Winter 2013

Orchestrating Systems Engineering Processes and System Architectures within DoD: A Discussion of the Potentials of DoDAF

Taylor Hughes

Andreas Tolk

Orchestrating Systems Engineering Processes and System Architectures within DoD: A Discussion of the Potentials of DoDAF

Mr. Taylor Hughes, Mr. Andreas Tolk, Ph.D.

Introduction

The Department of Defense (DoD) is responsible for the development, procurement, introduction, integration, maintenance, upgrading, and retirement of all defense and related support systems. To facilitate this task, enterprise-level system architectures are used to describe capability, operational capability requirements and lower level system architectures are used to describe the functionality of system solutions which are necessary to satisfy those operational capability requirements. To ensure the attainment of the required capability solution on time and under budget, system-engineering processes are mandated to ensure consistency and rigor across all participating organizations. However, even though architecting is necessary to insure that all understand user requirements in the same way (providing logical rigor, structure, semantic and syntax), the practice of architecture is sometimes set aside as being unimportant, and programs sometimes fail for lack of good requirements. Defense engineering leadership understands this problem, and the DoD now has the challenge to determine how best to harmonize enterprise/systems architecting practices with systems engineering practices. [1]

The International Council on Systems Engineering (INCOSE) defines systems engineering as an engineering discipline whose responsibility is creating and executing an interdisciplinary process to ensure that the customer and stakeholder's needs are satisfied in a high quality, trustworthy, cost efficient and schedule compliant manner throughout a system's entire life cycle. It is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, performance, test, manufacturing, cost and schedule, training and support, and disposal. Systems engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user's needs. [2]

It is generally accepted that system architectures are used to add structure, logic, semantics and syntax to stakeholders' need-concepts so that their needs may be captured as formal requirements and understood in the same way by multiple stakeholders and developers for the sake of increasing performance and reducing risks and costs. They provide a common knowledge repository that allow all team members to store and evaluate their special facets of the overall challenge in the context of all other system contributions. Chigani and Balci observe that the process of architecting takes the problem specification and requirements specification as input and produces an architecture specification as an output work product. [3] It seems to be immediately obvious as a good practice that the system engineering process must drive the activities that contribute to the system architecture.

The DoD Systems Engineering (SE) Process is defined in the Defense Acquisition Guidebook Chapter 4. [4] The DoD Architecture Framework (DoDAF) [5] defines how to model system architectures within the DoD. The Deputy Assistant Secretary of Defense for Systems Engineering (DASD-SE) already mandates that architecture products be included in every Systems Engineering Plan (SEP) to include how architecture products will be related to requirements definition. [6] In this paper, the authors evaluate opportunities and the potential for establishing and extending DoDAF as the common architecture framework in support of a coherent systems engineering process to align data and harmonize processes of the different technical team members of all stakeholders and over all phases of the system life cycle embedded into the DoD Enterprise.

Architectures as Knowledge Repositories

The people—processes—tools framework is well known in industry. In order to fix or improve something, the right people are needed. These are the systems engineers supporting DoD with their knowledge and expertise in a multitude of domains. To facilitate their collaboration, common processes are needed. Within the DoD, the DoD SE process fulfills these requirements. Finally, the right tools to support the processes are needed, and DoDAF has been designed to meet this need. It is a good practice to look at the systems engineering process and the system architecture process as mutually-supporting activities that are harmonized for the benefit of the enterprise. In practice, however, the authors have identified several potential reasons for the observed insufficient use and alignment of DoD SE and DoDAF in industry: [7]

1. Engineers are placed in charge of projects who do not have a formal understanding of DoDAF practices and their value for management, governance, and administration.

2. Architecting for requirements is considered as only necessary for developing software but not hardware, and definitely not for hybrid systems.
3. The need to integrate software with hardware in increasingly complex ways has outpaced the willingness or ability of systems engineers to adopt or adapt architecting practices to traditional systems engineering practices.
4. In spite of the legal and regulatory requirement to architect requirements before system solutions move forward through various acquisition phases, engineering leaders sometimes commit to acquisitions without architecture for the sake of saving time and resources or for political reasons.

Most of these challenges can be addressed by education, as they point toward people challenges, not method—i.e., the DoD SE process—or tools—i.e., the DoDAF. However, if we do not apply the system architectures as intended, system architects and systems engineers are in danger of working in a 'vacuum,' side-by-side without really utilizing the mutual benefits of orchestrated collaboration as described in the introduction. To this end, the DoD SE process must guide the processes of collaboration, and the DoDAF artifacts must capture the views and constraints of all participating team members.

In other words, architectures must become the knowledge repository for the team, as proposed in the MIT-based doctoral work of Kim. [8] The enterprise architecture provides the context for the system architectures as well as for any portfolios. However, every phase of the DoD SE process and every view of each team member in each life cycle phase must have its data captured in the form of an individual view, following a common standard, in order to enable such collaboration. For the DoD, the question arises: is this possible with the current state-of-the-art DoDAF artifacts?

The DoD Systems Engineering Process

The introduction course to Systems Engineering (SYS101) at the Defense Acquisition University (DAU) starts with the story of two stone cutters that are working side by side and are asked, "What are you doing?" The first one answers: "I am cutting this stone into blocks." The second one explains: "I am on a team that is building a Monument!" [9]

This story is given as an example to understand the context for all required activities and to communicate a vision for the final product. Only with the big picture in mind can the effects of changes within the actual work being conducted become perceivable for all team members. The DoD SE process has been established to ensure that the right work is done, and that the work is done right! This is done with a set of technical work processes orchestrated by a set of technical management processes.

The technical work processes, sometimes differentiated into technical processes for designing systems and technical processes for realizing system products, are:

- Requirements Development
- Logical Analysis
- Design Solution
- Implementation
- Integration
- Verification
- Validation
- Transition

The supporting and guiding technical management processes are:

- Technical Planning
- Requirements Management
- Interface Management
- Risk Management
- Configuration Management
- Technical Data Management
- Technical Assessment
- Decision Analysis

As discussed by Buede [10] in more detail, understanding the requirements is pivotal, and all activities must be driven by requirements. Requirements specify the users' view on the system, what they want to accomplish, what gaps need to be closed, with whom collaboration is needed to conduct a successful operation and with whom resources will have to be shared, etc. A system is only successful if it meets all requirements and a system architecture enables all team members to contribute to this solution efficiently. The questions that need to be answered now are *"Is DoDAF designed to support all phases of the DoD SE process effectively?"* and *"How well does DoDAF support tracking of requirements?"*

The DoD Architecture Framework

The DoDAF evolved over the last decade into a solid method and tool. The current version is DoDAF 2.02 [11]. Earlier versions were driven by views defining the facets needed by several "privileged" team members. DoDAF originally incorporated a data model able to store all the data needed to support these views, the Core Architecture Data Model (CADM). With the introduction of DoDAF 2.0, the underlying paradigm changed to be data-driven instead of view based. DoDAF 2.0 focuses on an extensible data model that captures all data required by any team member in any life cycle phase in a consistent way. With DoDAF 2.0, the DoDAF Metamodel (DM2) defines conceptual categories for all these data elements needed to describe system architectures. The

viewpoints are generated by applying models to the data. i.e., the data model is not generated by the views, as it was the case in the earlier version, but the data can now drive the models to produce views. If new views within viewpoints are needed, they can be generated from the data. If additional data is needed, the data model can be extended within the constraints of the DM2.

Some key conceptual categories of the DM2 are captured in Figure 1.

models generating these viewpoints and eventually by extending the data model as well.

However, although requirements are recognized in the DoD SE process to be pivotal they do not show up in DoDAF, neither as a view in earlier versions nor as a viewpoint or even as a concept within DM2. Does this mean DoDAF does not model requirements? To be fair, let us now review how the various DoDAF viewpoints are expressly intended to affect requirements. [12]

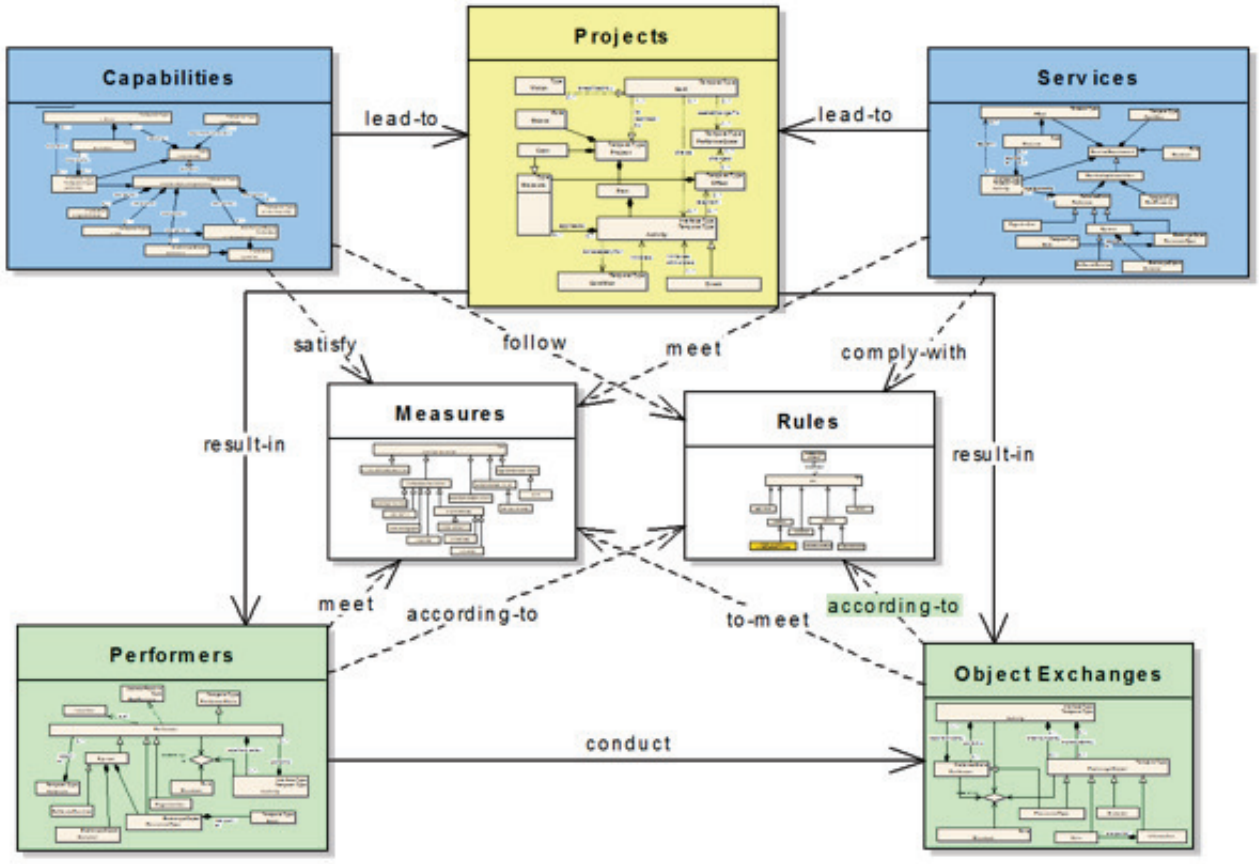

FIGURE 1: HIGH LEVEL CONCEPTUAL CATEGORIES OF THE DM2

Projects hold all activities that belong to a system or a portfolio pursuing the same set of goals and objectives together. They bring in particular required *Capabilities* and available services that provide the functionality needed to expose these capabilities together. To do so, *Performers* exposing the services and resource exchanges needed to orchestrate the participating performers and provide the necessary information are connected under observation of all *Rules*. Finally, measures comprise the *Measures* of merit needed for performance evaluations.

Figure 2 (following page) shows the set of viewpoints provided by DoDAF to support the information needs of team members required in all DoD related projects. These viewpoints are also a courtesy for the users of earlier DoDAF versions to facilitate their work with the new version. It also allows for easier migration of earlier system architectures. As stated before, more views within viewpoints can be generated by each group by introducing new

The Capability Viewpoint

This viewpoint *articulates the capability requirements, the delivery timing, and the deployed capability*. There are seven different "views" or models within this viewpoint, each with a different focus. The following views call out intent to support requirements in some way:

CV-2: Capability Taxonomy. The CV-2 is intended to identify capability requirements, codify required capability elements, and to be a source for the derivation of cohesive sets of user requirements.

CV-6: Capability to Operational Activities Mapping. The CV-6 is intended to be used to trace capability requirements to operational activities.

CV-7: Capability to Services Mapping. The CV-7 is intended to be used in tracing capability requirements to services.

We can therefore conclude that Capability Viewpoint is only

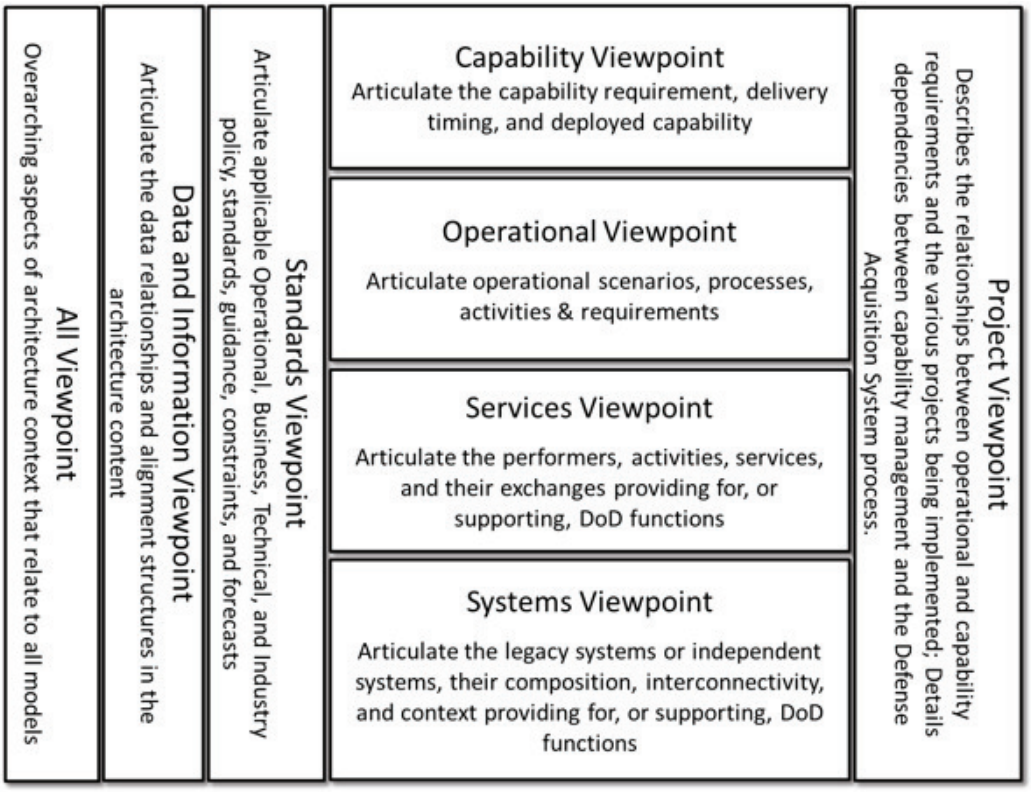

FIGURE 1: HIGH LEVEL CONCEPTUAL CATEGORIES OF THE DM2

expressly used to inform capability requirements. But what is a *capability requirement*? This is not defined in the DoDAF version 2.02. [13]

The Data and Information Viewpoint

This viewpoint *articulates the data relationships and alignment structures in the architecture content for the capability and operational requirements, system engineering processes, and systems and services*. There are three different views (aka, models) within this viewpoint, each with a different perspective. The following views call out intent to support requirements in some way:

DIV-1: Conceptual Data Model. The DIV-1 is intended to include information requirements.

DIV-2: Logical Data Model. The DIV-2 is not expressed as having a purpose of informing requirements. In that the DIV-2 reflects the theory captured in the DIV-1, there appears to be a gap in expression. It should be expressed as having a value-added purpose of adding logic, syntax and semantics for requirements.

DIV-3, Physical Data Model. Here, one will find for the first time that the DIV-2 is actually intended to help requirements. The DIV-3 is defined as an implementation-oriented model that is used in the Systems Viewpoint and Services Viewpoint to describe how the information requirements represented in DIV-2 Logical Data Model are actually implemented. So, we are led to think here that the DIV-2 contains information requirements and the DIV-3 exists to serve the DIV-2?

We can therefore conclude that Data and Information Viewpoint is only expressly used to inform information requirements. But what is an *information requirement*? This is not defined in the DoDAF version 2.02.

The Operational Viewpoint

This viewpoint *includes the operational scenarios, activities, and requirements that support capabilities*. There are 9 different views (aka, models) within this viewpoint, each with a different perspective. The following views call out intent to support requirements in some way:

OV-2: Operational Resource Flow Description. The OV-2 is intended for the elaboration of capability requirements and for the definition of collaboration needs. Ambiguity enters in here with the need to understand the difference between "capability requirements" and "collaboration needs."

OV-3: Operational Resource Flow Matrix. The OV-3 is intended to be used for the definition of interoperability requirements.

OV-5a/OV-5b: Operational Activity Decomposition Tree/ Operational Activity Model. The OV-5, according to DoDAF 2.02, is intended to be used for requirements capture. What kind of requirements? At what level?

OV-6c: Event Trace Description. The OV-6c is intended to be used for the identification of non-functional user requirements.

We can therefore conclude that Operational Viewpoint is only expressly used to inform capability requirements (elaboration),

interoperability requirements, and non-functional user requirements. But these are not defined in the DoDAF version 2.02.

The System Viewpoint

This viewpoint has an ambiguous definition within DoDAF 2.02: *for Legacy support, is the design for solutions articulating the systems, their composition, interconnectivity, and context providing for or supporting operational and capability functions.* Is the Systems View only "for Legacy support?" There are 13 different views (aka, models) within this viewpoint, each with a different perspective. The following views call out intent to support requirements in some way:

SV-1: Systems Interface Description. The SV-1 is intended to be used to capture System Resource Flow requirements. Why not System of System or system interface requirements?

SV-2: Systems Resource Flow Description. The SV-2 is intended to be used as a Resource Flow specification. In general, the term specification implies technical requirements, but that is not clear here. DoDAF does not define specification as it is intended to be understood.

SV-4: Systems Functionality Description. The SV-4 is intended to be used for identification of functional system requirements.

SV-5a: Operational Activity to Systems Function Traceability Matrix. The SV-5a is intended to be used for tracing functional system requirements to user requirements and for tracing solution options to requirements. What is a "user" requirement in this context?

SV-5b: Operational Activity to Systems Traceability Matrix. The SV-5b is intended to be used in tracing system requirements to user requirements and tracing solution options to requirements.

SV-7: Systems Measures Matrix. The SV-7 is intended to be used in the identification of non-functional requirements.

SV-10c: Systems Event-Trace Description. The SV-10c is intended to be used in the identification of non-functional system requirements.

We can therefore conclude that System Viewpoint is only expressly used to inform system resource flow requirements, functional system requirements, system requirements, user requirements, non-functional requirements, and non-functional system requirements. DoDAF version 2.02 does not define these requirements expressions. How does one align these expressions to those needs in the DoD Systems Engineering Process?

The Services Viewpoint

This viewpoint is defined as *the design for solutions articulating the Performers, Activities, Services, and their Exchanges, providing for or supporting operational and capability functions.* There are 13 different views (aka, models) within this viewpoint, each with a different perspective. The views within this viewpoint call out intent to support requirements in a way similar to those within the Systems Viewpoint.

In summary, requirements are captured implicitly and as such already are providing the various team members with guidance, but improvements are possible.

Recommended Improvements

Within the course of the underlying research, the authors searched all uses of the term requirements in the current DoDAF version. This quick review reveals several areas of concern:

- There are over 50 different ways of describing a requirement, with several different expressions referring to the same requirements concept.
- There is neither a DoD requirements taxonomy nor a DoD requirements ontology given.
- There is no obvious alignment of the DoDAF viewpoints to the DoD SE process. Even directly related DoD requirements specification documents and requirements are not mentioned. [14]

Given these observations, and given that enterprise and system architects within DoD are consistently required to utilize DoDAF as their mandated framework for developing architecture artifacts for DoD needs, one can imagine that there can be very serious ambiguity on the part of some architects regarding why they are developing their architectures. In other words: the requirement development phase and requirement management phase of the DoD SE process have to be unambiguously supported by DoDAF artifacts. In order to capture a requirement effectively and make sure it can be validated, one must communicate unambiguously what needs to be observed and measured and what values are within tolerance. In other words, a requirement that cannot be observed and measured at the end of the day is useless for the engineer. This leads to the improvement that requirements shall be traceable to DoDAF artifacts and shall be accompanied by a set of metrics applicable to decide if an implementing system fulfils this requirement within the boundaries of a tolerance interval.

The Systems Modeling Language (SysML) is a graphical modeling language adopted by the Object Management Group (OMG) in 2006. [15] It was developed in response to the huge accomplishments of the Unified Modeling Language (UML) in the software engineering domain with the objective to derive a language supporting system modeling equally successful to that of software modeling. OMG collaborated to this end with the International Council on Systems Engineering (INCOSE) and the European Systems Engineering Group (AP233) to orchestrate a consortium with members from government, industry, and academia. In recognition of the essential importance of requirements, they introduced two diagrams to explicitly capture the ideas described above:

- The *Requirements Diagram* represents visual modeling of requirements for the system, which are pivotal for systems engineering;
- The *Parametrics Diagram* captures the relations between parameters for system components at all levels and provides the metrics to be used for system performance evaluation.

While the requirements diagram supports the description of requirements in the language of the user, the associated parametrics diagram captures in detail which system attributes have to be measured and in which tolerance intervals their value may be under different circumstances. To do this, SysML uses a mathematical description language that can be directly translated into test plans. Another advantage of SysML was pointed out in the doctoral thesis of Shuman. [16] He evaluated several approaches in support of executable architectures and was able to show that architectures specified in SysML are part of the group that supported the idea to execute an architecture based on the produced artifacts best.

As DoDAF is data-driven, there is no reason not to include parametrics under the conceptual category of measures. Requirements fit well under the rule category, if the community doesn't want to give them their own category in upcoming versions.

Conclusion

As pointed out by the authors before, DoDAF Viewpoints should be formally aligned to the DoD SE Process products/requirements and their associated documents which they can directly influence. A clearer guideline on how to accomplish these objectives cannot be left to selected academic organizations, but needs to be integrated in the form of examples into future DoD guidelines. The community of practice must actively participate in discussions to address the concerns formulated in the beginning of this contribution. Harmonizing architecture efforts and DoDAF with the engineering practices captured in the DoD SE processes is pivotal to better support managing the increasing complexity in projects and portfolio efforts of today's engineering challenges.

DoDAF has the technical potential to support all phases captured in the DoD SE process. The flexibility of DM2 and models to drive viewpoints makes it a good tool and method to support the process in the best way. What is now needed is the commitment to educate the people to perform optimally in the triangle of people-process-tools for the DoD: Engineers and other team members following the DoD SE process guiding the collaboration utilizing the DoDAF with respective extensions to support them in all phases of the life cycle. This contribution has the objective to show that this is not a technical challenge, as DoD SE process and DoDAF already provide the foundation for the required capabilities.

References & Notes

1. Assistant Secretary of Defense for Networks and Information Integration (ASD(NII)) DoDAF Team Briefing: DoDAF Maintenance Update, Version 2.03, 5 January 2012.
2. International Council on Systems Engineering (INCOSE): Systems Engineering Handbook Ver. 3.2, October 2011, online: http://www.incose.org/ProductsPubs/products/ sehandbook.aspx
3. A. Chigani, and O. Balci (2012) "The Process of Architecting for Software/System Engineering," International Journal for System of Systems Engineering 3 (1):1-23
4. Defense Acquisition University (DAU), Defense Acquisition Guidebook, Chapter 4.3, "Systems Engineering Processes" online: https://acc.dau.mil/CommunityBrowser.aspx?id=638325
5. DoD Chief Information Officer (CIO), DoD Architecture Framework (DoDAF) Version 2.02, August 2012
6. Systems Engineering Plan (SEP) Outline (DUSD approved), Version 1.0, 20 April 2011
7. T. Hughes, and A. Tolk (2013) "Harmonizing Architecture Efforts With Engineering Practices in DoD: A Brief Discussion," The Newsletter of Reliability, Maintainability, & Supportability 17(3):1-3
8. G.G. Kim (2006) "An analysis of structure and process of corporate alliance development using system architecture frameworks." PhD thesis, Massachusetts Institute of Technology.
9. Learning Material Handout "SYS101: Introduction to Systems Engineering," Module 1, Defense Acquisition University, Washington, DC
10. D.M. Buede (2009) The Engineering Design of Systems: Models and Methods, 2nd Edition, Wiley
11. DoD Chief Information Officer (CIO), DoD Architecture Framework (DoDAF) Version 2.02, August 2012
12. Note that we state nothing about constraints here. While we are aware there are constraints that will influence requirements, diving into a discussion on constraints is beyond the scope of this paper.
13. Capability Requirement is defined in CJCSI 3170.01H, Joint Capabilities Integration Development System (JCIDS).
14. Deputy Assistant Secretary of Defense for Systems Engineering (DASD-SE) mandates that architecture products (i.e., DoDAF) be included in every Systems Engineering Plan (SEP) to include how architecture products will be related to requirements definition. See Section 2, DUSD-approved Systems Engineering Plan (SEP) Outline, Version

1.0, 20 April 2011 for details.

15. The current version of the standard as well as tutorials can be found at http://www.omgsysml.org/

16. Edwin A. Shuman IV (2011), "Understanding the Elements of Executable Architectures through a Multi-Dimensional Analysis Framework," Ph.D. Dissertation, Batten College of Engineering, Old Dominion University, Norfolk, VA

About the Authors

Andreas Tolk is Chief Scientist for SimIS Inc. and adjunct Professor for Engineering Management and Systems Engineering at Old Dominion University in Norfolk, VA. He holds a PhD and M.S. in Computer Science from the University of the Federal Armed Forces, Munich, Germany. He received the Technical Merit Award from the Simulation Interoperability Standards Organization (2010) and the Outstanding Professional Contributions Award from the Society for Modeling and Simulation (2012). Dr. Tolk has edited four published books and has contributed to more than 250 peer-reviewed publications in the categories of book chapters, journal articles, and conferences papers.

Mr. Taylor Hughes is a FEAC Certified Enterprise Architect (CEA) serving as a project manager supporting the United States Marine Corps in the development of enterprise and solution-level architectures and requirements. He is a United States Army veteran who has also served in several enterprise / systems development and operational leadership, management and engineering positions for small and major defense programs throughout his 25year career. He is an employee of SimIS, Inc. See www.simisinc.com.