML/SysML).

Thus, I expect they'll put out a DoDAF v2.0-compliant version when UPDM v2.0 is released (it is currently in review).

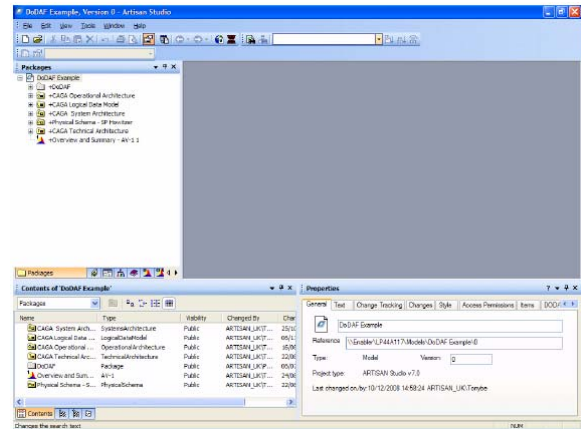
That being said, they take some liberties with the DoDAF in naming their product containers. A concrete example of this is their naming of OV-5 artifacts. They've named the artifacts OV-5a, b, and c. The SV-4 follows the same paradigm. The divisions there are somewhat unintuitive:

- **OV-5a/SV-4a:** functional decomposition diagram
- **OV-5b/SV-4b:** sequence diagram using swimlanes
- **OV-5c/SV-4c:** unattributed sequence diagrams (i.e., merely sequences activities, and doesn't attribute them to roles or systems that perform them)

This becomes a problem in DoDAF as follows:

- **DoDAF v1.5:**
  - OV-5: doesn't divide this artifact into OV-5a, -b, -c.
  - SV-4: divides products into SV-4a and SV-4b based on whether it's a systems function (SV-4a) or services function (SV-4b)
  - It's another argument entirely as to whether or not this makes sense in DoDAF v1.5 – however, it's definitely different than the above
- **DoDAF v2.0**
  - OV-5: divides the artifacts into an OV-5a (functional decomposition – they call this a decomposition tree) and an OV-5b (Activity Model)
  - SV-4/SvcV-4: doesn't divide the models into the –a and –b model paradigms used with the OV's
  - All problems with DoDAF v2.0 aside, it becomes a problem in explaining products created in the Artisan Studio DoDAF profile to a person familiar with DoDAF, but unfamiliar with how Studio has chosen to “extend” DoDAF to suit the needs of their modeling paradigm

Once into the tool, the interface is pretty standard for modeling tools. There's a browser-type interface on the left side, the modeling pane on the right side, and a results pane below.



Across the bottom of the explorer pane are icons for DoDAF Artifacts, as well as DoDAF primitives (objects you use to create the DoDAF artifacts) in “O,” “S,” and “T” panes, corresponding to Operational, System, and Technical Views, respectively. In several cases – Studio calls these the Operational/Systems/Technical Elements panes, and rather than creating the OV, SV, and TV artifacts in their panes, one creates these from the “Elements” panes in many cases.

Most standard Windows shortcut keys of, and the Microsoft Office paradigms for manipulating objects have been followed. We'll get into diagram types later, but notable exceptions to using Windows Standards:

- **Text Diagram:** doesn't allow an undo, and after pasting, only allows one to change fonts. No tools in this diagram – pretty limited... basically a container to paste objects into that you don't want changed.
- There's no redo key, for undoing an undo.
- One cannot multi-select objects in the browser (shift-click, or control-click). Can do this in a “Contents Pane,” but it is an extra step.
- When creating an object, there's not a “pick list” that you can choose from; only option there is to find it in the browser and drag-and-drop.

Other point-and-clicking notes:

- Sequence diagram implementation is clever; one can copy the text from an associated Use Case diagram, or create it on the diagram. Either way, the “book solution” forces one to create text describing the operations. It doesn't let



**All Views (AV's):** unlike the Operational and Systems View artifacts, Studio doesn't separate the AV's into a separate pane.

AV-1 Overview and Summary Information: a text diagram. Can be anything one can cut-and-paste into the document. This is a "Good news, bad news" situation: the good news is you can enter anything into the document. The bad news is you can enter anything into the document... One can possibly implement the DoD Architecture Repository System (DARS) AV-1 template here, but it's not a direct import/export.

AV-2 Integrated Dictionary: In the tutorial, it states "the AV-2 is the list of elements throughout the browser tabs." With the reporting capability available in Studio – this is simply unsatisfactory; the AV-2 should be an out-of-the box report. As previously discussed, Studio has two panes for "Elements" (Operational and Systems), as well as a separate Dictionary pane. Would recommend that in addition to creating a report, that they create an AV pane to provide both the AV-1 and AV-2 information in a single pane.

### **Operational Views (OVs):**

OV-1 High-Level Operational Concept Graphic: is a "Text Diagram" within which one can stuff about anything you can get on the clipboard. Text diagram doesn't have an undo capability. Thus, the OV-1 and its associated text should be created in a separate tool, and stored in Artisan Studio to keep the artifacts together.

OV-5 Operational Activity Model: Artisan Studio divides the OV-5 into OV-5a, -b, and -c. This differs from DoDAF, both v1.x and V2.x, and is thus an "invented manifestation" of Artisan. Studio doesn't appear to be any sort of parent/child paradigm for navigating through diagrams, nor does there appear to be any sort of notational marker that would indicate a child diagram exists.

OV-5a: implements Activities as Classes. This is the first time I've seen this done in any tool (it doesn't mean it's "wrong" – it's just "different"). The good news is the Activity-as-Class does map to the Activities in the Activity diagrams. Bad news:

Activities and System Functions aren't reusable as Activities when one goes into SysML.

OV-5c: unattributed sequence diagrams. Artisan implements the OV-5 in a, c, b order, respectively. Not sure why they did this, especially since their OV-5a, b, c nomenclature was self-invented.

Nonetheless, it makes sense that they move from decomposition to a sequence diagram, in that they define what needs to be done in the OV-5a, the order it needs to be accomplished in via the OV-5c, and then attribute the activities to OV-2 Operational Nodes and OV-4 Roles in the OV-5b.

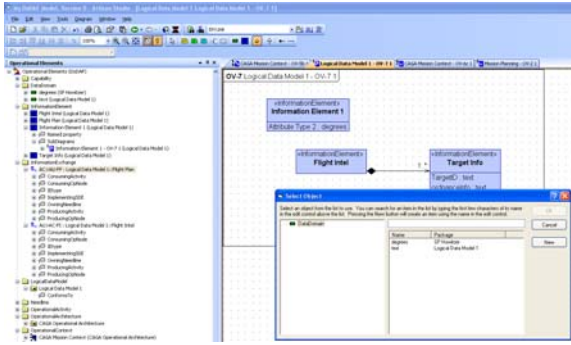
However, In accomplishing the tutorial, I found a couple of problems with the tool: 1) Child diagrams don't automatically populate the input/output parameters from parent diagrams; this would be a nice feature to have as a default. In fact, auto-populating everything under the "Populate" tool as a default would probably be a good thing. 2) Frame doesn't auto-size as objects are placed on the diagram. 3) The in/out pins don't move with the frame as it's resized. Trying to move them to the edge causes error messages about dropping the pin object (Invalid Drop: Activity Parameter items cannot be dropped directly onto the diagram canvas); no idea how to resolve until Artisan fixes this problem. Out pin appears to be attached to an invisible object that's independent of the frame. This needs to be fixed.

As Information Flows are created, they create objects (called InformationElements in the Operational Elements View) made available to the logical data model automatically. This makes sense, and is a good thing.

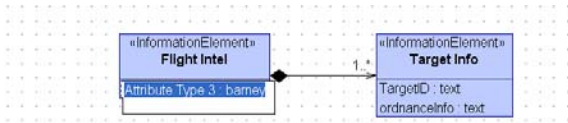
OV-5b will be discussed in the sequence Studio recommends, which is after creating the OV-4 and OV-2.

### OV-7 Logical Data Model:

Allows Information- Elements to be selected for inclusion on the diagram. Attribute creation somewhat counterintuitive.



One selects the attribute button from the toolbar, then selects the <<InformationElement>> associate the attribute with. One then types (as in “define the type of”) the attribute via hand typing the type into the whitespace next to the “Cancel” button hitting “New.” This could use some default data types; none exist within the tool. Additionally, when one hits “New,” (or hits “OK” after selecting one of the user-defined types), the entire attribute, including its type is highlighted.



If you type a name into this in this state, it nukes the typing (as in the assignment of type) that one has so painstakingly been forced to go through. To get around this, one must select the name part of the attribute only, and then type over it, renaming it to what the user intended.

This is also where an auto-size of the frame would be good – as attributes are defined, they can “bleed” outside of the frame, and it doesn’t resize automatically. Ditto for moving objects outside the frame.

The container objects are really classes – not sure why they continue to call them InformationElements, other than that’s what DoDAF calls them. This prevents these objects from being readily useable in a SysML diagram, which is a shame.

**OV-4 Organizational Relationships Chart:**

Interesting extension of Actors. They’ve covered all the relationships normally covered in an OV-4, including abstract organization types, organizations, and command

relationships (Commands, Coordinates, Supervises).

One can impose symbology different than the “stick figures” for Actors (for instance: one could use standard Army symbology for units). Unclear how Supervises relationship is different than the Commands relationship, other than the Commands relationship auto-populates the field for an associated organization (implication: OV-4 Role owns Organization).

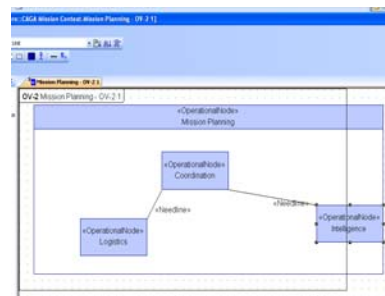
Interesting “book solution” relationships between OV-2 and OV-4. Within OV-4, you create the “Responsibilities” relationship between Organization and OpNode using the right click=>Link tool. However... you can create a “RoleAtNode” relationship within the OV-2 by dropping Roles within OpNodes. Therefore, the implied relationship creation is interesting: Organization owns OpNode owns Role.

Time Vampires: leaving the tool mode in the previous tool selected would be a timesaver (as in: create multiple roles at the same time, then name them as you go). Frame doesn’t auto-resize when objects dropped outside of it.

**OV-2 Operational Node Connectivity**

Description: repurposes a Composite Structure diagram, with Classes stereotyped as OperationalContext and OperationalNode.

The Operational Context frame auto-resizes if one drops objects directly on its border (<<Operational Node>> Mission Planning, in the diagram below). However, the Mission Context Frame does not (OV-2 Mission Planning – OV-2 1, in the diagram below).



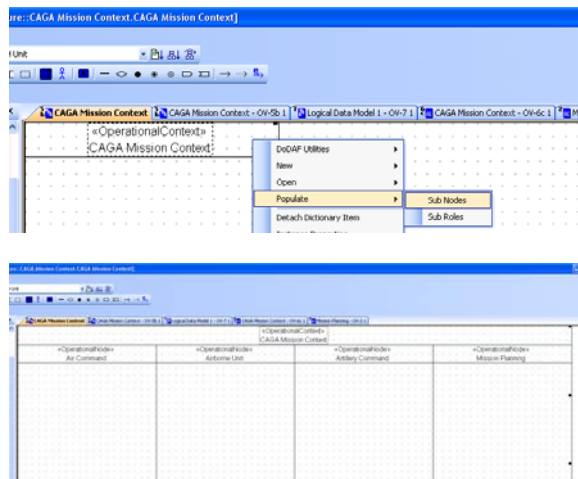
Unlike the Composite Structure Diagram SysML diagram, which allows for expression of external actors, this particular diagram doesn’t allow one to show interactions

between OperationalNodes and external OperationalNodes. This is “very bad” from the “platform as the center of the world” paradigm. One needs to be able to show “what’s in and what’s outside” of the architecture. This can be done diagrammatically (color, etc.), but apparently isn’t part of the Artisan Studio DoDAF profile paradigm. As such, using the OV-2 as the source of the OV-3 means all exchanges between all Operational Nodes are generated. If one has multiple internal subnodes, this can create manual work via having to remove generated exchanges between internal OperationalNodes.

Covers both the abstract and potentially the physical instantiation of OpNode via OperationalNodeType and OperationalNode conventions. However, on the OV-2, the diagram objects are instantiations of the OperationalNode, and the only way to get to OperationalNodeType is via properties of OperationalNode.

OV-5b: implemented via sequence diagrams using swimlanes. Based on the tutorial, it appears that one OV-5b diagram should be created per mission.

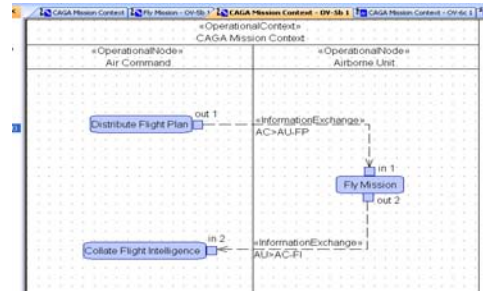
Interesting capability to populate the subnodes and subroles (select the name of the Mission Context lane, then right click => Populate => Sub Nodes [or Sub Roles]; don’t get too happy with the undo here though – it’ll un-create the diagram...).



Alternate method is to create them using the icons on the toolbars; a little unintuitive –

select the tool, right click in the lane, then select from list of created OpNodes.

One then associates the activities with the Nodes and Roles, as well as defining information exchanges between Activities. This is a repurposed SysML Activity diagram.



Of note when creating Roles on this diagram – rather than using the Role itself, it uses a construct it calls RoleAtNode. This allows for reuse of Role names within different nodes, wherein the named Role performs different Activities. Somewhat annoying is the RoleAtNode name isn’t defaulted to be the Role, but I suppose there’s situations where you want these names to be different. If one goes back to the associated OV-2, selects the related nodes, and right click => Populate => Sub Roles, these roles will appear on the OV-2 (however... the diagram will usually have to be resized to accommodate the newly inserted objects; it’d be nice if the tool did this “prettifying” on its own...). One further note regarding Roles: there’s possible dissonance between Roles, RoleAtNode, and UML roles (which are attributed to Classes instead of Actors); I’m not sure how to handle this, but it’s worth noting.

The same method can also be used to populate the Needlines on the OV-2 diagram (right click => Populate => Information Exchanges). Good news is it does it, the bad news is in complicated architectures, there’s limited ways of de-cluttering the diagrams.

Regarding being able to reuse Information Exchanges (say... on an OV-7), Studio makes a delineation between Information Exchanges and Information Elements, stating: “...information exchanges are used to model interaction between nodes, whereas information flows are used to model interaction between activities within nodes or between

activities across sub-nodes...” This is technically correct; information exchange instantiates information element. This manifests itself in the OV-2; when one uses the auto-populate function (select Needline, right click => Populate => Information Exchanges), it shows the “Information Exchange Name: Information Flow Name,” where Information Exchange Name is created on the OV-5b, and the Information Flow Name is taken from the OV-5c. It’s good that it takes this from the diagrams that it auto-populates with; however, I don’t see any value added in creating separate diagrams for the OV-5b and OV-5c.

**OV-3:** the OV-3 is a report automatically generated as a result of artifacts and logical assertions created during the course of creating the OV-2 and OV-5, and generated into Microsoft Excel. Column definitions are user definable; the default columns generated do not match the DoDAF (any version). It would be nice if the columns defaulted the DoDAF suggested template. My assumption is that additional fields can be user defined, such as the Critical Mission Thread called for in CJCSI 6212.01E (a governing instruction for the use of DoDAF within certain DoD processes).

**OV-6 Dynamic Behavior Models:**

**OV-6a Operational Rules Model:** the tutorial states “...it is created in Studio by right-clicking an operational architecture element, then entering the required text or pasting from some other document...” This is misleading. The only place one can create the OV-6a artifact that I saw was on the Operational Architecture that the user creates when initially declaring a “container” for their Operational Architecture. While it’s applicable here; the rules created herein may also apply to lower level architecture artifacts. It’d be nice to have a container for Operational Rules, and be able to associate the individual rules with architecture artifacts that implement them within the model (as well as being able to produce an overall artifact).

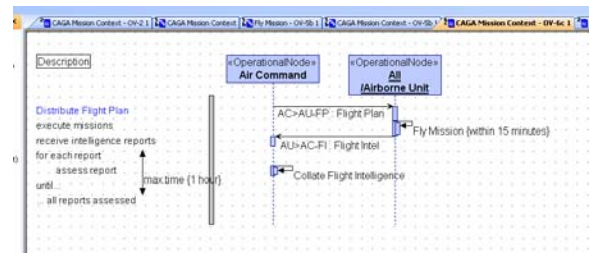
**OV-6b Operational State Transition Description:** tutorial introduces the concept of an Operational Node Type. The creation

tutorial for this diagram didn’t work... the step after creating the initiating event said to click on the “Do Activity” icon, and click in the “Operational” state created in the example, which should yield a pick list for an activity. Since there was no means of associating the “Operational” state with this, it didn’t work. In looking at the browser, the activity it was looking for was indeed associated with the Operational node as a subactivity, so this is a bug.

**OV-6c Operational Event Trace Diagram:** Events created here are 1-for-1 recastings in sequence of information exchanges created in the OV-5b.

The additional nicety is that one can show individual activities on the “pole” – this is generally not shown in most DoDAF architectures, but definitely adds fidelity to the diagram, especially when indications of timing are added to the expression of activities.

The descriptive text on the left is a very good feature.



**Systems and Services Views (SVs):** creation of the Systems and Services views followed generally the same pointing-and-clicking paradigms as the OVs, including the following steps:

**SV-1 Systems/Services Interface Description:** Has the same problem as the OV-2, in that it doesn’t delineate between internal and external nodes.

**SV-4 Systems Functionality Description:** Repurposed SysML Activity diagram. Same SV-4a, b, c paradigm as OV-5. System Data Exchanges created here are not directly useable by SysML; this is unfortunate. This would be a viable means of moving from architecture to Systems Engineering in a seamless fashion.

SV-6 Systems/Services Data Exchange Matrix: as with OV-3, the SV-6 is a Microsoft Excel report generated from underlying data and assertions in the model.

SV-2 Systems/Services Communications Description: divided into 2 types.

SV-2a depicts the communications network view. Comm links can be added between Nodes or Systems.

SV-2b shows the path between two nodes for a given interface. The tutorial didn't cover this notation, and thus it was unreviewed.

**Supplementary Views:** Artisan includes 3 supplementary views not included in DoDAF. These are all valuable additions to the tool.

SPV-1 Allocation Matrix: allocation of Operational Activities to Operational Nodes and Operational Node Types. When for SVs, SPV-1 allows you to view the allocation of System Functions to Systems.

SPV-2 Node to Node Matrix: - allows one to view how Operational Nodes and RoleAtNodes communicate in a matrix form. This is essentially a "check your homework" view.

Capability: allows for diagrams, activities, and systems to be grouped into a "Capability." This manifests itself on the OV side as the context-level activity in the OV-5a, and SV-4a. Interesting behavior: one can only open an OV-5c or another OV-5a from this type of diagram.

**Crashing/Aberrant Behaviors:** I navigated to "Open" under SystemDataElement in the System Elements Pane. Got an error message: Operation on Stereotype 'DoDAF::SystemElements.SystemDataElement' cannot be performed, after which the program crashed.

## About Artisan

Founded in 1997 with extensive venture capital backing, Artisan is headquartered in the USA and UK with offices in Germany and Italy, supported by a global distributor network.

US Corporate Headquarters:  
Artisan Software Tools Inc.  
16055 SW Walker Rd, #422  
Beaverton, OR 97006-4942  
Phone: +1-888-800-1416  
Fax: +1-503-296-2281  
Email: info@ArtisanSoftwareTools.com

## Tool Cost

For a single-seat, undiscounted \$10k for Studio, \$5.5k for the DoDAF Profile and \$3.3k for Annual Maintenance. \$18.8k for a single seat.

I was quoted a one-day one-on-one mentoring for \$2300; the website didn't offer US-based courses – the basic MODAF practitioner course was about \$1800 for a 3-day course. All that being said, before attending, we strongly recommend a tool-neutral course in DoDAF offered by a number of organizations.

## Conclusion

Overall, this is a very good tool with a lot of potential. The automation of population of exchanges and other information to top-level diagrams (OV-2, SV-1) is very powerful, and omitted in many other tools. Artisan is a key player in the UPDM standard, and I expect the tool will improve significantly as UPDM is implemented using this tool. Implementation of DoDAF v2.0 is also on the horizon, so I expect the tool will continue to improve as this moves along. Improvements can be made in the following areas:

- Cost: cost is out-of-line with comparable tools
- Enable Auto-resize of the frame object
- Enable Auto-generation of an AV-2 report
- Creation of an AV Pane
- More ease of use of artifacts created in the DoDAF Interface in the SysML interface; this may be impossible due to stereotypes, but it would create a more seamless transition between DoDAF Architectures and Systems Engineering
- Encapsulation of architectures: incorporation of the "internal vs. external" paradigm will help in the generation of OV-3 and SV-6, as well as in creating scalable architectures.

Quick Scorecard: (0-4)

3	Overall Ease of Use
3.5	Coverage of EA Modeling Needs
3	Adherence to Model Standards
3.5	Automation of Model Build/Edit
4	Appearance and Readability of Models
4	Use of Model Data Internally
3	Use of Model Data Externally (interfaces)
3	Reports Included
3	Reporting Writing Capability
3	Stability
3	Use on a Large EA Program
3	Customizability
<b>39</b>	<b>Total</b> (out of 48)

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

### Author Biography

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 50 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 3 papers for the International Command and Control Research and Technology Symposium, and is a regular contributor to a|EA Journal.