Improving the Practice of DoD Architecting with the Architecture Specification Model

Huei Wan Ang, Dave Nicholson, and Brad Mercer
The MITRE Corporation

Abstract

As the Department of Defense (DoD) moves towards the development of very large systems-of-systems intended to enhance and increase net-centricity (e.g. FORCEnet, LandWarNet, ConstellationNet) our ability to achieve the stated goals of these systems may be limited by our ability to architect them. Coincident with the emergence of such large efforts has been the realization of significant lessons learned during the first generation of DoD architecting practice.

Large-scale systems architecting within DoD and throughout the federal government is a fundamental practice. The DoD Architecture Framework[1] (DoDAF) Version 1.0 serves to guide the DoD's architecture practice. As we come to the end of the first generation of such practice, we find that it is in jeopardy of failing to achieve future goals because of a lack of formal conceptual underpinnings. This paper proposes that the limiting factors are fundamental and result from inadequacies in the semantic foundations for describing such architectures. We describe an Architecture Specification Model that addresses these concerns and significantly improves the ability to use architecture to provide actionable information in support of the DoD's core processes.

Keywords: architecture, architecture semantics, architecture data model, capability-based planning, DoDAF, enterprise architecture, portfolio management

1. Introduction

The first generation of DoD architecting has been characterized by a general failure to meet the expectations of its clients. Although DoD "architects" have likely produced thousands of architecture descriptions, they generally have not produced actionable decision information in support of core organizational processes; yet, the purpose of DoD architecting has always been to support the development of such actionable information. Core organizational processes include:

- capabilities-based planning processes that provide useful architecture descriptions of ends, ways, and means expressed as the full range of Doctrine, Organization, Training, Material, Leadership, Personnel and Facilities (DOTMLPF) architecture alternatives.
- systems acquisition and portfolio planning/investment processes supported by unambiguous ways to compare architecture alternatives.
- systems engineering processes driven by architecture information for requirements development and analysis.

During this first generation of practice variations in definitions, methods, and results constrained DoD architecting to being a *cottage-industry*. Lessons learned have shown that the DoDAF, which was intended as the foundation for DoD architecting, could not be applied in any repeatable way without first "interpreting some formalism into it." Such interpretation has usually been dependent upon a few key local practitioners who could "explain" away DoDAF's gaps and inconsistencies. The resulting differences in interpretation and definition of DoDAF concepts have lead to *pockets of good practice* that cannot explicitly interoperate.

On the other hand, architecting efforts attempted by less sophisticated practitioners, who are in the majority, have remained confused and unsure of how to apply DoDAF. As a result they have practiced "check the box" and "PowerPoint" architecture.

[1] DoD Architecture Framework (DoDAF) was finalized in three volumes (Introduction, Product Descriptions, and Deskbook) on 9 February 2004. The DoDAF emerged when is was determined that its predecessor the C4ISR Framework, vers 2.0, dated 18 December 1997 (vers 1.0 was dated 7 June1996), needed revision to serve as the foundation for a broader practice of DoD architecting focused on DOTMLPF solutions.

This paper takes the position that these aspects of the first generation of DoD architecting characterize it as a period of *immature practice* and that the achievements needed from the next generation require the maturation of such practice.

2. Maturing the Practice of DoD Architecting

Figure 1 illustrates a simple model of mature practice as the application of reliable knowledge through mature processes. This model characterizes mature processes as defined, repeatable, and measurable; and implies that it is only through mature practice can expected results be achieved. Both knowledge and process must be developed in order to achieve mature practice.

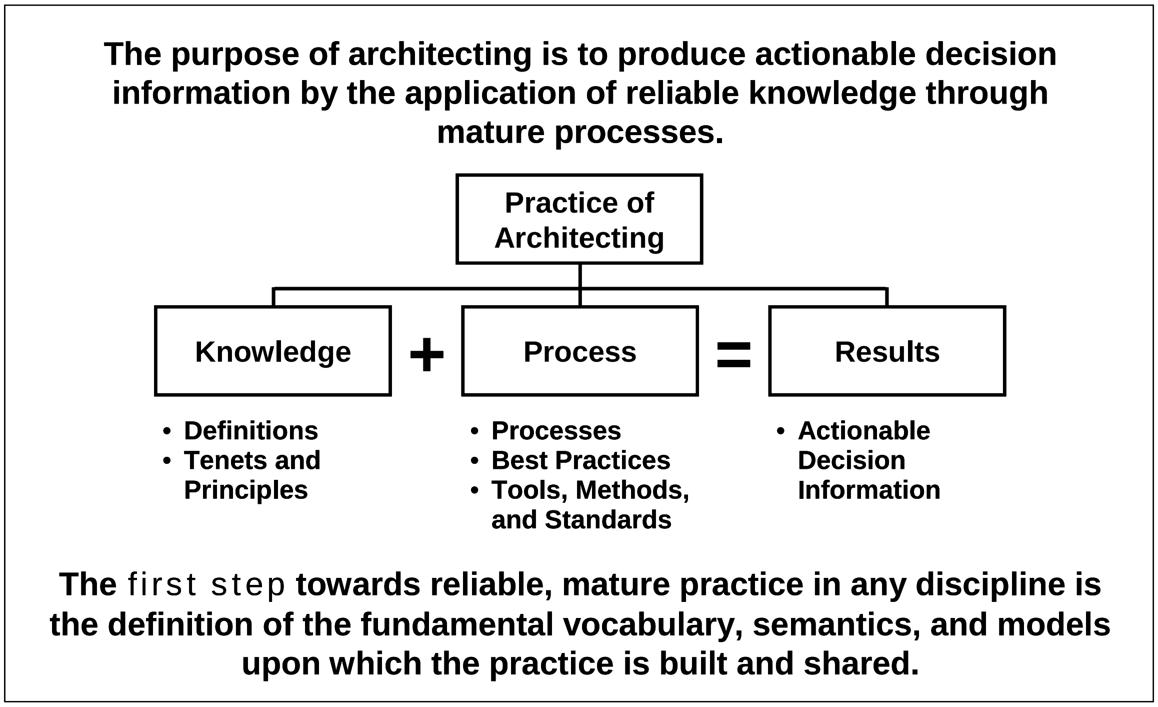
The purpose of architecting is to produce actionable decision information by the application of reliable knowledge through mature processes.
Practice of Architecting
Knowledge
+
Process
=
Results
• Definitions
• Tenets and Principles
• Processes
• Best Practices
• Tools, Methods, and Standards
• Actionable Decision Information
The first step towards reliable, mature practice in any discipline is the definition of the fundamental vocabulary, semantics, and models upon which the practice is built and shared.

Figure 1. Maturing the practice of DoD architecting will lead to its next generation.

Our ability to achieve future goals in describing and evaluating very large system-of-systems architectures is limited both in knowledge and process. First, the body of knowledge underpinning DoD architecting lacks a sufficiently formal and complete vocabulary capable of expressing the increasing conceptual complexity of such systems. Second, DoD architecting practice lacks mature processes capable of integrating multiple individual architecture descriptions and evaluating the architecture of the integrated whole.

This paper proposes that the primary limiting factor in very large system architecture development fundamentally results from inadequacies in the semantic foundations of architecture description—a knowledge deficiency. Because the first step toward reliable, mature practice in any discipline is the definition of the fundamental vocabulary, semantics, and models upon which the practice is built and shared, we believe that the first step is to correct this deficiency.

3. Introducing the Architecture Specification Model

The development of the DoD Architecture Framework and the earlier C4ISR Architecture Framework can be viewed as first generation attempts at creating a descriptive vocabulary for expressing architecture concepts and for creating structures for collecting and organizing data describing specific architectures. These first generation frameworks were developed through consensus activity that emphasized practical compromise over conceptual integrity. As a result, they lack sufficient conceptual foundation; they express complex superstructures without clear conceptual underpinnings.

The ad hoc methods employed in developing these frameworks imply that there does not yet exist a semantically complete[2] conceptual language for describing DoD architectures. The resulting variability between architecture descriptions produced by different pockets of good practice that have each "interpreted the needed formalism" means that their interoperability is very low and their affinity for integration is poor. Architecture descriptions developed by different teams of practitioners in almost all cases cannot be federated, compared, analyzed, or assessed without extensive manual normalization to a temporary and assumed standard representation. This limitation may likely become the most constraining factor in net-centric systems-of-systems architecture development.

The Architecture Specification Model (ASM) has evolved as the result of a U.S. Air Force objective to manage the risk detailed above and generally inherent in describing DoD architectures based on the DoD Architecture Framework. It emerged from the results of an earlier effort, the Activity Based Methodology (ABM)[3], which created processes for DoD architecture description.

The ABM was developed to establish a common means to express integrated architecture information consistent with the intent of DoDAF. As described later in this paper, both the ASM and the ABM take a data-centric approach for architecture element and product generation rather than the product-centric approach inherent in the current DODAF. The ABM was developed specifically to provide a tool independent method for architecture description that focused on the flow of "activities" within the architecture.

While the ABM effort focused on the creation of architecture description process, development of the ASM was focused on providing a tool and methodology independent semantically complete model of the concepts used in understanding and describing DoD architectures. The goal in developing the ASM was to alleviate architecting knowledge deficiencies uncovered by the ABM effort.

Specific objectives of the ASM effort have been to provide a small yet powerful set of DoD architecture description concepts that would provide a semantically complete shared vocabulary for the DoD architecting community; that could serve as a formal foundation for development of a next-generation architecture framework; that would support interoperability between DoD architecting practices; and that also would support architecture-based analysis in support of DoD core processes.

4. Data-Centricity versus Product-Centricity

It is important to note that the ASM differs from other similar efforts and current framework development in that it does not begin with, nor is it constrained by, any existing framework or set of architecture description products. Architecture description products are a means, graphical, textual, or tabular, for capturing and presenting a defined set of architecture description elements and their relationships in a visually consistent way. An architecture description product is often focused on one of the six interrogatives. It has been the objective of this effort to begin with the descriptive concepts employed in DoD architecting and then to develop a holistic model of the same. By beginning with a holistic model any submodel composed of the set of architecture description concepts required by any desired view[4] or product may be composed. In addition, the concern on visual representations (e.g., notations) of descriptive concepts in products is separated from the concern on their definitions and can be deferred to architect or stakeholder's choice. Figure 2 illustrates the fundamental pattern composing the pattern composing the ASM.

The views composing the DODAF and its associated viewpoint products were conceived as the result of consensus debate among a group of early architecture practitioners concerning the proper composition of a complete description of a DoD architecture. In general, each practitioner came to the debate with his or her own opinion of

[2] **Semantic Completeness**: The condition of a formal system in which (1) the formal language has the power to express as well-formed formulas all of the propositions intended by the maker to be meaningful, and (2) the deductive apparatus has the power to prove as theorems all the propositions intended by the maker to be true. The second condition can be put more succinctly: all logically valid well-formed formulas of the language are theorems of the system. The first of these is also called expressive completeness; the second is called deductive completeness.

[3] Ring, S., Nicholson, D., Thilenius, J. and Harris, S. "An Activity-Based Methodology for Development and Analysis of Integrated DoD Architectures," 2004 Command and Control Research and Technology Symposium, June 15-17, 2004, San Diego, Ca.

[4] IEEE Standard 1471-2000, *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems,* defines an architecture **view** as "a representation of a whole system from the perspective of a related set of concerns." With respect to the holistic model of an architecture, this paper defines a **view** as a subset of the descriptive concepts comprising the holistic model that supports a specific purpose and stakeholder.

the appropriate products needed to compose such a description. The debate resulted in a compromise set of architecture description products and a melting pot of various methodologies and notations.

Focus on products during the debate led to embedding that approach into the conceptual structure of the DoDAF—the underlying structure of DoDAF is based upon twenty-six separate architecture description products, not a holistic model of the concepts for describing DoD architectures. The lack of a single holistic model of all the concepts as the underlying foundation of these products characterizes DoDAF as product-centric rather than data-centric. This has led to conceptual deficiencies apparent in the DoDAF today.

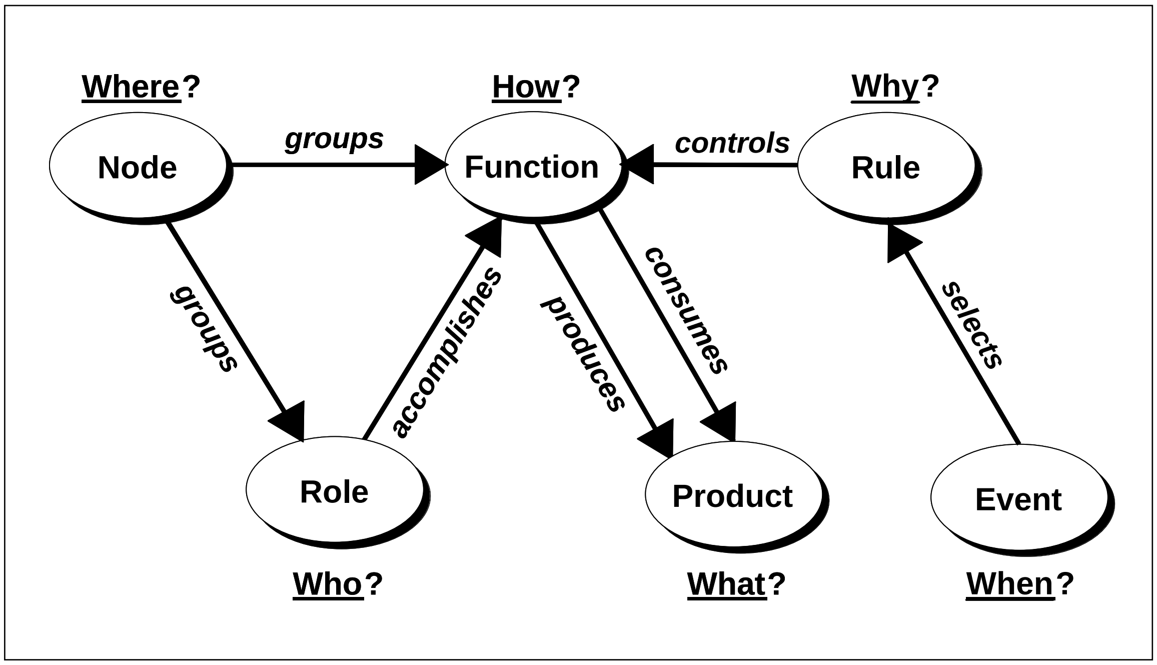

Figure 2. The ASM's fundamental pattern is a response to the six interrogatives.

As our understanding of architecture description and use has matured, a need has emerged for the specification of additional views and products. The DoDAF's product-centric approach has been perceived in practice as too rigid to allow for the creation of new views and products. In addition, DoDAF does not specify a mechanism for creation and adoption of additional views. This has led to calls to abandon DoDAF and to create new views and product sets. In reality, it is not difficult to create new DoDAF products since DoDAF also doesn't specify any constraint on new products. The difficulty has been in getting the DoDAF Working Group to agree on which new products to add and adopt and how to make the new products cohesively relate and integrate with existing ones given DoDAF's lack of an underlying holistic model. Unfortunately, the advocates of abandoning or extending DoDAF do not understand that the fundamental motivation for their own approaches is the rigidity of the current DoDAF products without an underlying holistic model to cohesively integrate them and that they are committing the same mistake. If carried to an extreme one can conceive of an endless stream of "new starts" attempting to specify the "correct" set of products that are not cohesively integrated as our knowledge of architecture practice and our needs for architecture information matures.

A more appropriate approach is to specify a holistic collection of descriptive concepts, in the form data types, that fully describe architecture and to use it as the foundation for architecture description. From this specification, views could be created as needed. We refer to this as a data-centric approach to architecture description. The ASM follows this data-centric approach.

The data-centric approach does not imply a data modeling viewpoint of architecture description. It merely implies that the specification of views and products is supported by the most flexible aggregation of architecture data types defined in the holistic model. In fact, we advocate that the need for whole-formed products should be replaced by a need for conceptually equivalent views that are constructed from lower-level architecture data types and leaving it flexible for the Communities of Interest/Practice or the architects to select their own notations for the composing architecture data types. Additionally, we suggest that Communities of Interest/Practice may identify a specific set of views as a predefined set of products if such a set supports their shared interests, but that each product is recognized as being a view (submodel)of the holistic model with specified notations.

An additional benefit of this data-centric approach is the ability to create a single integrated view of an architecture's data as a whole should it prove useful. So, rather than persistently storing completed product artifacts (instances), our approach would store the constituent data in relation to all data constituting the single integrated architecture description and allow generation of product artifacts as needed and at the time of need. The generation of product artifacts would be based on product definition in terms of the underlying architecture data types and notations. This is a more robust, extensible approach to DoD architecture description.

5. The Architect's View

The architect's role is to formalize and represent the needs of his client. This role motivates a unique view of the architecture–*the architect's view*. The architect's view is that view taken by the architect in formalizing and expressing the client's needs as an architecture description. It contains only those elements needed by the architect to describe the architecture and nothing more.

Data models as used in the development of database systems do not represent the architect's view. They include too many non-architecture artifacts. The ASM is a formal conceptual model that provides a common set of semantics for expressing the "architect's view" in describing DoD architecture. As described above the ASM data-centric approach does not imply a data modeling viewpoint of architecture description. The ASM's descriptive concepts are the same as those employed by DoD architects. Asking an architect to express architectures in a data modeling language is inefficient and ineffective (e.g., productivity reducing and error prone).

6. Key Features of the Architecture Specification Model

In addition to the primary characteristics of formal semantic completeness, a holistic conceptual model, and expression of the architect's view, the ASM incorporates a number of other key features that correct conceptual deficiencies in the current DoDAF. These additional features are all part of creating the improved semantic foundations necessary to maturing the practice of DoD architecting. In this section, *italics* are used to identify entities and relationships used in ASM.

Separation of architecture elements into six "interrogative" groups. As illustrated in figure 2 each of the primary descriptive elements composing the ASM provides the semantics for one (and only one) of the six interrogatives: how, who, where, what, when, and why. The current DoDAF provides descriptive elements with unambiguous meanings for some of the interrogatives (e.g. operational activity and system function provide "how" semantics), while it "overloads" some descriptive elements with the semantics of two or more interrogatives (e.g. operational node provides both "who" and "where' semantics)[5]. It is primarily the DoDAF's inconsistent expression of the semantics for the six interrogatives that characterizes it as semantically incomplete.

Separation of "views" currently combined in DoDAF's "Operational View." DoDAF not only overloads the meaning of architectural description elements, it also overloads the meaning of one of its primary views: the "operational view." A DoDAF view consists of a collection of architectural products (e.g. OV-1 through OV-7) that present architectural information related to a single context (i.e. operational, systems, and technical). Although not clearly explained by the DoDAF documentation, the DoDAF in fact uses the operational view to describe both a human performer-only view and an undifferentiated view that treats performers as neither human nor machine, but instead as resources composed of both.

[5] DoDAF exhibits many occurrences of **semantic overloading**, which means that one descriptive element (e.g. operational node) is used to convey more than one semantic concept. This creates ambiguity that may or may not be resolved through examination of the descriptive context in which the element is used.

In the case of "operational node" Vol II of DoDAF overloads its semantics as:

- "The critical taxonomies requiring concurrence and standardization for integrated architectures are the following: Operational Nodes that represent Organizations, Organization Types, and Occupational Specialties." (p3-10)
- "An operational node is an element of the operational architecture that produces, consumes, or processes information." (page 4-7)
- "What constitutes an operational node can vary among architectures, including, but not limited to, representing an operational/human role …, an organization …, or organization type …, and so on." (p4-7)

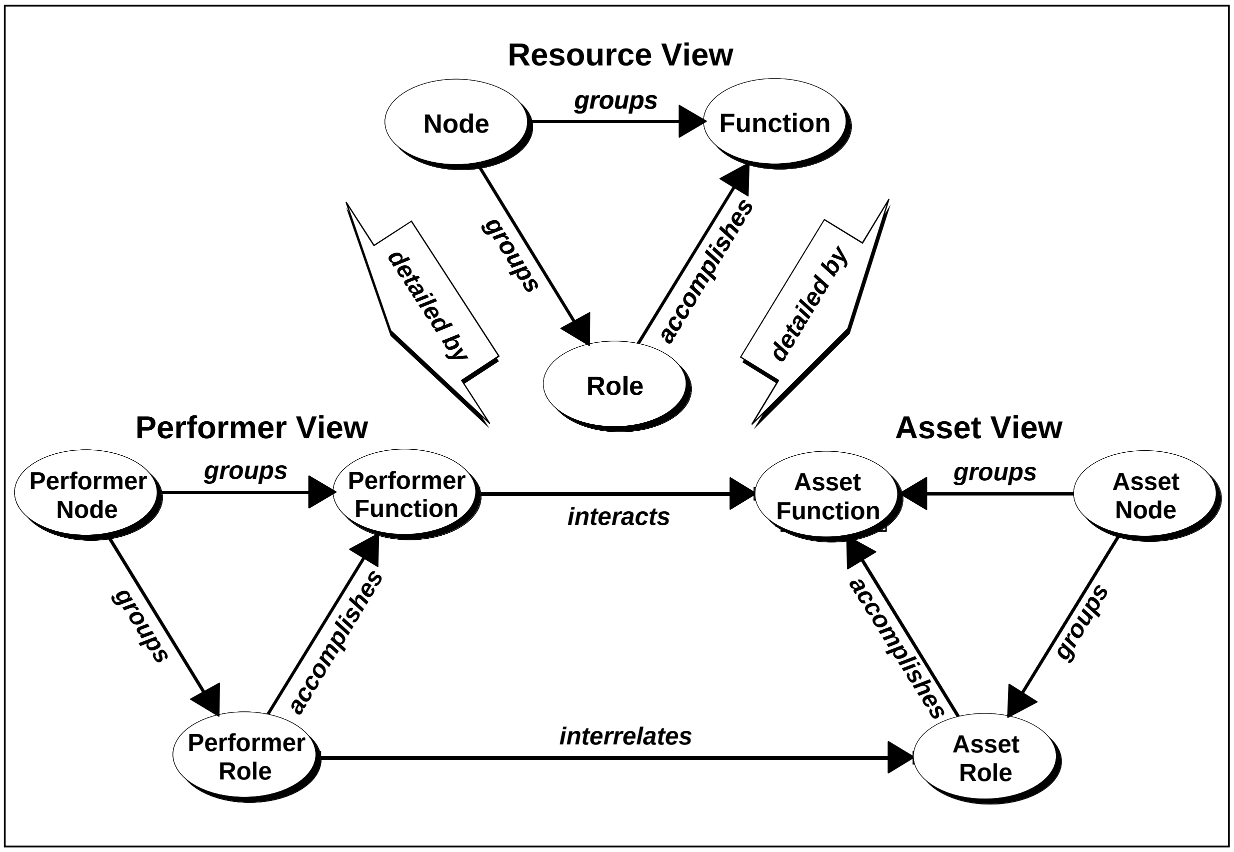

Figure 3. The ASM separates the DoDAF's overloaded "Operational View" into two unambiguously defined views—the *Resource View* and the *Performer View*.

The ASM corrects this deficiency by clearly delineating two separate views that collectively compose the DoDAF's operational view. As illustrated in figure 3, these views are: a *Performer View* that describes only human *Performers* and their relationships with other descriptive elements; and a *Resource View* that does not differentiate –whether humans or machines accomplish a function - all are *Resources* that have relationships with other descriptive elements. Figure 3 also shows a third ASM view, the *Asset View*, which replaces the DoDAF's "Systems View." The *Asset View* is analogous to the *Performer View* and describes machine-based *Assets* and their relationships with other descriptive elements.

Separation of "requirements" and "solutions" currently combined in the DoDAF. In another occurrence of overloading, the DoDAF treats an existing resource tasked to accomplish a *Function* as the same thing as a role specified as needed to accomplish a *Function*. In ASM we address this issue by treating *Role* as the set of abilities needed, or required, to accomplish a *Function*; and by treating *Resource* as the existing thing tasked to accomplish a *Function*. In effect, *Role* represents a "requirement" and *Resource* represents a "solution" to the requirement.

Figure 4 illustrates how this "pattern" is reflected in all three of the ASM's views. As with *Role* and *Resource* in the *Resource View*, *PerformerRole* and *Performer* represent the "requirement" and the "solution" in the *Performer View* while *AssetRole* and *Asset* are analogous in the *Asset View*.

Identification of fundamental symmetries and patterns within DoD architectures. The development of the ABM led to the observation that there were fundamental patterns in the arrangement of the descriptive elements employed in any one view and that those patterns were reflected in the other views to create a larger symmetry of such patterns. The previous discussion of how the semantic overloading of various elements in the DoDAF was corrected in the ASM demonstrates several of these patterns and symmetries. These patterns and symmetries were expressed further in the continuing development of the ASM. In fact, searching for their existence became a guiding principle in development of the ASM.

The creation of symmetric structuring concepts for organizing resources was another example of symmetry and patterns exploited in the ASM. The ABM effort led to the realization of the importance of organizing performers with performance abilities into organizations and then relating those organizations to the other entities in the architecture. In the ASM this symmetric structuring pattern was extended to apply to all resources—*Resources*, *Performers*, and *Assets*.

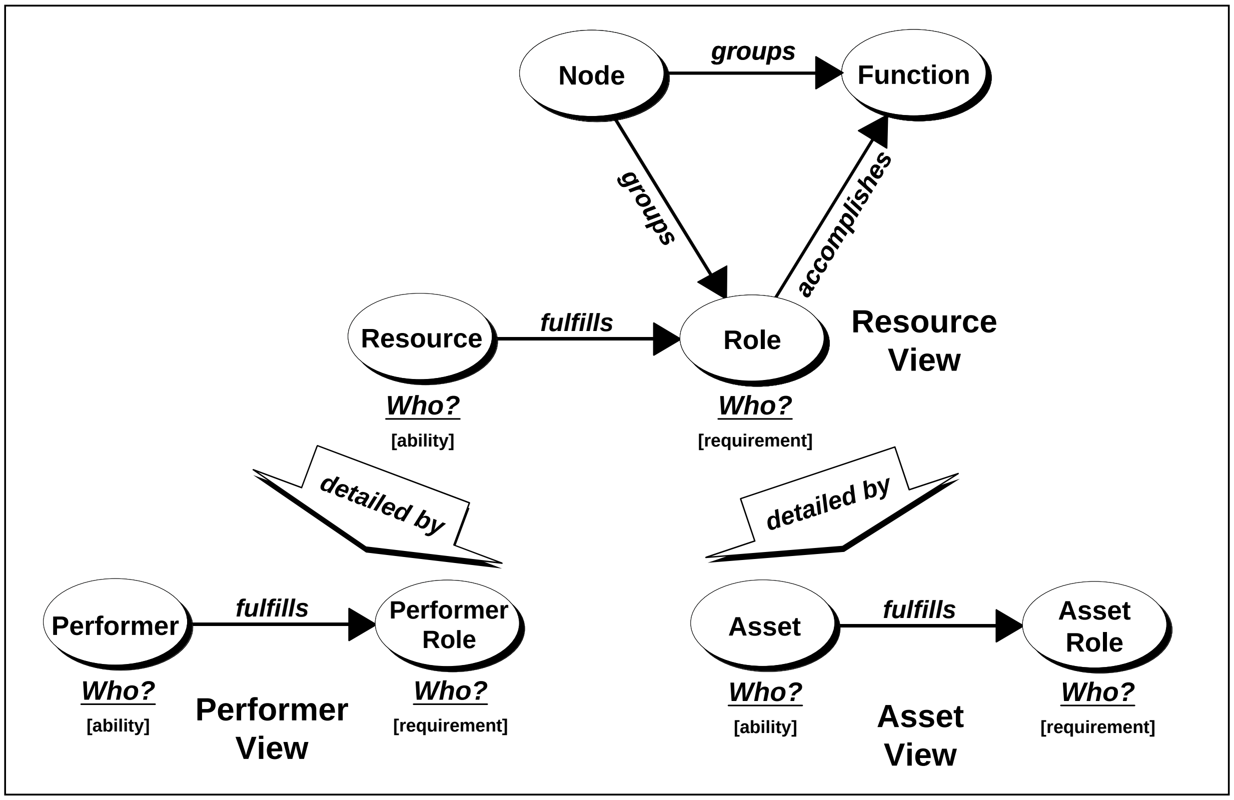

Figure 4. The ASM separates the DoDAF's overloaded "Role" into separate "requirements" and "solutions" descriptive elements

As shown in Figure 4, *Resources* (humans with machines) are detailed by *Performers* (humans) and *Assets* (machines). Through the use of a common symmetric structuring concept all three are organized into larger structures (*Resource Structures*, *Organizations*, and *Networks*) using the three basic relationship types: "control" (i.e., *supervise*, *own*), "command" (i.e., *direct*, *operate*), and "coordinate" (i.e., *collaborate*, *coact*). This pattern of structuring is applied to *Performers* (humans) to define *Organizations* (i.e., chain of command, teams), to *Assets* (machines) to define *Networks*, and to *Resources* (humans with machines) to define *Resource Structures* (i.e., unit, task force). The ability to model these structuring constructs is especially important in architecting the full range of Doctrine, Organization, Training, Material, Leadership, Personnel and Facilities, or DOTMLPF—another key feature supported by ASM.

Supports more than Information Technology Architecting. The DoDAF was designed to support modeling Information Technology systems. The *Activities* and *System Functions* in the DoDAF are only capable of producing and consuming Information Elements and Data Elements respectively as inputs and outputs. ASM addresses this limitation by supporting *Products* (*Material* and *Data*) as inputs and outputs and *Events* as outputs of *Functions* (e.g., *Function*, *Performer Function*, *Asset Function*). In addition, ASM supports the association of *Standards* with all views (i.e., *Resource*, *Performer* and *Asset*) as opposed to the Technical Views (TV) in the current DoDAF.

Models Layered and Linked Architectures. A key unresolved requirement that is not enabled by the current DoDAF is the ability to federate and integrate architectures. Separation of concerns is key to successful systems engineering that depends on layering, allocation and synthesis as primary means to deal with complexity. ASM includes several constructs that support this need.

As depicted in figure 5, *Exchanges transport* a *Product* between two *Functions* and are "function-like", while *Needlines* represent the ability to *accomplish* an *Exchange* between two *Nodes* and are therefore "role-like". This feature of ASM enables an architect to describe an *Exchange* and its accomplishing *Needline* in a production architecture as a *Function* and a *Role*, respectively, in a transportation (or communications) architecture while maintaining consistencies between the two architectures. As a result, the production and transportation architectures can be separately defined and then linked.

Other supported means that enable linking, or "federating," of multiple architecture descriptions include the use of "external" representations of the core entities (figure 3) to facilitate modeling interfaces among architectures, and through the use of levels of abstraction (*decomposes*) and multiple perspectives ("*detailed by*" as depicted in figures 3 and 4) to support allocation and synthesis.

Supports Executable Architecture Development and Analysis. The authors of the DoDAF clearly recognized the need for modeling behavior as evidenced by the inclusion of the OV-6 and SV-10 product sets. However, the DoDAF does not adequately address how to integrate these representations of rules, state dynamics and sequencing with the structural descriptions depicted in the other OV and SV products. ASM resolves this deficiency by clearly linking structural entities to behavior through an *Action_Assertion_Rule*. The basic form of the rule is:

If (*Condition*) (*Current_State*) Then (*Function*) (*Standard*) (*Constraint*) (*Next_State*)

and is depicted in Figure 6. *Condition*, which can be an *Event* generated by another *Function*, the *State* of any relevant *Resource*, or the value of any other architecture description element in the architecture, selects the appropriate rule which in turn controls the execution of the *Function*. The rule specifies the *Standards* and the *Constraints* that control the execution of the *Function*. The *Standards* include *Functional Requirements*, which specify which of the *Function's* inputs and outputs are relevant, *Functional Quality Requirements*, which specify how well (measures of effectiveness) the *Function* executes, and/or *Product Quality Requirements*, which specify the required quality of the outputs. The addition of *Standard* (beyond technical standards), *Constraint* and *Condition* into ASM addresses a key deficiency in DoDAF as these are the currency used by the DoD for assessing performance and readiness as well as key to providing the means to support portfolio analysis.

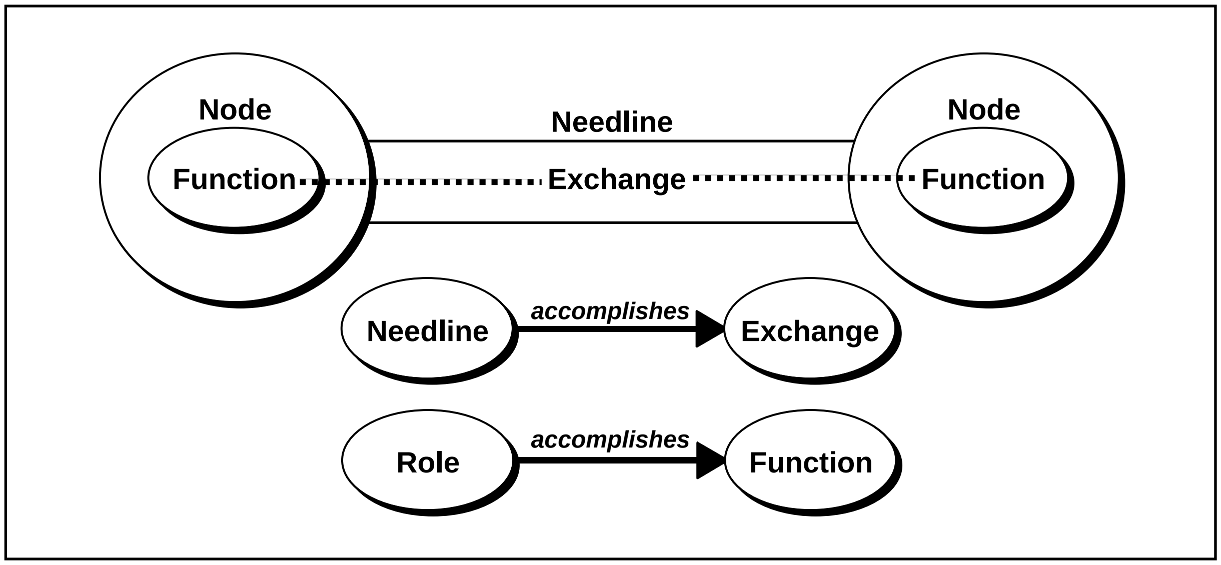

Figure 5. Layering Architecture Perspectives.

Supports Capability-Based Planning and Analysis. The DoD has adopted a Capability-Based approach to support key decisions for transformation and portfolio management. An Effect is a change to a condition, behavior, or degree of freedom resulting from the application of Capabilities, An Effect includes physical, behavioral, or knowledge changes; can be intended or unintended; and can affect enemy, friendly, and non-aligned (red, blue, or gray) forces. Therefore, the ASM Rule (figure 6) applied to an *External Function* provides a means model an *Effect*. Capabilities are defined as the combination of means (operational and support resources) and ways (activities) to achieve an Effect to a standard under specified conditions. Therefore, the ASM *Function – accomplished by – Role – fulfilled by – Resource* (figure 4) provides a means to model the ways and means of a

Capability, where the outputs (*Product* and *Event*) of *Function* are used to set the *Condition* of a *Rule* that represents an *Effect* (figure 6).

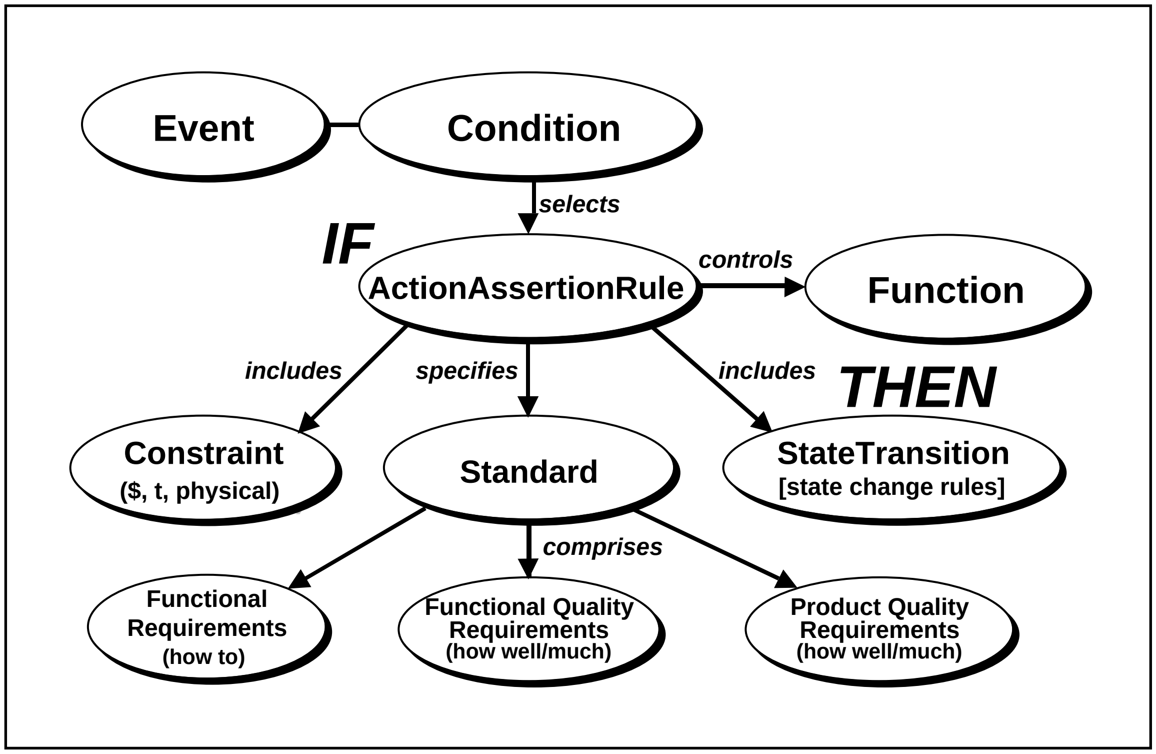

Figure 6. Rules.

Summary. In addition to those key features described above, ASM is independent of any methodology but able to support the commonly used methods (e.g., Activity-Based, Object Oriented, Rule-Based) and the tools that support them, it supports the emerging need to use Reference Models (e.g., FEA and DoD), it provides a holistic model (data centric) from which any relevant view (submodel) can be created, it allows systematic extension to incorporate new concepts and architecture description elements, and it enables cost-benefit analysis through the *Cost* and *Constraint* entities.

ASM was developed to provide a small yet powerful set of DoD architecture description concepts that provide a semantically complete shared vocabulary for the DoD architecting community of interest and serve as the foundation for the development of a next-generation DoD architecture framework - a framework that will support interoperability among DoD architecting practices and also support architecture-based analysis for DoD core processes. The development of the ASM was guided by the recognition that there are numerous conceptual deficiencies in the current or first generation of DoD architecting practice that must be resolved to support the growing number of intended uses.

7. Examples of how the ASM is being applied within the United States Air Force (USAF).

An early adapter and key application of ASM can be found in the USAF Operational Support Enterprise Architecture (OSEA) effort. The OSEA documents the Air Force operational support (e.g., business and combat support processes) transformation path with an architecture-based strategic roadmap that synchronizes functional modernization efforts both within the Air Force and with DoD/external organizations. Its primary use will be to enable analyses to prioritize operational support capability development for more effective commander / warfighter support and more efficient business processes/operations. The goal is faster, more agile and lethal combat forces supported by responsive, flexible and horizontally-integrated operational support processes and information services from an interoperable AF/Joint perspective. An initial focus has been to identify critical operational support processes that need to be measured and improved to support warfighter-focused process engineering.

To that end the OSEA team is applying the ASM to federate the numerous architectures that have been and are being developed by the various functional domains within the operational support mission area (e.g., financial, logistics, personnel, etc). Rather than attempting to integrate the disparate architecture products that have been generated by the functional domains, the OSEA team approach is to take a data centric approach. The team uses ASM entities and their relationships as key classes of architecture primitives and works with the functional area architects to disaggregate their various products into the relevant primitives and then synthesize the results into a common set of operational support primitives to which all the architecture products can be mapped. This approach has resulted in the identification of over 600 “touch points” or process intersections among the functional domains that facilitate focusing on opportunities for improvements in process, identifying and addressing portfolio gaps and overlaps, and identifying and prioritizing enterprise data opportunities.

Although still in its early stages, the resulting model has been successfully used to analyze the key functions and processes within the Operational Support Mission Area in support of an analysis to determine an appropriate Enterprise Resource Planning (ERP) solution for the Air Force.

8. Conclusions

In this paper we have described the issues with the first generation of architecting in the DoD and what we believe is the root cause. We have also described an Architecture Specification Model that was developed specifically to provide solutions and serve as the basis for the next generation of architecting. Our hope is that reconciling the lessons learned in the first generation, particularly the importance of these semantic foundations, will enable a transition to a next generation of DoD architecting practice that will be capable of architecting very large systems-of-systems and providing actionable information to the DoD’s leadership.

Acknowledgements

The authors would like to acknowledge the United States Air Force Deputy Chief of Staff, Warfighting Integration (AF/XI) for their support in the development of ASM.