2015

Evaluation of the DoDAF Meta-Model's Support of Systems Engineering

Giachetti, Ronald E.

Elsevier B.V.

Giachetti, Ronald E., "Evaluation of the DoDAF meta-model's support of system engineering," Procedia Computer Science v.6161 (2015) pp. 254-260
https://hdl.handle.net/10945/48345

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 61 (2015) 254 – 260

Procedia Computer Science

Complex Adaptive Systems, Publication 5
Cihan H. Dagli, Editor in Chief
Conference Organized by Missouri University of Science and Technology
2015-San Jose, CA

Evaluation of the DoDAF Meta-Model's Support of Systems Engineering

Ronald E. Giachetti[a]*

[a]*Systems Engineering, Naval Postgraduate School, Monterey, CA 93940, USA*

Abstract

The Department of Defense Architectural Framework (DoDAF) is the DoD's mandated method to document system architectures. Consequently, it has a far-reaching and deep impact on systems engineering and acquisition within the DoD. DoDAF version 2.0 departs from earlier versions in its emphasis on the underlying architectural data. It is now based on a meta model called DM2 that defines all the concepts and their relationships. This paper examines DM2 with a critical eye toward how it is defined, modeled, and used to support systems engineering. The evaluation is based on a text analysis of systems engineering guidebooks and a comparison with the DM2 definitions. We classify mismatches between the systems engineering guidebooks and DM2 and highlight some omissions in DM2 of important systems engineering concepts. We find that DM2 has some serious omissions that have implications for the ability of DM2 to support systems engineering. We make two recommendations for how DM2 can address these issues. First, the underlying ontology should be developed and modeled using more widespread used tools to facilitate greater understanding, adoption, and acceptance by the community. Second, the ontology must be designed to support the decision processes using language familiar to the intended users.

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology
Keywords: Ontology; systems engineering; system architecture; model-based systems engineering;

1. Introduction

The U.S. Department of Defense Architecture Framework (DoDAF) describes a comprehensive set of models and vocabulary for developing system architectures to support Department of Defense (DoD) managers in making decisions about the acquisition and deployment of systems. The use of DoDAF is mandated to support six core

* Corresponding author. Tel.: +1-831-656-2670.
E-mail address: regiache@nps.edu

Peer-review under responsibility of scientific committee of Missouri University of Science and Technology
doi:10.1016/j.procs.2015.09.208

processes, which includes systems engineering. DoDAF encourages the use of common principles, terminology, and models so that architecture descriptions can be shared across organizational boundaries and thus enable more extensive deployment of joint operations with the DoD. The role of DoDAF as an architecture framework is to define the many model views, what the model views contain, and what data definitions are used in each model. But the DoD also recognizes the framework will be used by myriad different organizations with different needs, consequently DoDAF leaves sufficient freedom for tailoring models to suit project needs, to use any suitable modeling language, and to use different methodologies. As a result, DoDAF does not specify particular modeling languages to use for any of the views, nor does it specify a single architecture development methodology. The main requirement is that whatever modeling language is used, it shall comply with the DoDAF Meta Model (DM2) which expresses data requirements for each modeling view.

The mandated adoption of DoDAF effectively promotes Model-Based Systems Engineering (MBSE) because models are used to support all systems engineering activities throughout the system life-cycle.[1] Models become the primary means for documenting the system design. Models are expressed in terms of a modeling language that specifies the symbols or primitive elements of the modeling language.[2,3] Each model symbol has a syntax defining how it can be used in the model and associated semantics defining how to interpret the symbol. In systems engineering, there is a need for the semantics of the model to be formalized, which is accomplished with a meta-model. The reason for formal semantics is to avoid problems inherent in the communication of ambiguous representations, to enable computer interpretation of models, and to exchange models between systems engineering teams. The meta-model provides a standardized and consistent terminology resulting in a shared and precise interpretation of a model.

DoDAF versions prior to version 2.0 focused on the output products. Version 2.0 of DoDAF represents a major shift from the product-oriented approach of earlier versions by introducing the new meta-model, DM2, and by being data-centric and emphasizing that it is the data created by the modeling activity that is important. This further aligns DoDAF with MBSE where all model data are stored in a shared model repository. More importantly, the data-centric approach embodied by DoDAF v2.0, means it is the underlying data schema, DM2, that is delivers the value in the architectural products. An evaluation of how well DoDAF v2.0 fulfills its purpose of supporting systems engineering is summed up as first whether DoDAF contains information pertinent to the decision processes, and second how well the models express this information in formats amenable to the decision makers. The first question can be answered by deriving the information needs of systems engineering and examining DM2 vis-a-vis these information needs.

What the above discussion highlights is the importance of the underlying meta-model to the stated purpose of DoDAF as well as to the broader MBSE paradigm. Unfortunately, there is little published in the open literature concerning how well DoDAF and DM2 fulfill their design purposes. This paper conducts a critical analysis of DM2 with a focus on how well it supports systems engineering, one of the six core decision processes. The analysis highlights several problems that we feel can be remedied by adopting commercial ontology de facto standards and tools. The paper explains how this can be done. One caveat about this paper is that DoDAF is a work-in-progress that is constantly being updated. The version as of this article is v2.02.[4]

1.1. History of DM2

The history of DM2 is subordinate to the history of DoDAF, which starts with the establishment of the C4ISR Architecture Framework. The motivation for the C4ISR architectural framework was to comply with federal regulations such as the Clinger-Cohen Act, which was promulgated to reform how federal agencies plan for and acquire information technology. System architecture is seen as a way to comply with the law by aligning information technology with performance goals and the agency's strategic mission. In 2003, DoDAF v1.0 was released, and was largely based on the C4ISR Architecture Framework, which was at version 2.0 in 2003. The goal of DoDAF v1.0 was to provide more documentation for greater dissemination, and also to broaden the use of architecture beyond the C4ISR community. DoDAF v1.0 incorporated the Core Architectural Data Model (CADM). CADM was an entity-relationship model initially defined using IDEF1x notation. DoDAF v1.5 was released in 2007 to transition the DoD toward its vision of network-centric warfare and a more service-oriented architecture.[3] The underlying data model, CADM was also updated, mostly in attempts to simplify the model but also to include net-

centric elements. Since CADM was the predecessor to DM2, we will briefly describe it and how its inadequacies led to the development and adoption of DM2.

CADM had 687 entities and 3914 attributes. Entity-relationship models such as CADM are usually normalized to third normal form for implementation in a relational database.[3] One side effect of normalization is that the number of entities and attributes tends to increase in order to achieve the normalized data structure. The large number of entities and relationships, and the complexity of CADM made it difficult to work with, which led to much criticism. This was dealt with by the release of DoDAF v2.0, which involved the complete abandonment and replacement of CADM with the DoDAF Meta Model (DM2). DM2 is a significant change in approach because whereas CADM was a data model, DM2 is an ontology or knowledge model. It is a fundamentally different approach to modeling the underlying architectural data.[5]

The brief summary of the evolution of DoDAF is provided to highlight two trends. The first trend is that while the initial motivation for developing an architectural framework was to develop system architectures in compliance with federal regulations such as the Clinger-Cohen Act, the motivation has been expanded to support six core DoD processes. We believe this derives partly from a desire to obtain greater value from the investment made in developing and maintaining system architectural descriptions. The second trend, acknowledged in the DoDAF manuals, is the transition from a model-centric view of architecture to a data-centric view of architecture. Since DoDAF does not currently specify particular model types for any of the architectural views, it then does not have any influence over the content of those views in terms of semantics unless it adopts an underlying data model. Moreover, the models are simply views of the overall architecture from a particular perspective.

1.2. DM2

The DM2 supports DoDAF with the specific purposes to establish a domain specific language with formal syntax and semantics. The design intent of DM2 is to enable a common and shared understanding of architecture models due to the precise semantics, which is the basis for semantic interoperability a stated goal of DoDAF. The DM2 is described in terms of three levels. The top level is the Conceptual Data Model (CDM) that defines the main data constructs and the relationships between them so that nontechnical stakeholders can understand the underlying data content. The Logical Data Mode (LDM) is the formal specification of the domain concepts, their semantics, and the relationships between them. The Physical Exchange Specification (PES) is for sharing models. In this review we will focus on the LDM where the architecture occurs.

Underlying the DM2 is the IDEAS ontology. The IDEAS (International Defense Enterprise Architecture Specification) group says it was developed by SMEs knowledgeable about the Australian, Canadian, UK and US defense departments some presentations use more suggestion language saying it was “co-developed by Australian, Canadian, UK, and US defense departments.” The U.S. DoDAF is the first to adopt it and DoDAF 2.0 released in May 2009. A foundation idea of the IDEAS ontology is that only things that have physical extent are modeled.[5] Here physical extent means in space and/or time. The IDEAS ontology does not seem to be used elsewhere and there is little published literature on it. One challenge in using IDEAS is that it did not conform to common design approaches for ontologies. However, in the past year the DM2 working group released an OWL ontology of the DM2, and it is this version that is reviewed here.

2. Other Evaluations of DoDAF

Few external evaluations of DoDAF or DM2 exist. In 2009 the National Defense Industry Association (NDIA) organized a working group consisting of a panel of experts to review, analyze, and make recommendations concerning DoDAF v2.0.[6] They summarize their recommendations as two main types, the first being to extend the DM2 with additional information typically needed by systems engineers and the second being greater standardization of the methods to develop and maintain DoDAF models. The NDIA report recommends standardization in terms of the modeling language, arguing for UML and SySML to be used in conjunction with an object-oriented methodology. DoDAF makes explicit in several parts of the documentation that it is intended to be tool-agnostic and methodology agnostic. While not discussed in the NDIA report, a recommendation for standardization must be in two steps. First, to establish whether standardization on how to model is desirable, and

then second, to promote one approach over all others. The position of DoDAF is to standardize the "what" or the information content of architecture models. Some of the specific criticisms made in this report by the panel of experts are similar to what is arrived at in this paper through the text analysis.

The Navy's Chief Systems Engineer commissioned a study in 2008 to evaluate how well DoDAF can support modeling and simulation activities.[7] The analysis was conducted by using DoDAF to model a realistic case study and identifying gaps that DoDAF could not or did not model well. The use of DoDAF for executable architecture is an important goal to support systems engineering. Other than these evaluations we could not find other evaluations of DoDAF v2.0. Evaluations of earlier versions of DoDAF are mostly not pertinent because the earlier versions of DoDAF were product-centric and the evaluations tended to focus on the products or models developed and not the underlying data elements.

3. A Linguistic Analysis of DM2

Linguistic relativity theory was initially proposed by Sapir and Whorf and says an individual is constrained to describe and think about a phenomenon by their language.[8,9,10] The theory claims that when an individual observes something, they are not guided only by the physical evidence they see, but are guided also due to their linguistic backgrounds, which influences their worldview. The concept is not new, Wittgenstein[11] wrote, "the limits of my language means the limits of my world" where he is expressing the viewpoint that how we perceive the world is bounded by language. This idea also arises in Soft Systems Methodology with the use of *Weltanschuuang* to describe a worldview that influences how a problem is defined and a system solution developed.[12]

The premise of this paper is that linguistic relativity theory applies to architecture frameworks as used by systems engineers. DoDAF, as an architecture framework, includes the DM2 ontology that seeks to define all the terms needed to support the systems engineering life-cycle. In other words, DM2 provides the language for DoDAF-compliant models. The reason linguistic theory applies in this domain is that if a concept is not defined in DM2, then it cannot be represented in DoDAF, and consequently a person is unable to express this concept. Ideally, the DM2 ontology should use the domain language of the system engineer and support all the decisions the system engineer must make throughout the engineering life-cycle.

To identify the concepts that should be included in the ontology, we perform a concordance analysis of systems engineering guidebooks. A concordance ranks words used in a corpus (text document) according to their frequency in the text compared to an expected frequency in another set of texts. Concordance analysis allows analysis of collation, frequency, and whether the frequency of a word is unusually compared to other texts. In this analysis, we posit that if a term appears frequently in systems engineering guides, then this term is important and should be included in an ontology to support systems engineering.

A text analysis software called AntConc was used to mine these documents and analyze the text content.[13] The performance of a text analysis involves four types of text documents: (1) source documents; (2) reference document; (3) lemma list, and (4) not include list. The source documents are those documents that are being analyzed. Table 1 shows the selection of source documents utilized to identify concepts. We selected guidebooks because they document the systems engineering process and are readily available electronically for a text analysis. One alternative we considered, but decided against, was textbooks on systems engineering because while these might be good sources, they are not readily available electronically in a format amenable to text analysis.

The reference document is used for the comparison of word frequency and usage. The way the text analysis works is if a word is more frequently used in the source than in the reference document, then that word is more significant to the source domain. If a word is common in English usage, then that word ranks very low. For example, the word "is" will likely occur with similar frequency in the source documents as well as any reference document so it would not rank high in the text analysis. The word "stakeholder" is used much more commonly in systems engineering than common usage so this would rank high. In order to highlight those words of unique significance to systems engineering we chose a set of reference documents that included a *Yahoo News* article from 2012, a science and technology article from the *Economist*, a *CNN News* article, and an article from *American Scientist*. The lemma list is a list of words that should be treated as a single concept. We combined occurrences of the same word where the difference was singular or plural. For example, the term "stakeholder" and "stakeholders" is counted together. Table 2 shows some of the entries in our lemma list. It was constructed iteratively by running

the text analysis, observing words that corresponded to the same concept and then adding them to the lemma list until there were no repeating entries under similar words. The final list is the list of words not to include in the analysis. This list was also generated iteratively and included words such as "INCOSE, handbook, council, International" all of which occur very frequently in the handbooks but are not relevant to the systems engineering domain we are examining.

Table 1. Sources for Text Analysis

Navy System-of-Systems Guide (2011)
INCOSE Systems Engineering Handbook v3.2 (2010)
NASA Systems Engineering Handbook
Naval Systems Engineering Guide (2004)

Table 2. Partial Lemma List

Activity = Activities
function->functionality, functions, functional
stakeholder->stakeholders, customer
capability->capabilities

The first analysis is a keyword list that identifies the most frequently used words in the documents. Table 3 shows the results of the text analysis. The list is ordered by the term's keyness which is calculated as a log-likelihood measure.

Table 3. Top 20 results from text analysis

Rank	Frequency	Keyness	Keyword
1	6315	227.165	system
2	2866	112.384	requirement
3	3052	80.185	process
4	1843	63.114	project
5	1552	60.858	management
6	1467	48.826	design
7	1223	47.957	risk
8	1087	42.624	function
9	1008	39.527	activity
10	885	34.703	decision
11	759	29.763	stakeholder
12	738	28.939	performance
13	726	28.469	verification
14	695	27.253	interface
15	879	26.793	review
16	672	26.351	program
17	665	26.077	products
18	835	25.17	cost
19	580	22.743	elements
20	545	21.371	validation

The list of key terms from the guidebook was compared to the DM2 in order to evaluate how well the DM2 supports systems engineering. The DM2 and DoDAF is a work that seems to be constantly under development. Consequently, we need to specify what version and the source for our comparison. For our comparison, we are use the DM2 from DoDAF v2.02 and as defined in the DM2 Baseline file `DM2.02_OWL_Full.xml` that was available from the DoDAF website. It is possible that some terms are missing from this file and thus are omitted from our analysis, but we need to select a source for comparison and the alternative is the website. We were hesitant to use the website because websites change frequently and then with updates you lose the previous versions.

In our analysis we identified three types of mismatches between the text analysis of the systems engineering guidebooks and DoDAF. First, there is the mismatch where the term used by DoDAF is different from the term generally used in systems engineering for the same concept. Second, there is a mismatch in the aggregation or boundary between how a concept is defined in DoDAF compared to the systems engineering guides. Third, the DoDAF ontology is missing a concept that is important as defined as being used with high frequency in the systems engineering guidebooks. To illustrate some of the evaluation, Table 4 shows some terms from the systems engineering guidebooks, any corresponding DM2 term, and a brief summary analysis.

Table 4. Some illustrative evaluations

Rank	Keyword	DM2 Term	Summary Analysis
1	System	Performer	A subtype of performer. A mismatch in the aggregation level.
2	requirement	-	Not included
7	risk	-	Not included
8	Function	Activity	
11	Stakeholder	-	Not included. The DM2 Data Dictionary has a term Actor that was considered as a candidate term in DM2 but not included. A stakeholder cannot be a Performer because a Performer performs an activity and provides a capability, which stakeholders do not do.
14	Interface	Port	A Port is a sub-type of Performer. Definition of Port does not correspond well to common SE usage of Interface.
29	Capability	Capability	

First, we note that many DoDAF DM2 terms are not found at all in the systems engineering source documents. For example, the term “Performer” in DoDAF is not found in any other source document. It is a term originating in DoDAF and now slowly making its way into other systems engineering documentation published by the U.S. Department of Defense. A Performer as defined by DM2 is, “Any entity - human, automated, or any aggregation of human and/or automated - that performs an activity and provides a capability.” The subtypes of Performer are PersonRoleType, OrganizationType, System, Service, and Port. DM2 defines Performer based on what it does in a system rather than what it is. The difference between the types of performers are important to system engineers, and a possible issue is the lack of distinguishing between the types in DM2. For example, requirements on a system are different than requirements on a person.

DM2 defines Activity as work that transforms inputs into outputs or changes their state. The DM2 activity represents all types of activities such as operational activities, functions, process, and tasks. The use of the single term can be problematic because of how systems engineering in the defense industry distinguishes between operational activities and functions. An operational activity is part of the operational architecture and describes an enduring activity to accomplish a mission in the terms of the users. A function describes what a system does to implement the operational activity. Functions are part of the system architecture in DoDAF. Since the system engineering users of DoDAF make a distinction between operational activities and functions, then the underlying DM2 should also make this distinction.

The problem of common SE concepts such as requirements or risk not appearing in DM2 is more of a concern

than the differences in aggregation. One argument against including requirement is that a model defines the requirements. What this means is that by creating a model such as the OV-6c, we are defining the system requirements. The problem with this argument is that not all requirements can be represented in models in this way. For example, a review of the modeling done by [7] who uses DoDAF to model requirements shows that they are all functional requirements – no performance requirements are captured in the models. This is probably why SysML includes a model for system requirements. We believe this omission is a significant shortfall for using DoDAF to support systems engineering, and that it will cause system engineers to use work-arounds or to misuse DoDAF in order to represent system requirements. A similar argument can be made for risk, which is also missing from DM2.

4. Conclusion

The purpose of this research was to evaluate how well DoDAF supports systems engineering. Our premise, derived from linguistic relativity theory, is that if certain concepts are important to systems engineering, then these terms should also appear in DoDAF. We assume that the concepts important to systems engineering appear with high regularity in the four guidebooks that were used as the source material for a text analysis. In comparing DoDAF against the results of the text analysis of these systems engineering guidebooks we find three types of mismatches: differing names for the same concepts, differing aggregations or boundaries of a concept, and missing concepts. We argue the most serious of the mismatches is the missing concepts in DoDAF because if the concept is not in the ontology, then there is no means to represent it and consequently no way to include it in any systems engineering activities that use that concept. The text analysis identified mismatches between the systems engineering guides, our representation of the knowledge domain, and the DM2. The mismatches were classified according to three types, of which the missing match is the most serious.

Currently, it is reported that the majority of system architectures modeled using DoDAF are done using tools such as Microsoft Powerpoint, Word, and Excel. From this we can safely infer that in most cases, the DoDAF models were created as documentation after all the systems engineering was performed and not as the medium used in a model-based systems engineering process. Since the DoDAF models are not being used for systems engineering analysis and design activities, then many of the short-falls we highlight have not emerged. It is not until the DoDAF models are used as formal models in some MSBE process that these problems will emerge. This helps explain why knowledge about the issues raised in this paper may not be more widespread.

In this research we only considered the concepts that should be included in an ontology. An ontology also includes relationships between concepts, and we are working to extend the analysis to identify relationships by performing a cluster analysis that determines when two or more concepts occur adjacently in a text. Overall, we believe that the shift of focus in DoDAF v2.0 to a data-centric model based on an ontology defined by DM2 is a good approach. What we find fault with is that the ontology does not do a good job of support systems engineering.

References

1. Piaszcyk, C. Model based systems engineering with Department of Defense architectural framework, Syst Eng, 2011, 14: 305-326.
2. Friedenthal, S., Moore, A., and Steiner, R., A practical guide to SysML, Morgan Kaufmann, 2008.
3. Giachetti, R.E., Design of Enterprise Systems: Theory, Architecture, and Methods, CRC Press, Boca Raton, FL 2010.
4. DoDAF V2.02, Volume II: Architectural Data and Models, Department of Defense, January 31, 2015.
5. Bailey, I. and Hozhabrafkan, F., IDEAS group model for interoperability, presentation at CISA Conference, 2006, Williamsburg, VA.
6. NDIA AFWG Report, DoDAF Satisfaction of Systems Engineering Needs, November 2009.
7. ASN CHSENG Metadata Interoperability Study Phase 0 Gap Analysis, SPAWAR, December 2008.
8. Whorf, B.L. Language, Thought, and Reality. Selected Writings of Benjamin Lee Whorf. Cambridge: MIT Press, 1956.
9. Lucy, J.A., Language Diversity and Thought: A Reformulation of the Linguistic Relativity Hypothesis. Cambridge: Cambridge University Press, 1992.
10. Gumperz, J. J., and Levinson, S.C. (Eds.). Rethinking Linguistic Relativity. Cambridge: Cambridge University Press, 1996.
11. Wittgenstein, L. Tractatus logico-philosophicus. London: Routledge and Kegan Paul, 1922.
12. Checkland, P., Systems Thinking, Systems Practice, John Wiley & Sons, Chichester UK, 1981.
13. Anthony, L. AntConc [Computer Software]. Tokyo, Japan: Waseda University. Available from http://www.antlab.sci.waseda.ac.jp/