10 Steps to Building an Architecture for Space Surveillance Projects

Eric A. Barnhart, M.S.
Eric.Barnhart@harris.com

Howard D. Gans, Ph.D.
Howard.Gans@harris.com

Harris Corporation, Space and Intelligence Systems; Melbourne, FL

ABSTRACT

Space surveillance is an increasingly complex task, requiring the coordination of a multitude of organizations and systems, while dealing with competing capabilities, proprietary processes, differing standards, and compliance issues. In order to fully understand space surveillance operations, analysts and engineers need to analyze and break down their operations and systems using what are essentially enterprise architecture processes and techniques. These techniques can be daunting to the first- time architect. This paper provides a summary of simplified steps to analyze a space surveillance system at the enterprise level in order to determine capabilities, services, and systems. These steps form the core of an initial Model-Based Architecting (MBA) process that can help architects and system engineers ease into the complexity of space surveillance architecture using the Department of Defense Architecture Framework (DoDAF.)

For new systems, a well-defined, or well architected, space surveillance enterprise leads to an easier transition from model-based architecture to model-based design and provides a greater likelihood that requirements are adequately fulfilled the first time, avoiding wasteful iteration. Both new and existing systems benefit from being easier to manage, and can be sustained more easily using portfolio management techniques, based around capabilities documented in the model repository. The resulting enterprise model helps an architect avoid 1) costly, faulty portfolio decisions; 2) wasteful technology refresh efforts; 3) upgrade and transition nightmares; and 4) non-compliance with DoDAF directives.

The Model-Based Architecting steps are based on a process that Harris Corporation has developed from practical experience architecting space surveillance systems and ground systems. Experience is drawn from current work on documenting space situational awareness enterprises. The process is centered on DoDAF 2 and its corresponding meta-model so that terminology is standardized and communicable across any disciplines familiar with DoDAF architecting, including acquisition, engineering and sustainment specialties.

Each step provides a guideline for the type of data to collect, and also the appropriate views to generate. The steps include 1) determining the context of the enterprise, including active elements and high level capabilities or goals; 2) determining the desired effects of the capabilities and mapping capabilities against the project plan; 3) determining operational performers and their inter-relationships; 4) building information and data dictionaries; 5) defining resources associated with capabilities; 6) determining the operational behavior necessary to achieve each capability; 7) analyzing existing or planned implementations to determine systems, services and software; 8) cross-referencing system behavior to operational behavioral; 9) documenting system threads and functional implementations; and 10) creating any required textual documentation from the model.

1 INTRODUCTION

The Department of Defense Architecture Framework (DoDAF) provides "*an overarching set of architecture concepts, guidance, best practices, and methods to enable and facilitate architecture development in support of major decision support processes.*" [1] Architecture, in the DoDAF sense, mainly addresses enterprise level architectures, but also has some applicability to solution level architectures.

An enterprise-level architecture is a strategic information asset which *"defines the mission of the organization, the behaviors and information necessary to perform the mission, the resources necessary to perform the mission, and the processes for transforming the organization and its resources to satisfy changing mission needs."* [1] Compare this to a solution-level architecture, which is "*a framework or structure that portrays the relationships among all the elements of something that answers a problem*." [2] Although DoDAF 2.0can span from enterprise-level architectures down to solution-level architectures[1], this paper addresses the enterprise-level.

Originally in versions 1.0 and 1.5, DoDAF emphasized a view-centric approach to architecture. Specific views of architectural models were specified up front and then architects would collect the architectural data to support a required view regardless of its appropriateness to the end purpose. With version 2.0, DoDAF has switched emphasis from the view-centric approach in DoDAF 1.x to a data-centric approach. In the data-centric approach, the architect collects architectural data required to support specific purposes and decisions. After the data is collected, appropriate models of the data are built and then captured in those views that support the decision makers.

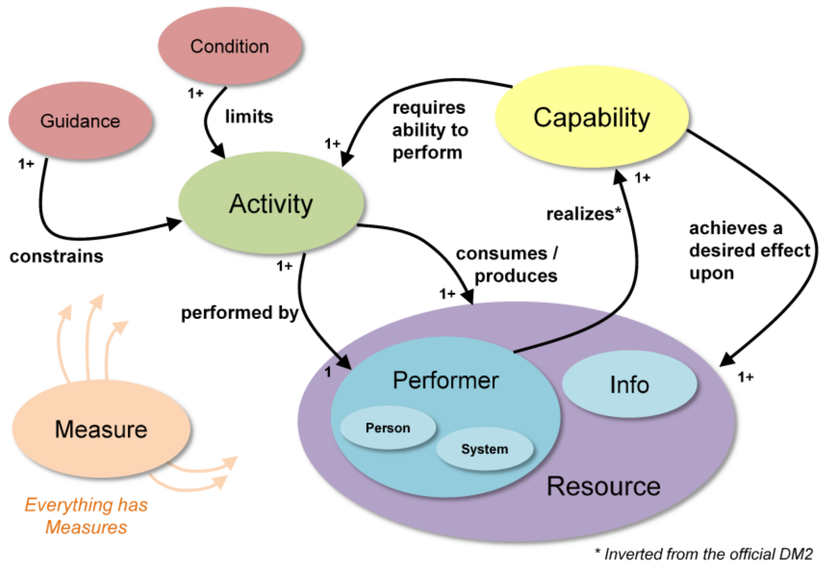

Figure 1- Simplified DoDAF Meta-Model (DM2)

In DoDAF 2.0, the architectural data, concepts and relationships are defined in the DoDAF Meta Model (DM2). The DM2 enables a common understanding of architecture concepts, allowing discovery, sharing and reuse of architectural data across organizational boundaries as well as reducing redundancies and rework. Proper implementation of the DM2 can be challenging, but application of a data centric architecture process can help programs avoid costly, faulty portfolio management decisions; wasteful technology refresh efforts; upgrade and transition nightmares; and non-compliance with architecture directives, both programmatic and legal.

The architectural data that is collected encompasses information about both logical and physical entities that describe the architecture as well as the relationships between those entities. Those entities include resources, capabilities, activities, performers, and the relationships among them. In addition, the architectural data includes information about the guidance, conditions and measures associated with the entities, their relationships and behavior. A high-level summary of the entities and relationships is provided in Figure 1.

Resources are either physical or logical entities, which include materiel, information, and data as well as active and structured entities that perform tasks, such as systems, services, organizations and people (see performers below.) Ultimately, the architecture is realized, or implemented, as a collection of resources acting together to achieve the enterprise goals.

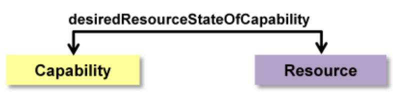

Figure 2 Relationship between Capability and Resource

Capabilities are logical entities that describe the ability of the enterprise to achieve a desired effect on one or more resources. When realized, capabilities change the states of resources, either internal or external resources, therefore DM2 relationships exist between capabilities or the changed resources as shown in Figure 2. Capabilities can interact in various ways to achieve goals of the enterprise.

[1] Solution architectures are deprecated in DoDAF version 2.02, Change 1; *"A fourth type, solution architectures trace back to the other three types but are not included in the DoD enterprise architecture."*

Figure 3 Relationship between Activity and Capability

Activities are logical entities that describe the tasks needed to fulfill the capabilities; they are the ways in which the capabilities are fulfilled. Activities may be organizational operations when they are performed at the organizational or staff levels. When performed at a system level, activities are considered to be system functions. Activities implement capabilities, therefore DM2 relationships exist between activities and capabilities as shown in Figure 3. Activities also produce and consume resources, therefore DM2 relationships exist between activities and resources as shown in Figure 4.

Figure 4 Relationships between Activity and Resource

Performers are physical resources that provide the means to execute activities, therefore there are DM2 relationships between performers and activities as shown in Figure 5. Because the performers execute activities, which in turn are the ways to fulfill a capability, the performers are the physical resources that realize the capabilities. There are DM2 relationships between performers and capabilities as shown in Figure 6.

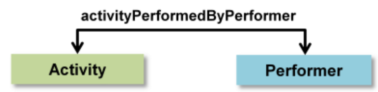

Figure 5 Relationship between Performer and Activity

Together, activities and performers are the ways and means to realize capabilities.

Throughout the DoDAF process, the architect collects information on resources, capabilities, activities, performers and the relationships among them to build a set of strategic information about the enterprise. The subsequent sections of this paper describe the DoDAF architecting process and the Harris simplified sequence of ten steps in six categories for collecting the information in a logical order. The architect should use these steps for guidance, customizing the steps for the particular subject area of interest and the purposes and needs of the architecture stakeholders.

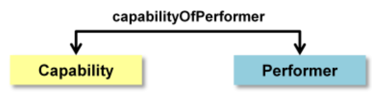

Figure 6 Relationship between Performers and Capabilities

2 DoDAF PROCESS

DoDAF provides a very high-level 6 step process to guide an architecture's development. The DoDAF process is a guide to the types of tasks needed to plan and organize the development of architectural data for given goals. The steps include:

1. Determine the Intended Use of the Architecture
2. Determine Scope of the Architecture
3. Determine Data Required to Support Architecture Development
4. Collect, Organize, Correlate and Store Architecture Data
5. Conduct Analyses in Support of Architecture Objectives
6. Document Results In Accordance With Decision Maker Needs

Each of these steps is described in the DoDAF documentation, which includes definitions of tasks for personnel roles in the DoDAF process. DoDAF provides little specific and concise guidance for architects new to DoDAF. In many architecture projects, DoDAF process steps 1 through 3 are driven by the stakeholders and management, whereas architects are concerned with step 4-6 in which architectural data is collected. The 10 Step Process described in this paper primarily addresses step 4 and provides specific and concise guidance that should be familiar to the architect.

At the core of the DoDAF process is the concept of "fit for purpose". The architecture efforts should always be focused on the underlying intended use of the architecture (defined in step one of the DoDAF process) and the strategic information needed to fulfill the purpose. There is no benefit in collecting unneeded entity information, nor in producing unnecessary views and artifacts. Keep this in mind throughout this paper. The process described herein is a simplified template of a complex process, yet the architect should still adapt the scope of activities in each step of the 10 step process so it fits the purpose of the architecture. The architect should avoid collecting unnecessary information or generating views of the data that provide little value.

3 TEN STEP PROCESS

At Harris Corporation, we recognized that DoDAF is a complex framework for architects to jump into for the first time. Rather than overwhelm architects in the full use of DoDAF from the very beginning, we developed an architecture process template that can be tailored for specific programs as needed. This 10 step process addresses mainly the collection, organization, correlation and storing of architecture data, which corresponds to DoDAF Step 4 above. Architects can then use this generic process and tailor it for individual programs. As experience grows, the architects can expand upon the process as needed for more complex architectures. The 10 Step Process is shown in Figure 7.

3.1 Establish Context

The first step should be familiar to system engineers; establish the context of the subject of interest for the architecture being developed. The architect will assemble architecture information by identifying the enterprise-level missions, high-level capabilities, performers and the relationships among these elements.

The enterprise-level missions are the top-level goals that the subject of interest achieves. For example, a ground terminal might have enterprise-level missions of receiving downlinked data, transmitting uplink command data, and others. Missions are fairly high-level and might encompass more than one scenario from a user's point of view.

The high-level capabilities represent the essential desired effects necessary to achieve the enterprise-level missions. In some cases the capabilities might correspond one-to-one with an enterprise-level mission. In other cases, a mission might require multiple capabilities to satisfy it. Since a capability achieves a desired effect on a resource, both the desired effects and resources should be noted for later steps when more detailed architecture data is collected.

Performers are the elements that perform activities, realize capabilities, and interact with the subject of interest. At this level the architect is concerned with capturing the existing performers that are known to interact with the enterprise. Performers also may include automated systems that already exist within the subject of interest. Performers may be systems, services, organizations, persons acting in specific roles, or assemblies of these elements. Although many processes and architecture capture tools use the generic term, performer, for all these elements, we recommend describing the elements by their more specific subcategory, such as Organization, System, Service, or Person, when an architecture capture tool allows it.

The context is documented with a DoDAF view, Operational Concept OV-1, which is a graphical representation of the subject area of interest. Generally an OV-1 is an artist's rendition of the enterprise, with art-based representations of the elements and their relationships in the scope of the mission. Some software-based architecture capture tools will allow the architect to place architecture elements on the OV-1, and even associate images with the elements. Having actual model elements on the OV-1 helps maintain traceability from the top most level down to more detailed architecture views.

Figure 7 Categories of the Ten Steps

3.2 Determine the Desired Effects

The mission of the enterprise will result in changes to elements or resources within the subject of interest. Deliberate changes to resources are the desired effects the subject of interest achieves. In DoDAF terms, the desired effect is captured as a relationship between a capability element and a resource element.

A capability needs to describe a specific change in a resource. Normally capabilities are named with verbs or verb phrases that describe a transform. If the name chosen for the capability does not describe a specific transform, then the architect probably has not defined the capability very well. For example, a capability named "Manage Space Objects" is a poorly defined capability. It can be replaced with capabilities such as "Track Space Objects", "Create Ephemeris", and others which describe a transform of one or more resources.

When dealing with new architectures, capabilities can be related to project plans. Capabilities may be aligned with project phases and timelines when projects are delivered in multiple increments. At this step, the architect would create relationships to project milestones if capabilities need to be tracked against schedules.

Capabilities are generally documented with DoDAF view, Capability Effects CV-1, Capability Taxonomy CV-2, and Capability Schedules CV-3. Often a textual description of the capability is stored in a database such as DOORS. Project Views, PV-1 and PV-2 can provide additional information about the project plans, including the project phases and milestones as they relate to capability delivery.

CHECKPOINT: At this point the architecture can start to support cost-effective, accurate portfolio management decisions and tracking against project plans and funding.

3.3 Define the Operational Organization of Performers

Having identified the performers for the enterprise and the capabilities that are achieved, the next step is to model the structure and relationships between the performers necessary to realize the capabilities. The architect should model the performers as systems, services, organizations, or personnel, and capture their interdependencies and relationships. The relationships will capture the movement of resources between the performers which are necessary to realize the capabilities. The flow of resources between performers indicates the interfaces. The resources should be recorded for later steps.

At this point, the model is intended to show overall operations and relationships at an organizational level. This model shows how organizations work together, NOT how systems work internally. Since organizations will own systems, or use systems for communications and processing, it's often valuable to include the systems in this model for completeness. If the model is entirely based on systems and interactions, then the architect should defer this model to later steps. It is a common mistake to attempt to model the operations of systems at this point, but instead the architect should concentrate on organizational and personnel interactions at this step.

Operational performers are generally documented with DoDAF views, Organizations and Resources OV-2, and Organizational Relationships OV-4.

3.4 Build the Data Dictionary

The purpose of this step is to build a single source of truth for the definitions of the information and data resources in all the models in the architecture. By now the models have identified resources that flow between performers. Those resources might be data, information, or even materiel. The definition of each of these is captured in a data dictionary and made available for consistent use throughout the architecture and in future steps. Every resource that flows between performers must be identified, named and defined. The data dictionary provides consistency and reduces confusion and conflicts in later steps.

The data dictionary is documented in DoDAF views Conceptual Information DIV-1, and Data Requirements Model DIV-2. (The Data Implementation DIV-3 is seldom needed at this point.) The DIV-1 can be created for very high-level definitions. The preferred representation in this process is a DIV-2. It can be implemented in a database of terms and attributes, but class diagrams, block diagrams or even entity-relationship diagrams can fulfill this. A DIV-1 on the other hand, is often just a textual representation. The data dictionary is maintained throughout the architecture's life cycle.

In additional to the data dictionary, a summary of the resources exchanged between performers should be created to supplement the OV-2. The summary can be accomplished with a matrix of the resources and the performers that produce and consume them. The DoDAF view for this matrix is Organizations, Activities & Resources OV-3.

3.5 Define the Capability Transforms

The purpose of this step is to rigorously define the desired effects of the capabilities, since ultimately the architecture will be validated against the desired effects. For each capability the architecture should capture the major activity or transform that defines the desired effect. Also, any constraints and guidance for each capability should be captured. In some cases, it might be useful to document the desired effect by recording input resources, output resources, and the transform that occurs. In other cases, the desired effect may be described as a state change in a resource. When a resource is data or information, the state change might include simple creation, update, and deletion state changes.

The architect should update the capability statements to include the description of the desired effects and the applicable resources, using the textual database introduced in step 2 if the architecture is using a database.

CHECKPOINT: At this point the architecture has consistent and accurate definitions of the capabilities and desired effects thus assuring accurate communications to support lower risks and lower costs. This benefit is especially evident when working with interfaces and within systems of systems.

3.6 Analyze the Behavior

Now that the architecture captures a detailed description of the desired effects upon the resources, the next step for the architect is to analyze the behaviors required to achieve those desired effects. The architect determines the operational activities that are needed to fulfill a capability, and assembles them into one or more operational scenarios or threads. The scenarios or threads can be analyzed using a variety of techniques, although the most common is building flow charts, activity diagrams, or sequence diagrams. At this stage, the analysis should be kept at an operational level, that is, the behavior should reflect the interactions and interfaces of the organizations, but not system to system interfaces. Since systems are generally owned by organizations, in this step it might prove useful to include operational interactions with systems in their parent organizations, with the understanding that system to system relationships are defined later. Systems are fully addressed in the next step.

In this step, guidance, conditions and measures also should be associated with the specific operational activities that fulfill the capabilities. Collectively, the activities and their guidance, conditions and measures form the basis of an implicit requirements model which includes functional requirements and non-functional requirements. The requirements model can be translated to written requirements in later systems engineering stages.

Operational behavior is generally documented in DoDAF views Operational Activities OV-5b, Operational Rules OV-6a and Operational Activity Sequences OV-6c. There may be many different scenarios or user stories modeled in this step, so there may be many instances of OV-5 and OV-6 views to illustrate the models. An often-made mistake is to think there is a single model and corresponding view of each OV type; in fact, there can be many models documented with a view for each type of DoDAF view collected within a viewpoint.

CHECKPOINT: At this point the architecture includes capabilities, desired effects, activities and guidance and constraints, which are sufficient artifacts to support smart, scenario-based, what-if decisions. Constraints can be varied and different operational scenarios can be explored.

3.7 Analyze the Configurations

If the subject area of interest includes existing systems, the next step is to analyze their as-is configurations. If the subject area of interest requires new systems, the next step includes modeling the to-be configurations of systems to the degree that they are known. Both configurations are modeled using similar techniques.

Now that the architecture has identified the systems, the next steps for the architect is to model the structure and relationships between the systems, including their interdependencies and relationships. The architect should include subsystems where appropriate to fully explain the structure. The relationships will capture the movement of resources between the systems, which is necessary to execute the activities defined earlier. The flow of resources between systems indicates where the interfaces exist. The resources should be recorded for later steps. Resources discovered at this level are added to the data dictionary. Next, identify and document the means by which the resources move between systems, which indicate the physical interfaces that connect the systems together.

The configurations are documented in DoDAF views, System Composition and Interfaces SV-1 and System Interface Means SV-2. The Data and Information Views need to be updated with new resources that are discovered during this step, or with new attributes discovered for existing resources in the data dictionary.

3.8 Define Service Functions

Not all subject areas of interest include services, so this step is optional. If the subject area of interest includes existing services, the next step is to analyze the as-is set of services. If the subject area of interest requires new services, the step includes modeling the to-be services to the degree that they are known. Much like the system entities were modeled in step 7, the architect should identify the services, as well as the relationships and interfaces between services. Next the architect should identify and document the resources that move between services to fulfill their function.

The service functions are documented with Service Views Services SvcV-1, and Services Interfaces SvcV-2. As with system modeling, the Data and Information Views need to be updated with new resources that are discovered during this step, or with new attributes discovered for existing resources in the data dictionary.

3.9 Define System Behaviors

Now that the architecture captures a detailed description of the systems and the services, if any, the architect is ready to analyze the functional activities (functions) the systems performs to implement the operational activities. The functions are associated with a system and linked to one or more operational activities that the functions support. If the subject area of interest includes existing systems, the architect models their as-is functionality. If the subject area of interest requires new systems, the architect models the to-be configurations of functions to the degree that they are known. Both as-is and to-be behaviors are modeled using similar techniques.

The architect uses the operational scenarios/threads that were defined earlier and re-analyzes them to show how they are performed in terms of system functions. At this stage, the analysis moves to a lower level; that is, the behavior should reflect the interactions and interfaces of the systems, services and functions in the configuration. The tools used to analyze system behavior are similar to the tools used to analyze operational behavior, that is, flow charts, activity diagrams, or sequence diagrams.

Also similar to the operational behavior analysis, constraints, guidance, and rules should be associated with specific functions at this point. The analysis at the system level might very well derive more specific constraints, guidance and rules from those at the operational level. For example, operational performance constraints may be portioned and allocated to multiple functions and systems. Collectively, the functions and their constraints, guidance and rules form the basis for an implicit systems requirements model which includes both functional and non-functional requirements. The requirements model can be translated to written requirements in later system engineering stages.

At this point the architect can cross reference system functions to operational activities. Because the activities implement capabilities, the architect can also cross reference the systems to their realized capability or capabilities. Given properly maintained cross references, the architect can trace from an initial capability to a set of activities, on to the functions that implement the capabilities, on to the systems that perform the functions, and come full circle back to the capabilities that are realized by the systems. (See Figure 8.) Properly maintained cross-references can assist in the validation of systems against capabilities.

System behaviors are documented in DoDAF views, System Functions SV-4, Systems and Operational Activities SV-5a, and Systems and Capabilities SV-5b.

CHECKPOINT: At this point the structural mapping to behavior supports low-risk upgrade and smart technology refresh and transition schedules—including across systems of systems.

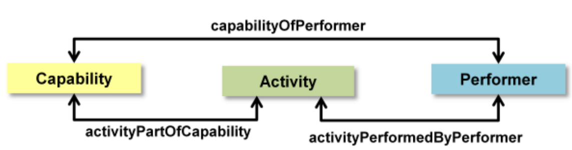

Figure 8 Cross References Support Validation

3.10 Document the Architecture

Inevitably, written documentation of the architecture will be required, perhaps for document delivery, or simply because the stakeholder lacks access to a compatible automated architecture tool. The forms of the final architecture documentation will be determined by the requirements to communicate the architecture to other stakeholders, and their needs and compatible viewing methods.

Documentation products might include word processing documents, database exports, spreadsheets, and XML exports as needed by the stakeholder.

CHECKPOINT: At this point the model contains all the information needed to generate standard documentation for communicating with other agencies, contractors, and organizations according to their needs and goals for the architecture.

4 SUMMARY

DoDAF is a complex framework and has a fairly shallow learning curve with a long time to become proficient. At Harris we have reduced the time to proficiency in DoDAF by providing a process template for collecting data in an orderly fashion. Although it is a smaller subset of the DoDAF process, our process template can be tailored for specific programs and "fit for purpose". Use of the 10 Step process template has helped programs and IRADs at Harris and may help other programs in the adoption of DoDAF.

5 REFERENCES

[1] DoD Architecture Framework Version 2.02 Change 1, 2.02 Change 1 ed., Vols. II - Architectureal Data and Models, United States Department of Defense, 31-January-2015.

[2] DoD Architecture Framework Version 2.0, Volume 1: Introduction, Overview, Concepts - Manager's Guide, Department of Defense, 2009.

.