DoD Architecture Framework

Version 2.02, Change 1

Volume IV: DoDAF Journal

Best Practices

31 January 2015

This page is intentionally left blank

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1 DoDAF Background

1.1.1 Historical Evolution of DoDAF

The Command, Control, Communications, Computers, and Intelligence, Surveillance, and Reconnaissance (C4ISR) Architecture Framework v1.0, dated 7 June 1996, was created in response to the passage of the Clinger-Cohen Act. It replaced the Technical Architecture for Information Management (TAFIM). Version 2.0 of the C4ISR Framework was published on 18 December 1997.

The DoDAF V1.0, dated 30 August 2003 restructured the C4ISR Framework V2.0 and broadened the applicability of architecture tenets and practices to all JCAs rather than just the C4ISR community. DoDAF V1.0 addressed usage, integrated architectures, DoD and Federal policies, value of architectures, architecture measures (metrics), DoD decision support processes, development techniques, analytical techniques, and moved towards a repository-based approach by placing emphasis on architectural data elements that comprise architecture products. DoDAF V1.0 was supported by a CADM which provided for data organization and sharing.

DoDAF V1.5, dated 23 April 2007, was a transitional evolution of the DoDAF V1.0, provided additional guidance on how to reflect net-centric concepts within Architectural Descriptions, included information on architectural data management and federating architectures through the Department, and incorporated the pre-release CADM V1.5, a simplified model of previous CADM. DoDAF V1.5 provided support for net-centricity concepts within the context of the existing set of architectural views and architecture products.

DoDAF V2.0 expands previous framework development efforts to capture architecture information about net-centricity, support Departmental net-centric strategies, and describe service-oriented solutions that facilitate the creation and maintenance of a net-centric environment. DoDAF V2.0 will continue to be updated in the future as it improves its support for the increasing uses of architectural data and its derived information to meet the growing needs of decision makers in a Net-Centric Environment (NCE).

1.1.2 DoDAF V2.0 – The Need for Change

Over time, and as experience with architecture has grown within the Department, it has become obvious that there are two types of architectures. The first and most traditional type is the ***Program Level or Solutions Architecture***. This architecture has been required, defined, and supported by major Departmental processes for solution evaluation, interoperability, and resource allocation. ***Enterprise Architecture***, the second type of architecture, provides a roadmap for change as well as a context and reference for how and where programs fit within a larger ‘enterprise’ picture. Because of the complex structure and function of the DoD, an enterprise can be defined at the Department level, the JCA level, and the Component level. These ‘tiers’ need architecture content at their level to guide and direct their lower level mission requirements. The JCA and Component tiers are critical to address the high-level capabilities and semantics of a specific JCA or Component within the enterprise so that federation of individual architectural data is possible.

An architecture can represent either a current (i.e., “As-Is” or baseline) viewpoint, or a future, desired (i.e., “To-Be”) viewpoint. When the architecture is a baseline viewpoint, it should illustrate the enterprise, or a portion of it, as it exists at some point in time. The future state architecture depicts the changes that are desired (whether operational, system/service-centric, or technology-driven) at some future point in time, and the strategies, programs and projects that are employed to achieve the desired transformation[1]. The future view extends beyond details or summaries of operational and systems solutions, and includes program plans, programmatic status reporting, financial and budget relationships, and risk management assessments, along with a transition plan.

DoDAF V2.0 supports the development and use of both solution architectures and enterprise-wide architectures to illustrate the context for change at the capability and component level, and/or the interdependencies among the components or capabilities. Future updates and revisions to DoDAF will extend beyond the solution space to provide standard mechanisms for communicating program plans, financial information, and project status. These future updates will more fully support the ability of managers and executives to evaluate and direct their programs. Without such standards, interdependent programs and projects will continue to be evaluated separately, and managed as individual budgets and consequently as stovepipe solutions. Such an advance in enterprise architecture would facilitate PfM as a whole, help ensure that program direction is coordinated and accountable, and address impact and alternative analysis across programmatic boundaries.

1.1.3 Architecture Focus

DoDAF V2.0 focuses on the use of architecture throughout the various tiers of the department as they relate to operational and transformational decision-making processes. Working directly with process owners, through a set of comprehensive workshops, to validate and extend architectural data content, and provide meaningful and useful architectural views for their decision-making, DoDAF V2.0 provides better harmonization of architecture content and process requirements. Additionally, these tailored architectures can be shared and provide insight into best practices that benefit programs, architects, and process owners. Architectural data content also includes data defining generic performance measures (metrics), capabilities, and the relevant PfM data, all of which are analytically useful to process owners and systems engineers.

1.1.4 Shifting from Product-Centric to Data-Centric Focus

Both the prior versions of DoDAF and earlier C4ISR versions of the Architecture Framework have emphasized reusable and interoperable data organized into ‘products’ (e.g., graphical representations or documents). DoDAF V2.0 places its emphasis on utilizing architectural data to support analysis and decision-making, and greatly expands the types of graphical representations that can be used to support decision-making activities. With appropriate architectural data, it is possible to support innovative and flexible presentation of the architectural data in a meaningful, useful, and understandable manner through the views described in Volumes 1 and 2.

1.1.5 Assumptions

Development of DoDAF V2.0 is guided by several assumptions. These are:

[1] Derived from OMB Circular A-130 that an enterprise architecture consists of a baseline architecture, a target architecture, and a transition strategy.

a. The DoDAF will continue to evolve to meet the growing needs of decision makers in a NCE.
b. As capability development continues, and Infrastructure continues to mature, architectures will increasingly be a factor in evaluating investments, development, and performance at the various portfolio levels.
c. As the DoD increases its use of architectural data and its derived information for decision-making processes, architects will need to understand how to aggregate the data as useful information for presentation purposes at the enterprise level.
d. The DoDAF plays a critical role in the development and federation of architectures. It will continue to improve its support for the increasing uses of semantically linked and aligned architectural data.
e. Architectural data described in DoDAF is not all-inclusive. Architectures may require additional data, and it is expected that architecture developers at all levels will extend the set of architectural data as necessary.
f. Prescription of required architect data sets or views to be included in an architecture is a decision made by process owners based on the purpose of the architecture, not by DoDAF. Some specific minimum architectural data will be described in DoDAF for the exchange of architectural data in the federated environment, and will be included in the architect data set supporting products required by the process owners.

1.1.6 Relationships to Other Architecture Frameworks

DoDAF is designed to align, map, and socialize with industry, allies with their own national frameworks, and other reference documents required for interoperability, reuse, and operational purposes. The DoDAF approach to alignment is to incorporate relevant concepts into DoDAF from other frameworks and reference documents and understand, integrate and describe the differences.

1.1.6.1 Frameworks

Frameworks are documents that describe useful methods, practices, and procedures for developing Architectural Descriptions. Frameworks can be prescriptive (e.g., their use is required) or descriptive (i.e., their use is recommended). DoDAF has both prescriptive and descriptive elements that organizations within the Department require its use in developing Architectural Descriptions that respond to their mandates.

1.1.6.1.1 Federal Enterprise Architecture Program

The FEA promotes shared development for common Federal processes, interoperability, and sharing of information among the Agencies of the Federal Government and other Governmental entities through the use of a set of reference models and practices that apply to all Federal agencies in the Executive branch. The DoDAF leverages the FEA construct and core principles to provide the Department with the enterprise management information it needs to achieve its strategic transformation goals, while ensuring that upward reporting and review can be accomplished against the FEA.

1.1.6.1.2 The Zachman Framework

The Zachman Framework provides a formal and highly structured way of defining an enterprise. It is based on a two-dimensional classification model, displayed as a matrix, which utilizes six basic communication interrogatives (What, How, Where, Who, When, and Why) and intersecting six distinct model types which relate to stakeholder groups (Strategists, Executive Leaders, Architects, Engineers, Technicians, and Workers) to give a holistic view of the enterprise. Decomposition of the matrix allows for several diagrams of the same data sets to be developed for the same architecture, where each diagram shows an increasing level of detail. DoDAF V2.0 supports the needs of various stakeholders' perspective by supporting various levels of abstraction and granularity.

1.1.6.1.3 The Open Group Architecture Framework

TOGAF is a comprehensive architecture framework and methodology, which enables practitioners to design, evaluate, and build an appropriate architecture for the organization. The TOGAF Architecture Development Method (ADM) supports the TOGAF architecture development approach for architectures that meet business needs. TOGAF's ADM prescribes methodology, not products, or modeling notation, and should be used with other architecture frameworks as appropriate. TOGAF evolved from the DoD Technical Architecture Framework for Information Management (TAFIM). DoDAF V2.0 and TOGAF both provide a practical, design agnostic method for creating enterprise architectures. The DoDAF V2.0 "Fit-for-Purpose" approach for developing views, presentations, or generated reports are based on TOGAF's business, data, application, and technology views.

1.1.6.1.4 The Ministry of Defense Architecture Framework

MODAF is based on the DoDAF V1.0 baseline, which it represents through the MODAF Meta Model (M3). MODAF retains compatibility with United States modeling initiatives, but is specifically designed to support architecture modeling for the UK Ministry of Defense (MOD) business. MODAF uses aspects of the existing DoDAF with additional viewpoints (acquisition, capability) that are required to support MOD processes, procedures, and organizational structures. The additional viewpoints provide a rigorous method for understanding, analyzing, and specifying capabilities, systems, System of Systems (SoS), business processes, and organizational structures. DoDAF V2.0 incorporates the data elements from MODAF required to support an acquisition and capability views in DoDAF V2.0.

1.1.6.1.5 NATO Architecture Framework

The NAF provides the rules, guidance, and product descriptions for developing, presenting, and communicating architectures across NATO and other national boundaries. Earlier versions of NAF were tightly coupled to the DoDAF. NAF's new features include a capability, service-oriented, and program view. DoDAF V2.0 has adopted the capability and program views described in NAF as defined by NAF.

2 Application of DoDAF to Specific Roles and Responsibilities

2.1 What DoD Managers and Executives Need to Know About DoDAF

Architecture development is a management tool that supports the decision-making process. A Process owner (an executive responsible for a specific process or program) has the direct responsibility for ensuring that a particular process or program works efficiently, in compliance with legal and departmental requirements, and serves the purpose for which it was created. Periodically a review and evaluation of the efficiency of the program or process is required. Those requirements for review, to include those detailed in legislation such as the Clinger-Cohen Act and OMB Directive A-130, include the need to create or update an information architecture supporting any budget requests for funding of those projects and processes.

A manager or executive may delegate the responsibility for creation of the architecture to an architect with the professional qualifications needed, along with an architecture development team. However, that delegation of authority does not alter the continuing responsibility of the executive or manager. As described throughout this volume, the decision-maker needs to be actively involved in the architecture development process and support Architectural Description development. Active involvement means that the decision-maker:

a) Identifies the Purpose and Scope for the Architecture. The 6-Step Architecture Development Process (depicted in Section 7.1.1 6-Step Architecture Development Process) provides a structure for development of scope and purpose.
b) Transmits to the architect and development team the scope and purpose of the architecture effort, along with those goals and objectives that support the need.
c) In conjunction with the architect, identifies the general data categories needed for architecture development; assists in data collection and validation.
d) Determines desired views and presentation methods for the completed architecture.
e) Meets frequently with the architect and development team to ensure that the development effort is on target (i.e., is "Fit-for-Purpose") and provides new direction, as required to ensure that the development effort meets established requirements.

> **Figure 1.5-1** is a more detailed view of the 6-Step Architecture Process, and depicts the sub-steps that the decision-maker needs to perform in coordination with the architect within the 6-Step Architecture Development Process described in Section 7. In each step, the 'Meta-model Groups' referred to by the step is that data in the Meta-model Groups in DM2 described in this volume, and more technically in volume 2.

The decision-maker generally performs the following functions:

1. Reviews the Purpose (Step 1 of the DoDAF Methodology) and Scope (Step 2) with the Architect. In order for the architecture to be "Fit-for-Purpose," the decision-maker needs to provide the list of the categories of data needed and a description of how the data will be used to the Architect. The decision-maker, not the Architect, is the subject matter expert for the problem to be solved, the decision to be made, or the information to be captured

and analyzed. The architect is the technical expert who translates the decision-maker's requirements into a set of data that can be used by engineers and analysts to design possible solutions. Determining the data needed and the requirements (Step 3.1) to be applied is an important responsibility for the decision-maker and cannot be delegated to the Architect.

2. Reviews the Views, Concepts, Associations, and Attributes that the architect has determined meets the data needs and requirements (Step 3.2). The Models, Concepts, Associations, and Attributes required are determined in the Architect's detailed process (Step 4.1 and 4.2) described in Section 1.6 of Volume II.
3. Assists with data collection, or provides the data needed (Step 4.1) using the architecture collection method described in the Architect's detailed process (Step 4.3) found in section 1.6 of Volume II. In that step, the architect determines the appropriate collection methods for the "Fit-for-Purpose" needs. Section 2 of Volume II contains a Method subsection for each of the Meta-model groups, which provides potential collection methods. Step 3 includes those actions taken to ensure that data integration occurs across all views created as a part of the architecture development effort.
4. Verifies with the architect that the data collected meets the need (Step 5.1) described in use-cases to support the analysis that will be performed in Step 5 of the 6-Step Architecture Development Process. The architect has collected the architectural data that will meet the decision-maker's purpose ("Fit-for-Purpose") and support the decision review processes. Section 2 of Volume II contains a Use subsection for each of the Meta-model groups, which provides example uses.
5. Determines the appropriate views for the "Fit-for-Purpose" needs and support to decision deliberations (Step 6.1). Volume II, Section 3 contains a DoDAF Viewpoints & Models subsection which describes each of the DoDAF-described Models. This step results in presentation creation in Step 6 of the 6-Step Architecture Development Process.

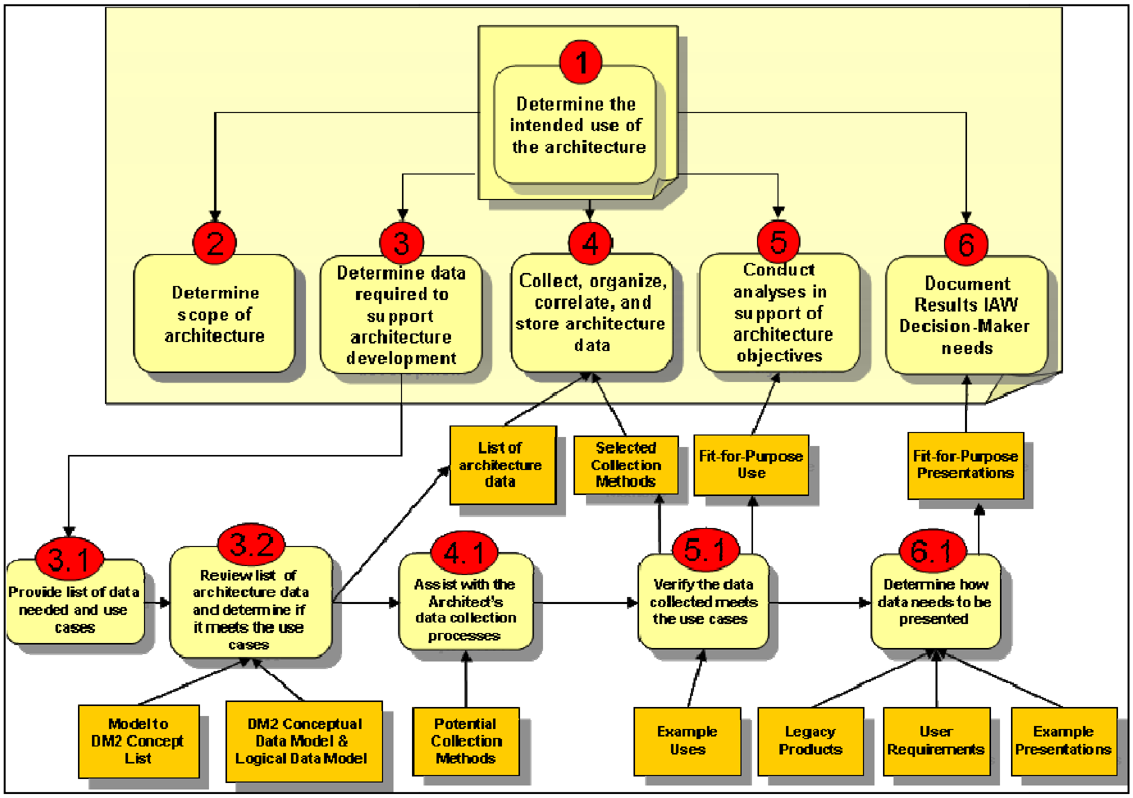


Figure 1.5-1: What the Decision-Maker Needs to Do

Working with the architect and team, the decision-maker has a critical role in ensuring that the architecture not only supports the creation of executable requirements that will achieve the desired outcome, but also that senior executives and managers can view the desired solution in an understandable and logical manner.

2.2 What Does the DoD Manager (Decision maker, Process Owner, Executive, or Stakeholder) Need to Do

The DoD Manager identifies the Purpose and Scope for the Architectural Description and gains agreement with the architect. Within the 6-Step Architecture Development Process (described in Volume I, Section 7.1.1, 6-Step Architecture Development Process), the DoD Manager needs to be involved in the entire process to support the Architectural Description development.

Figure 1.5-1 depicts the sub-steps that the DoD Manager needs to perform in coordination with the architect within the 6-Step Architecture Development Process.

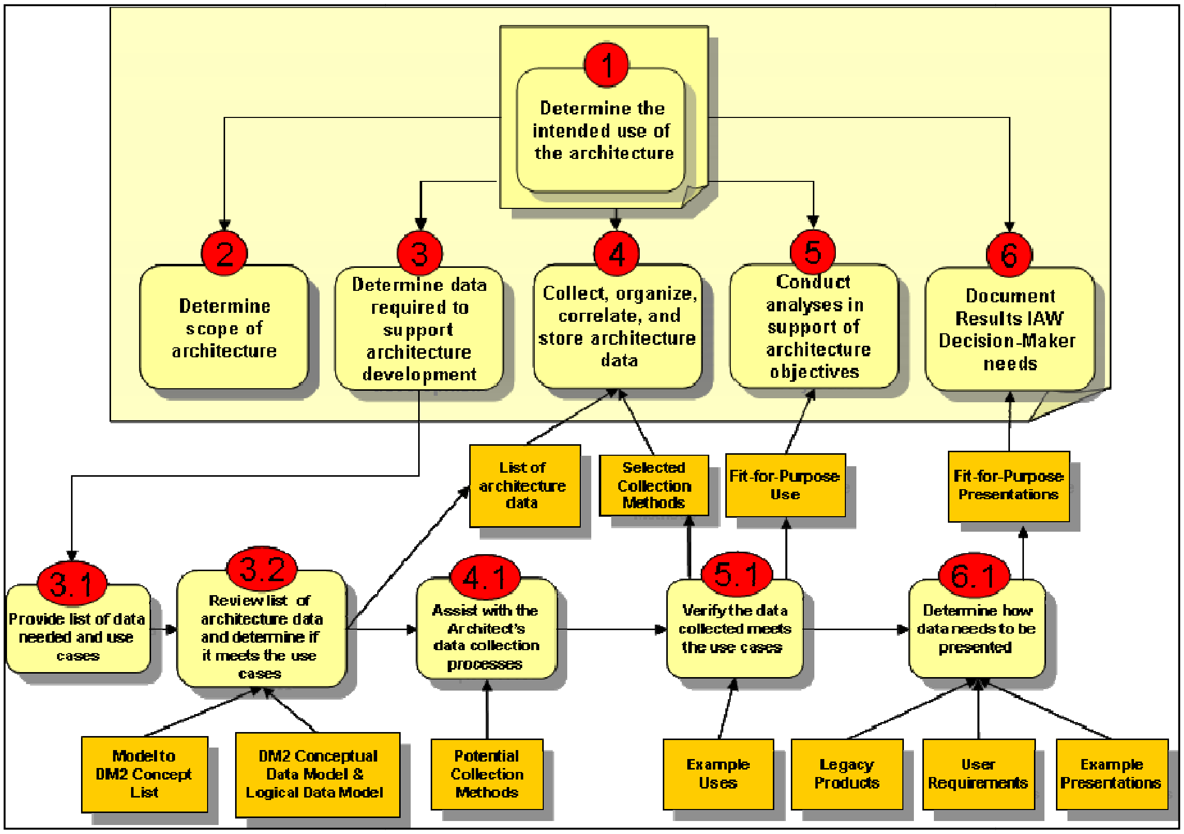


Figure 1.5-1: What Does the Decision-Maker Need To Do?

The detailed steps are:

- **Step 3.1**: After the DoD Manager has determined the Purpose and Scope as part of Steps 1 and 2 of the Architecture Development Process, the DoD Manager needs to review the Purpose and Scope with the architect. In order for the architecture to be “Fit-for-Purpose”, the DoD Manager needs to provide the list of data needed and the usage of the data (use-cases) to the architect. The DoD Manager, not the architect, is the subject matter expert. The DoD Manager, in concert with the architect, will determine the problem to be solved, the decision to be made, or the data and information to be captured and analyzed. Determining the data needed and the uses is an important responsibility for the DoD Manager and can not be delegated to the architect.
- **Step 3.2**: The DoD Manager reviews the DoDAF-described Models and Fit-for-Purpose Views, Concepts, Associations, and Attributes that, according to the architect, meet the data requirements and use-cases. The Models, Views, Concepts, Associations, and Attributes required are determined in the architect’s detailed process (Step 3.2) described in Section 1.6 of Volume II.
- **Step 4.1**: From the architect’s detailed process (Step 3.5) described in Section 1.6 of Volume II, the architect determined the appropriate collection methods for the “Fit-for-Purpose” needs. Section 2 of Volume II contains a Method subsection for each of the Meta-model groups which provide potential collection methods. The DoD Manager needs to assist or provide the data needed using the architecture collection method.

- **Step 5.1**: The architect has determined the architectural data that will meet the DoD Manager's purpose ("Fit-for-Purpose") and support their decision processes (use-cases). Section 2 of Volume II contains a Use subsection for each of the Meta-model groups which describe example uses. The DoD Manager needs to verify that the data collected meets their needs (use-cases) to support the analysis that will be performed in Step 5 of the 6-Step Architecture Development Process.
- **Step 6.1**: Based on data collected in Step 4 and the Use-cases, the DoD Manager needs to determine the appropriate presentations for the "Fit-for-Purpose" needs and to support their decision processes. This step should support the presentations that will be created in Step 6 of the 6-Step Architecture Development Process.

2.3 What Does the Architect Need to Do

Using the DoDAF V2.0 Volumes and the DoDAF Journal, the architect needs to perform two key activities:

- Develop the Architectural Description.
- Enable use of the Architectural Description in the solution implementation.

The following subsections describe the architect's activities in more detail.

2.3.1 Develop the Architectural Description

Once the Architectural Description Purpose and Scope are identified, what does the architect need to do? Within the 6-Step Architecture Development Process (described in Volume I, Section 6.1.1, 6-Step Architecture Development Process), in Step 3 the architect determines the data needed to support the Architectural Description development.

In each step, the Meta-model Groups referred to by the step is that data in the Meta-model Groups in the DoDAF Meta-model contained in this volume. **Figure 1.4.1-1** depicts the sub steps that the architect needs to perform within the 6-Step Architecture Development Process. Some of these sub steps are performed in concert with the decision-maker, but the architect has more steps than the decision-maker.

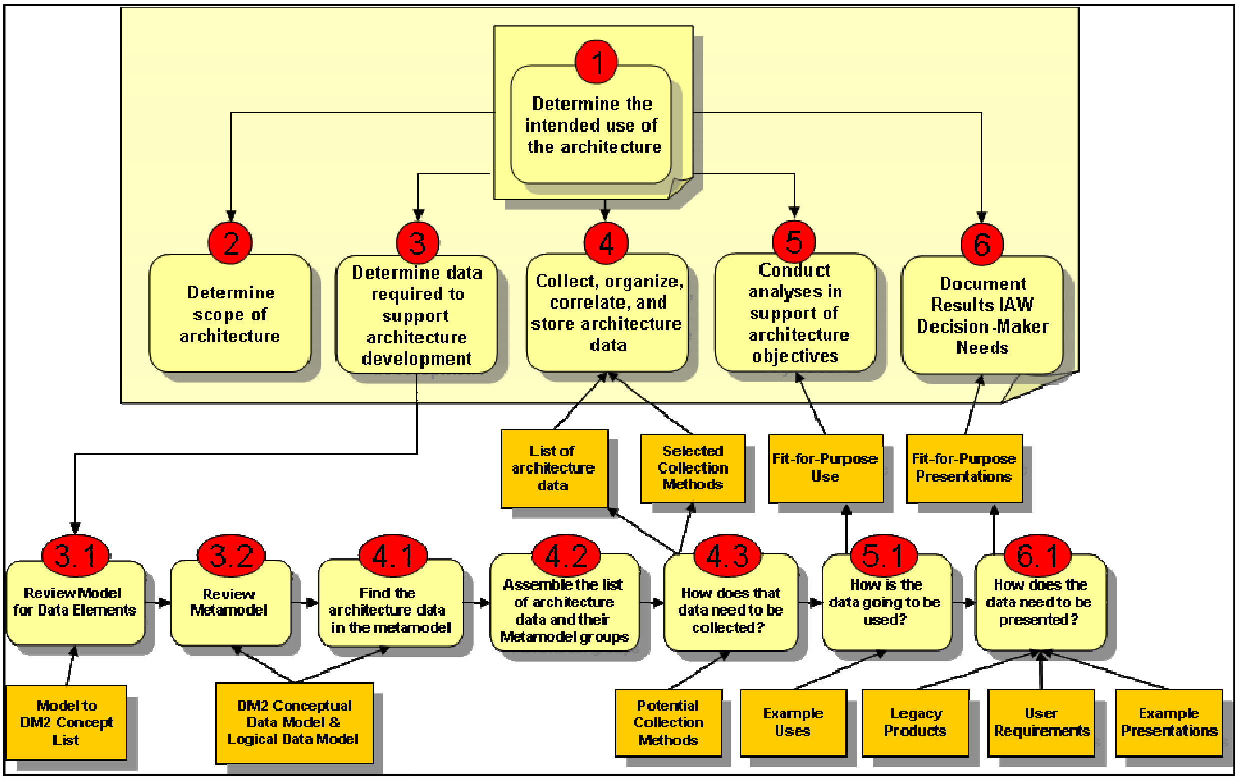


Figure 1.4.1-1: What Does the Architect Need to Do?

The architect's detailed steps, as part of the 6-Step Architecture Development Process are as follows:

- **Step 3.1**: Using Table B-1, DM2 Concepts, Associations, and Attributes Mapping to DoDAF-described Models in Appendix B, Mappings to DM2 Concepts, the architect determines the DoDAF-described Models needed, based on the concepts required to satisfy the architecture's purpose and scope (from Step 1 and 2 of the 6-Step Architecture Development Process). The architect also determines the Fit-for-Purpose Views needed, also based on the concepts required to satisfy the architecture's purpose and scope.
- **Step 3.2**: After determining the DoDAF-described Models and Fit-for-Purpose Views required, the architect reviews the:
  - DM2 Conceptual Data Model (described in Volume I, Section 8.1, The DoDAF Conceptual Data Model)
  - DM2 Logical Data Model (described in Volume II, Section 2, Meta-model Data Groups)
  - DM2 Concepts, Associations, and Attributes (described in the DoDAF Meta-model Data Dictionary and Table B-1: DM2 Concepts, Associations, and Attributes Mapping to DoDAF-described Models in Appendix B)
- **Step 4.1**: With the concepts identified in the Architectural Description's Purpose and Scope (from Step 1 and 2 of the 6-Step Architecture Development Process), the required DoDAF-described Models and Fit-for-Purpose Views, the available DM2 metadata, the architect determines the specific architecture DM2 Meta-model Groups, concepts, associations, and attributes that need to be collected for the Architecture Development Process. The tables in the Method subsections of Section 2, Meta-model Data Groups, identify the specific data.

- **Step 4.2**: The architect assembles the list of required DoDAF-described Models and Fit-for-Purpose Views, DM2 Meta-model Groups, Concepts, Associations, and Attributes. This provides the list of architectural data that needs to be collected, organized, correlated, and stored as part of Step 4 of the 6-Step Architecture Development Process.
- **Step 4.3**: Using the identified Meta-model Groups in the DM2, the architect determines the method to collect the data. With the specific list of required DoDAF-described Models, Fit-for-Purpose Views, DM2 Meta-model Groups, Concepts, Associations, and Attributes, the architect determines the appropriate collection methods for the "Fit-for-Purpose" needs. Section 2 of this document contains a Method subsection for each Meta-model group which provides potential collection methods. The results of this sub-step should guide the collection methods that will be performed in Step 4 of the 6-Step Architecture Development Process.
- **Step 5.1**: Using the identified Meta-model Groups in the DM2, the architect determines the usage of the data. With the specific list of required DoDAF-described Models, Fit-for-Purpose Views, DM2 Meta-model Groups, Concepts, Associations, and Attributes, the architect determines the appropriate usage to satisfy the identified "Fit-for-Purpose" needs. Section 2 of this document contains a Use subsection for each of the Meta-model groups which describe uses. The architect needs to determine the "Fit-for-Purpose" use of the architectural data that will meet the decision-maker's purpose and support the decision processes, including the analysis that will need to be performed in Step 5 of the 6-Step Architecture Development Process. The results of this sub step should support the analysis that will be performed in Step 5 of the 6-Step Architecture Development Process. Architectural Description analysis is key to proper use of an architecture by its stakeholders. Such analysis should be the joint responsibility of the stakeholders and the architect to ensure it answers the stakeholders' questions.
- **Step 6.1**: Using the identified Meta-model Groups in the DM2, the architect and decision-maker determines the presentations of the data.

  With the specific list of required:

  - DoDAF-described Models
  - Fit-for-Purpose Views
  - DM2 Meta-model Groups
  - Concepts, Associations, and Attributes along with the:
  - Legacy Products
  - User Requirements
  - Example Presentations

  The architect and decision-maker determines the appropriate presentations (Fit-for-Purpose Views) and data for the identified "Fit-for-Purpose" needs that will meet the decision-maker's purpose and support their decision processes.

  The results of this sub-step should support the presentations (Fit-for-Purpose Views) that will be created in Step 6 of the 6-Step Architecture Development Process. The DoDAF V2.0 Architecture Development Process for the DoDAF-described Models in the DoDAF Journal presents a non-prescriptive set of tasks to develop DoDAF-described Models in a Microsoft Project Plan.

2.3.2 Using Architectural Metadata

In addition, as the architecture is being developed, architecture metadata can be used (and updated) to support various processes and to populate architecture resources for implementation. One of the Net-Centric Data Strategy goals supported is to enable the architecture to be Discoverable, as a reusable Architecture Resource, mentioned in Section 3.5 in Volume I. Figure 1.4.2-1 illustrates the potential uses of architecture metadata for the processes they can support and the architecture resources that can be populated from the metadata captured in an architecture repository. It is important to note that architecture metadata can be used throughout the development process, not just at the end of the architecture effort.

The architecture metadata can support:

- Defense Acquisition System process with Project metadata.
- Planning, Programming, Budgeting, and Execution (PPBE) process with Cost metadata
- Information Support Plan (ISP) process with Capability metadata.
- Systems Design and Systems Engineering processes with various metadata, e.g., capability, activity, processes, systems, services, cost, project, data, and taxonomies.
- Service description, service port, and service Resource Flow metadata is used to populate a Service Registry.
- AV-2 metadata is used to create DDMS data catalog entries for authoritative sources.
- Resource Flow and Physical Schema metadata is used to populate the Metadata Registry.
- DoD Information Technology Portfolio Repository (DITPR) population with System data.

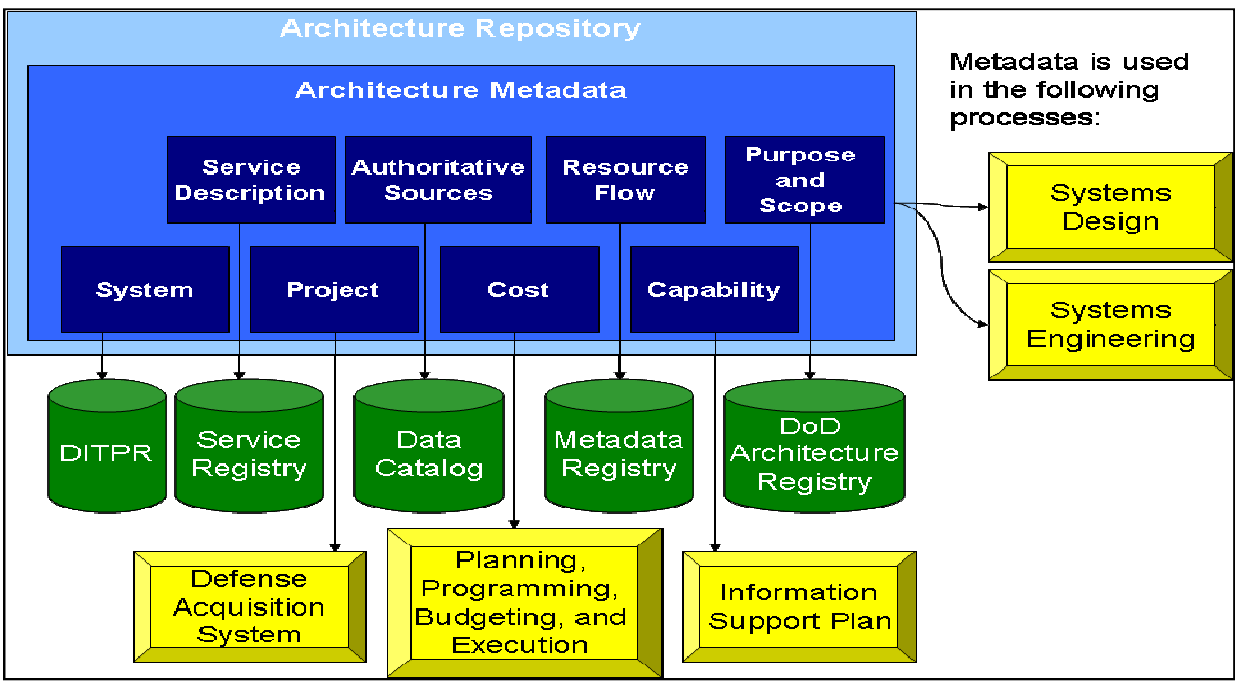


Figure 1.4.2-1: Architectural Metadata Supports Implementation

2.4 Relationship to System Engineering

There is not a separate set of system engineering DoDAF-described Models or Fit-for-Purpose Views since the entire DM2 could be used for a "Fit-for-Purpose" presentations. System engineers and system engineering decision-makers can use the existing DoDAF-described Models and create their own Fit-for-Purpose Views. If an existing model does not meet the purpose, the architect can select the appropriate data to create a "composite" Fit-for-Purpose View. In **Table 3.1.9-1**, a non-inclusive initial traceability of SE concepts to the DoDAF Meta-model Data Groups is below and can be the starting point for the "Fit-for-Purpose" presentations. Also, while not inclusive of all possible SE concepts, Table 3.1.9 is not a prescribed set of data. An example of a "Fit-for-Purpose" presentation is the System Engineering charts in chapter 4.0 of the Defense Acquisition Guide which can be rendered as Gantt or Pert Charts. Each organization and their decision-makers will need to determine their own architectural data needs. System engineering efforts could be tracked as projects and have an associated WBS and be reflected in a PV-1 and PV-2.

Table 3.1.9-1: System Engineering Concepts to DoDAF Meta-model Data Groups Mapping

System Engineering Concepts	DoDAF Meta-model Data Groups
Strategies, Scenarios, Threat, Objectives, Goals	Goals
Enterprise Priorities	Goals
Capabilities (UJTLs, Business Process Analysis [BPA] Standard processes, etc.)	Capability, Activity
Operational Performance Metrics (KPPs, etc.)	Measures
Processes/Activities	Performer, Activity
Need Lines (Connectivity)	Resource Flow
Information and Information Flow (Conceptual Data Design)	Resource Flow, Data and Information
Tactics, Techniques, and Procedures	Performer, Capability
Automation, Mechanization, Material Priorities	Goals
Strategies to Process Traceability	Goals, Performer, Activity
Operational Standards (Doctrinal, Procedural, Business Rules, etc. [Joint Chiefs of Staff {JCS} Pubs, etc.])	Rules

Table 3.1.9-1: System Engineering Concepts to DoDAF Meta-model Data Groups Mapping

System Engineering Concepts	DoDAF Meta-model Data Groups
KPP to allocated performance Traceability	Measures, Performer
Technical Standards	Rules
Process to System Function/Service Traceability	Performer, Activity
Top-level Requirement Specifications (ICD, CDD, CPD, CRD)	Capability, Services, Goals, Rules, Measures, Location, Doctrine, Training/Skill/Education, Performer, Resource Flow, Data and Information
Non-Acquisition and Acquisition WBS	Project
Cost (Training, Man Power, etc.)	Project, Measures
System Concept of Operations	Goals, Performer
System Functions	Performer, Activity
System Constraints	Rules
System Interfaces	Performer, Resource Flow, Activity
System Behavior	Performer, Activity, Rules
Trade Studies (Automation/Mechanization, Technology, commercial off the shelf [COTS], government off the shelf [GOTS], SOA, etc.) Tradeoffs	Project, Performer, Location (as in URL locations)

3 DoDAF Meta Model and Architecture Framework Relationships

3.1 Purpose of the DoDAF Meta Model (DM2)

3.1.1 Background

3.1.1.1 Core Architecture Data Model (CADM)

In 1995 the ASD(C3I) and the C4ISR Architecture Framework panel decided an architecture meta model would be a valuable component of the framework. Called the Core Architecture Data Model (CADM), it was to be a common specification of the data planned to be incorporated in architecture data repositories and databases. It would serve as the data model for the DoD architecture repository system, the Joint C4ISR Architecture and Planning System (JCAPS). The vision was that since architectures are typically developed as a set of views, merging the underlying (or implicit) data of these views into a database or other kind of data repository would enable architecture data to be maintained in a consistent way and to be reused by other architects. There were seven benefit types sought:

1. **Consistency**. The interest is both within a specific architectural description and between architectural descriptions. Within an architectural description, two types of view consistency were sought: 1) “horizontal”, across views, and 2) “vertical”, up and down levels of abstraction. A standard meta model is only one component in achieving consistency; controlled vocabularies, taxonomies, and / or common reference data in the instantiated meta model are also necessary.
2. **Re-use**. Re-use has two principal benefits:
    1) efficiency or cost savings / avoidance, and
    2) quality and consistency resulting from use authoritative and maintained data. The idea was, ‘develop once, use many.’
3. **Modeling and Simulation (M&S).** In addition to supporting the data requirements of the DoDAF, the standard architecture meta model was originally developed to support the needs of the M&S community for architecture and interoperability analyses. The goal was a standard data format for architecture data that could be ingested by models and simulations. For example, NETWARS was a GOTS/COTS tool that estimated communications throughput requirements from Information Exchange Requirements (IERs). NETWARS uses IER attributes for information element size, frequency, timeliness, security, required format, etc., along with operational node to physical node mappings to estimate bandwidth requirements at physical nodes and predict throughput bottlenecks.
4. **Methodology and Tool Agnosticism.** The framework was intended to be methodology and tool independent. The breed of architecture tools were generally methodology dependent, which often resulted in architecture data that were critical for analysis using those methodologies, but not readily aligned with the DoD architecture framework view set. A data standard aligned to the framework would enable the framework to be tool independent so architectural descriptions could be re-used across different modeling tools and methodologies. Multiple tools could be used to perform analyses. Commercial off-the-shelf software (COTS), Government off-the-shelf software (GOTS), and ad hoc reports, diagramming, executable modeling, and other modeling and simulation (M&S) tools could be interfaced to the data repository, so architecture developers and users would not be restricted to the functionality of one tool

5. **Architecture Cross Walking**. Interfaces to other architecture data repositories could be used to assess inter-organizational interoperability, gaps, or redundancy issues. Inter-organizational interoperability is one of the major reasons for employing architectural techniques. The use of a common standard meta model is a step in reducing the need for complex, costly, and sometimes infeasible reconciliations.
6. **Rapid Efficient Decision Support**. The integrated architecture data repository would become an enterprise Decision Support System (DSS). The data in a standard architecture meta model conformant repositories could be integrated, 'sliced and diced', queried and analyzed, and reports generated however needed. This capability would enable faster decision support and reduce data calls.
7. **Interfaces to Authoritative Data Source (ADS) and other Data Assets.** In many cases the ADS for architectural description information is not an architecture repository, but some other type of data asset. Examples of such ADS are the Universal Joint Task List (UJTL), DoD IT Standards Registry (DISR), DoD IT Portfolio Management Registry (DITPR), Occupation Net (O*NET), Naval Vessel Registry, and many others that are the ADS for information about organizations, occupational specialties, ships, aircraft, facilities, units, costing, and budget data. Ideally, these would be interfaced to the architecture repository rather than manually input, parsed, or imported by each architecture developer. In addition, because the DoD Data Administration policy of the time (DoDI 8320.02) required all data schemas to be standardized and managed via Functional Data Administrators (FDAd) closely aligned with the Principal Staff Assistants (PSA), the vision was also that architecture data so structured would fit seamlessly with other data assets and vice-versa.

CADM 2.0 was released with C4ISR Framework 2.0 in 1996 and was under configuration control by a Technical Working Group (TWG) and updated for DoDAF 1.0 and 1.5. MODAF followed suit but in a more UML-like manner with the MODAF Meta Model (M3) and similarly the NATO Architecture Framework (NAF).

3.1.1.2 International Defence Enterprise Architecture Specification (IDEAS)

Military operations for the future will most likely involve coalition partners. The trend towards net-centric and network enabled capability indicates that elements in an architecture are likely to be multinational to support our day-to-day requirements. To achieve interoperability requires that multiple nations and other organizations share key information elements across National Defense and other key allied organizations. In 2004, the UK, Canada, Australia, and the US defense departments discussed the need to exchange architecture data in anticipation of missions involving coalition forces. To support the requirement for exchange of critical data, an architectural exchange specification is needed to permit coalition partners to develop national, coalition, and joint enterprise architectures. The objective was to detect possible interoperability and / or capability gap or overlap problems early-on before mission commencement so that plans could be adjusted or the problems fixed while in garrison or reroute. Types of interoperability problems included 1) Doctrine mismatch, e.g., Tactics, techniques, and procedures (TTP), 2) Training and skills mismatch, 3) Systems mismatch – communications, processing, and / or data formats. The IDEAS Group believed that exchanging architecture data during coalition operations planning process:

1. Can automate interoperability comparisons to:
2. Reduce resource requirements
3. Speed the process
4. Potentially detect issues that may have been missed

5. De-bias national interpretations of other doctrines

The “as-is” and “to-be” for coalition operations architecture data exchange is shown in TBS. The “as-is” requires 3 mental data parses per country and 3 mental comparison per country, all biased by national background, as compared to the “to-be” which required only 1 mental data parses against an consistent ontology, not a national background.

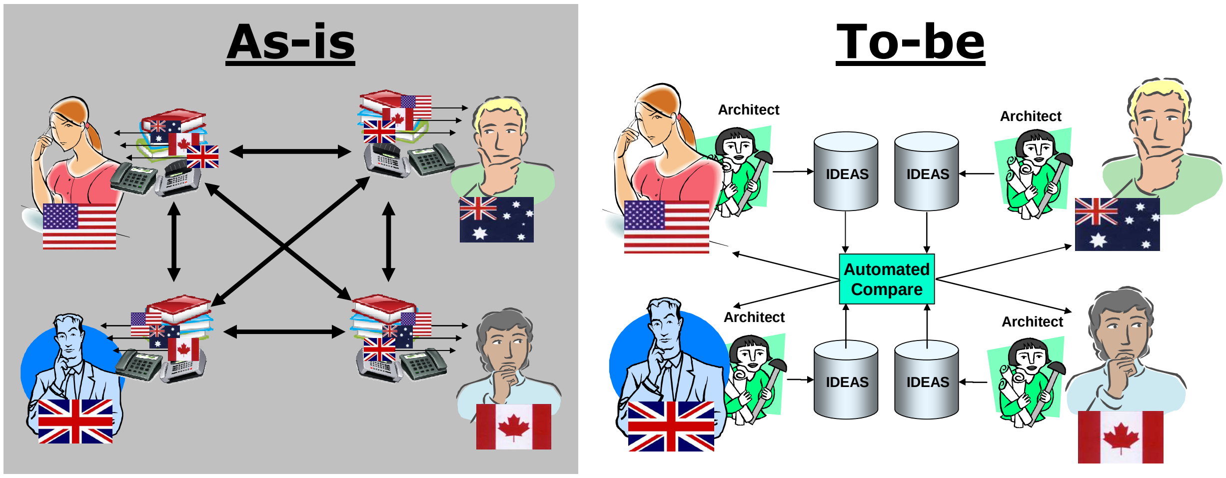


Figure 1. Coalition Architecture Exchange As-is and To-be

A use case with which the IDEAS Group experimented was for doctrine or procedural issues for casualty management, using the SCUD attack on the US barracks during Operation Desert Storm.

Early on in the project, called, the member nations realized an interoperable exchange specification could not be reached using conventional data modeling techniques because of their need for mutual consensus on the meaning of a large number of terms and their inter-relationships. The alternative chose was a formal ontology based on universally-agreed-upon mathematical concepts such as set theory and topology. The ontology itself is concerned with the nature of things and relies on the only thing that is irrefutable, the physical extent of something; it provided a language-independent way for the nations to develop the model as illustrated in Figure 2. It was important to ignore names when developing the ontology, as they carry too much baggage and confusion – people tend to cling onto names of things rather than trying to work out if things are the same or not. It was agreed that once the semantic de-confliction is done, the names could be re-assigned individually by the nations, in their own dialects, in context of their owners – and this is how interoperability was to be achieved. Figure 3 illustrates the concept of IDEAS commonality and nation-specific terminology and extensions for the MOD.

Figure 2. IDEAS Language Independence

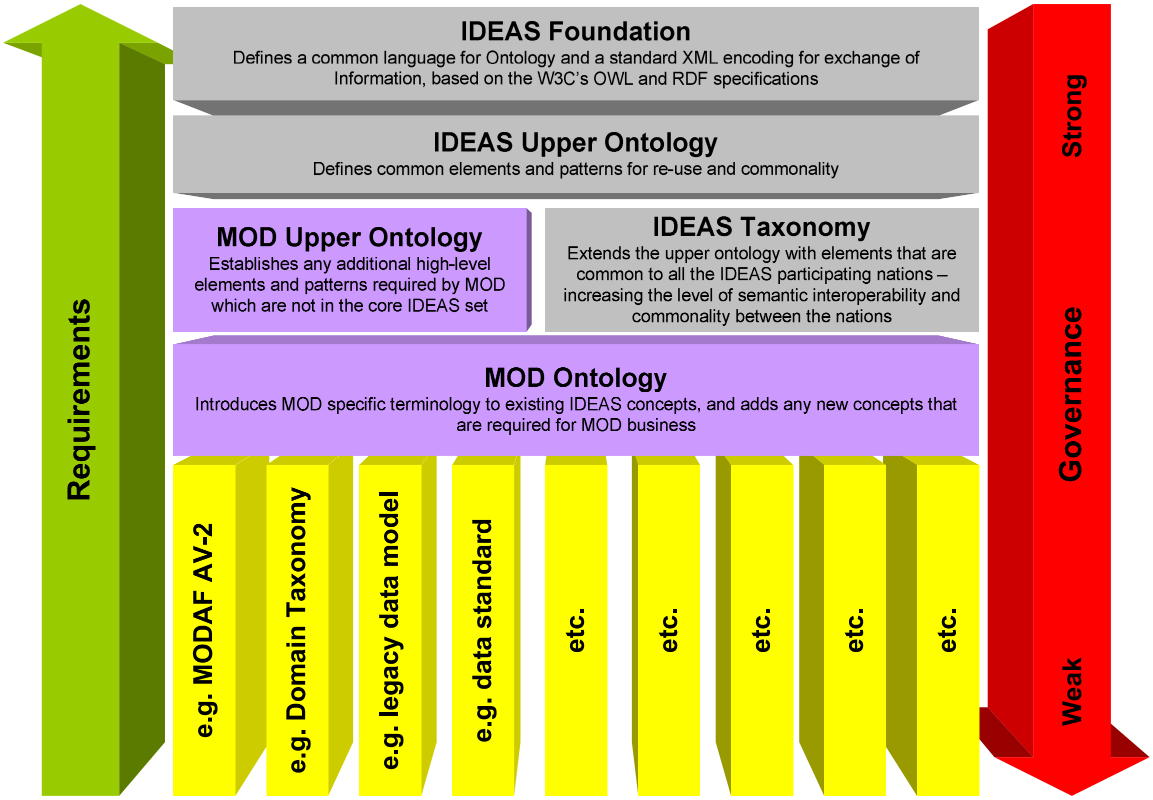


Figure 3. Concept for National Naming and Extensions to IDEAS

3.1.2 DoDAF 2 Meta Model (DM2) Development Drivers

There were two main drivers for the development of the DM2 as part of DoDAF 2.0 development:

1. Lessons-learned from prior frameworks
2. The requirement for DoDAF 2.0 to be responsive to DoD's six core processes.

Each of these is explained in the following subparagraphs.

3.1.3 Lessons-learned from prior frameworks

While the DoD CIO remained committed to seeking the seven benefits of CADM described in paragraph **Error! Reference source not found.**, there were some lessons-learned from 15 years of implementing CADM in repositories, architecture development or authoring tools, and analysis and M&S tools, summarized in the following subparagraphs.

3.1.4 Lesson on Dissociated Framework and Meta Model Groups:

In prior frameworks, the framework was developed by one group, the CADM another. This led to inconsistencies between the two. Many definitions of terms (e.g., "Node") were different in CADM than in the framework. In some cases there were several substantively different definitions of the same term. This was confusing to users. In addition, it resulted in the CADM not exactly matching the framework's models. A side effect was also that, over time, the CADM TWG became more focused on database management issues and less on the representation of architectural descriptions.

DoDAF 2.0 remedies:

1. One working group, not two separate ones. This keeps data modelers focused on the requirements of DoD's six core processes DoDAF is intended to support -- 1) JCIDS, 2) DAS, 3) PPBE, 4) CPM, 5) Systems Engineering (SE), and 6) Operations Planning (OPS) -- and architectural description support thereof. On the other hand, keeping the architects involved with the precision semantics of the data modelers improves the precision with which architects describe models
2. Single definition of terms for both the DoDAF models and the DM2.

3.1.5 Lesson on CADM's Data Modeling Style:

As mentioned previously, the CADM adhered to the DoD data administration policy in force at the time. That led to data structures that were not optimal for architectural description representation since many were re-used from other functional domains. The FDAd's for those domains had competing demands for data structures from all the different users and they had to strike a balance. Often that balance was suboptimal for some users. A consequence is that many of them turned out to be semantically equivalent and repository developers were often surprised to find how other developers had implemented CADM and dismayed that their data could not be exchanged. This result defeated many of the CADM's goals described in paragraph **Error! Reference source not found.**

In addition, even though the first letter in "CADM" is "Core", over time many data elements were requested and granted that arguably were not "core". An example was the Antenna table with many details about the microwave and physical characteristics of antennae. This was requested by a user attempting to use DoDAF for trade-off analysis of SATCOM. Another was the serial-number detail on materiel, requested by a Combatant Command attempting to use JCAPS for asset management. Neither of these could be argued to be common core elements across the DoD.

CADM ended up being very big -- 16,000 data elements -- and very complex, due to size, poor fit, and the semantic redundancy. The complexity made CADM conformance at-best challenging for architecture tool vendors and repository developers. The DoD data administration policies of that era were disestablished in 2002.

DoDAF 2.0 remedies:

1. Develop the DM2 based on the representation needs for the six core processes.
2. Scope strictly to the requirements for architectural description representation.
3. Ensure the same information cannot be represented multiple ways.
4. Develop the DM2 with multiple levels of access for developers, from simple for basic use to as complex as needed for those in need of that level of fidelity.

3.1.6 Requirement for DoDAF 2.0 to be responsive to DoD's six core processes

A major motivation for DoDAF 2.0 development was to focus on architectural description support for DoD's six core processes: 1) JCIDS, 2) DAS, 3) PPBE, 4) CPM, 5) Systems Engineering (SE), and 6) Operations Planning (OPS). The reasons were that prior frameworks were not focused on these processes specifically enough and that these processes had changed or were new (CPM) since DoDAF 1.0. (DoDAF 1.5 was a relatively minor update to accommodate net-centricity and SoA.) Examples of areas of DM2 that are wholly new or different from CADM are:

1. Capability model. In CADM, the entity "Capability" was actually just a numerical entity. DM2's Capability model very precisely matches DoD's definition of Capability.

2. Services model.
3. Measures and metrics.
4. DOTMLPF

The requirement for DoDAF 2.0 to be responsive to the six core processes implied more than just that there be adequate architectural descriptions, it also required that such descriptions needed to fulfill their roles in the core processes. A typical pattern for architectural description usage is shown in Figure 4.

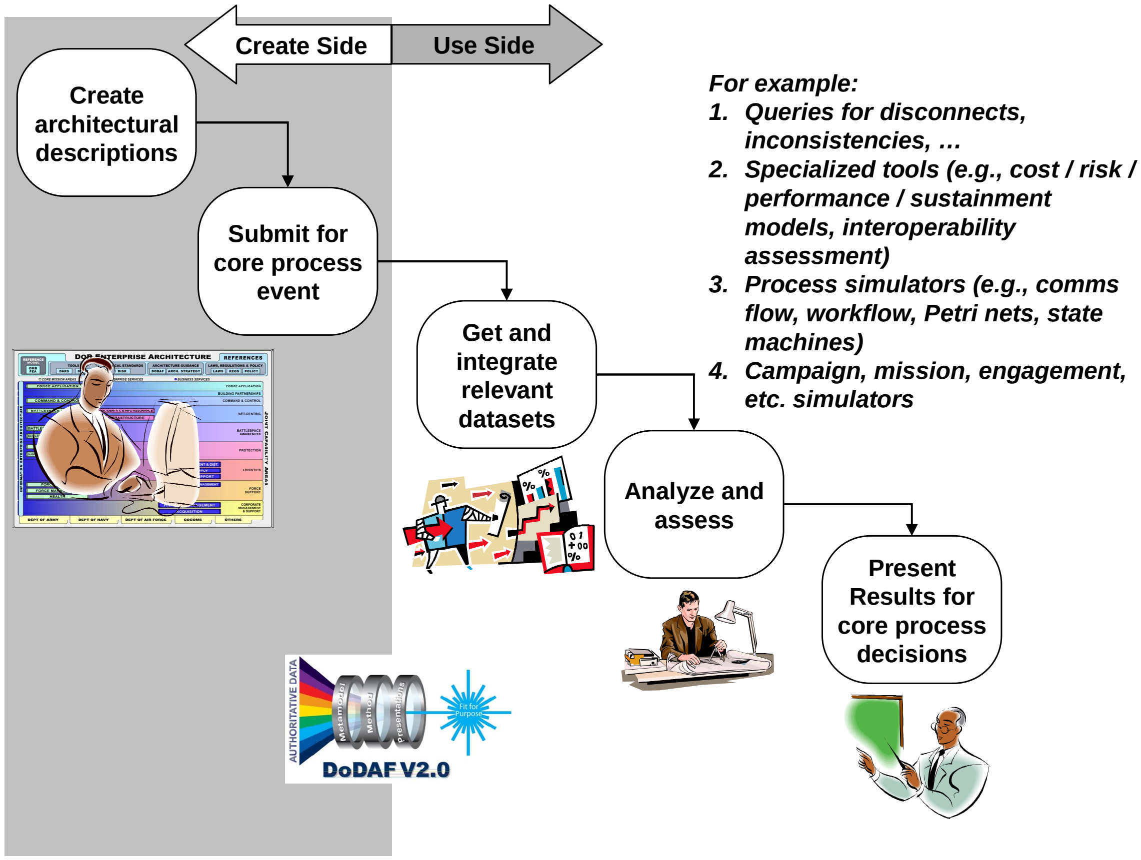


Figure 4. A Typical Pattern of Architecture Data Usage

The use of architecture data in conjunction with M&S, performance analysis, and assessment tools is an area of expanding interest because of its importance for capabilities-based assessments and analysis of alternatives. The potential value to an enterprise of a proposed architecture may not be obvious. Measures of merit can include cost; performance; interoperability; satisfaction of requirements; manpower and training; logistics, deployment, and asset allocation; schedule, and many others. The formulae for computing measures of merit may be quite complicated, as in a complex M&S program. An important ingredient in these measures is quality input data. Consequently, an implication of this pattern of core process architectural description use is that the data must, 1) integratable, and 2) of high quality. Data quality affects the ability to analyze architecture models and the ability to compare or integrate independently developed architectures. Architecture data quality can be characterized

1. One is conformance with established structural and semantic specifications (i.e., the definitions of fundamental data entity types or object classes and their attribute data type specifications).

2. Another aspect is conformance with preferred or mandated entity or object instance values (referred to here as reference data) established by recognized authorities, or authoritative sources. An authoritative source is a designated or recognized authority for specifying the acceptable or allowable data instance values (e.g., domain values) and their taxonomies. A reference data set refers to a set of element values that are approved or designated for use by a recognized authoritative source. DoD architects should use reference data from recognized authoritative sources wherever possible. The use of authoritative reference data in architectures eliminates ambiguity, provides consistency, and facilitates analysis and integration.

When architecture data elements are combined to form an architectural description, another aspect of data quality becomes important – that is, the degree to which an architecture model accurately represents an existing "as-is" architecture, or the proper association of components in a notional "to-be" architecture. This aspect of data quality is dependent on the knowledge of the architecture team about the capability domain being modeled and the reliability of the architects in accurately representing facts about the domain. This aspect of architecture data quality is difficult to measure, but can be controlled through subject matter expert (SME) review and architect training.

Data quality is ultimately dependent on the intended use. Intended use may vary from communicating general information about a mission scenario to providing a system engineering requirements baseline to providing inputs to a high-fidelity simulator.

3.1.7 Genesis of DM2

Based upon the lessons-learned and new requirements for DoDAF 2.0, DM2 was developed. In addition to the purposes previously established for CADM, DM2's purposes were to:

1. Provide the vocabulary for description and discourse about DoDAF models and views and their core process usage.
2. Provide the basis for generation of the "physical" exchange specification for exchange of data between architecture tools and databases.
3. Provide a basis for semantic precision in architectural descriptions to support heterogeneous architectural description integration and analysis in support of core process decision making.
4. Support information sharing across the DoD Enterprise Architecture COI with precise, universally understood, and commonly interpretable semantics.

DM2 development was begun by a TWG of voluntary members of the DoD architecture community and led by the DoD CIO development team. Following convention, three phases were planned: 1) Conceptual, 2) Logical, and 3) Physical as described in the following subparagraphs.

3.1.8 DM2 Conceptual Data Model

TWG members nominated many existing data models to be the DM2 including those listed in Table 3.1-1:. But the TWG was continuously advised to focus on the information requirements of the six core processes rather than immediately adopt an existing model. This led to the CDM being merely a data dictionary but one that had some additional features including,

1. Source Definitions. All the source definitions used to derive the CDM definition are part of the data dictionary. Some terms have ten source definitions. Sources included those listed in Table 3.1-2: .
2. Rationale. How the CDM definition was derived.
3. Researcher and Notes. Every term was assigned to research teams.
4. Aliases and Composite Terms. Because the TWG did not want to repeat the problems of CADM where terms were admitted to the diagram to satisfy various communities even if semantically equivalent ones could have been identified, an alias section of the data dictionary was setup. The aliases have the same structure as the main terms (sources, formulated definition, etc.) along with a mapping to the main terms. Because many terms are not simple one-to-one mappings, there is often a composite of terms that is used.

Table 3.1-1: Data Models Referenced During DM2 CDM Development

CADM 1.5	IDEAS
UPDM	BMM
Hay/Zachman	ASM
CRIS	Conceptual CADM in DoDAF 1.0 / prototype CADM 2.0
M3	NAF Meta Model
DoI Meta Model	JC3IEDM
GML	UCORE 1.1
GEIA 927	AP233
SUMO and ISO 15926 (via IDEAS)	FEA Reference Models
JFCOM JACAE	

Table 3.1-2: Sources Used for Definitions

IEEE	ISO
W3C	OMG
EIA	DODD & DODI
JCS Pubs, especially CJCSI's	Models in the Source_Candidates_071115.ppt
DoDAF 1.5	Other frameworks: Zachman, MODAF, TOGAF, NAF
FEA	BMM
Worknet	Wikipedia
English dictionaries	CADM

3.1.9 DM2 Logical Data Model (LDM)

As the logical design phase began, it became apparent that there were many repeating patterns:

1. The need to describe the parts of something or, conversely, to describe what something is parts of.
2. The need to categorize things, to say what type something is. That implied the need to describe subcategories or subtypes.
3. The need to describe consumption and production of resources by things.
4. The need to describe interactions amongst things.
5. The need to describe sequences of things, activities, processes.

6. The need to describe temporal states of things and transitions from one state to another
7. The need to describe where things are and when
8. The need to describe goals, objectives, desired effects, etc. and the means to achieve them

Several TWG members who were familiar with IDEAS recognized that IDEAS had abstracted these patterns as part of its formal ontology foundation so it was requested that the development team investigate use of IDEAS to capture these patterns in a consistent and non-redundant manner. The findings were that IDEAS was a very good fit and it was adopted as the DM2 foundation. There were many downstream benefits to adopting IDEAS that were not foreseen clearly at the time including:

1. Design economy. The final DM2 design had ~250 elements as compared to CADM's 16,000
2. Mathematical Basis for Analysis. Since every relationship in a DM2 dataset must have a mathematical meaning, they is much more suitable for analysis than datasets that require manual interpretation and possibly assumptions.
3. Ontologic Support for Heterogenous Data Integration. The very nature of DoD architectures results in the need to integrate independently-produced architectural description datasets. That is one of the benefits of formal ontologies.

Along with IDEAS, the TWG also adopted the IDEAS methodology or "BORO" methodology. The BORO methodology provides a way to mathematically analyze concepts so that they can be clearly and consistently understood and agreed upon. The BORO methodology is still used by the DoDAF-DM2 Working Group to this day to reach unanimous consent on model structure. (Whenever there is a disagreement, the parties must produce their analysis and eventually one can be proven correct.) A BORO decision tree for analyzing a concept at the top-level of IDEAS is shown in Figure 5. There are several other important "business rules" the TWG agreed to follow and those have matured into the DoDAF-DM2 WG rules shown in Table 3.1-3.

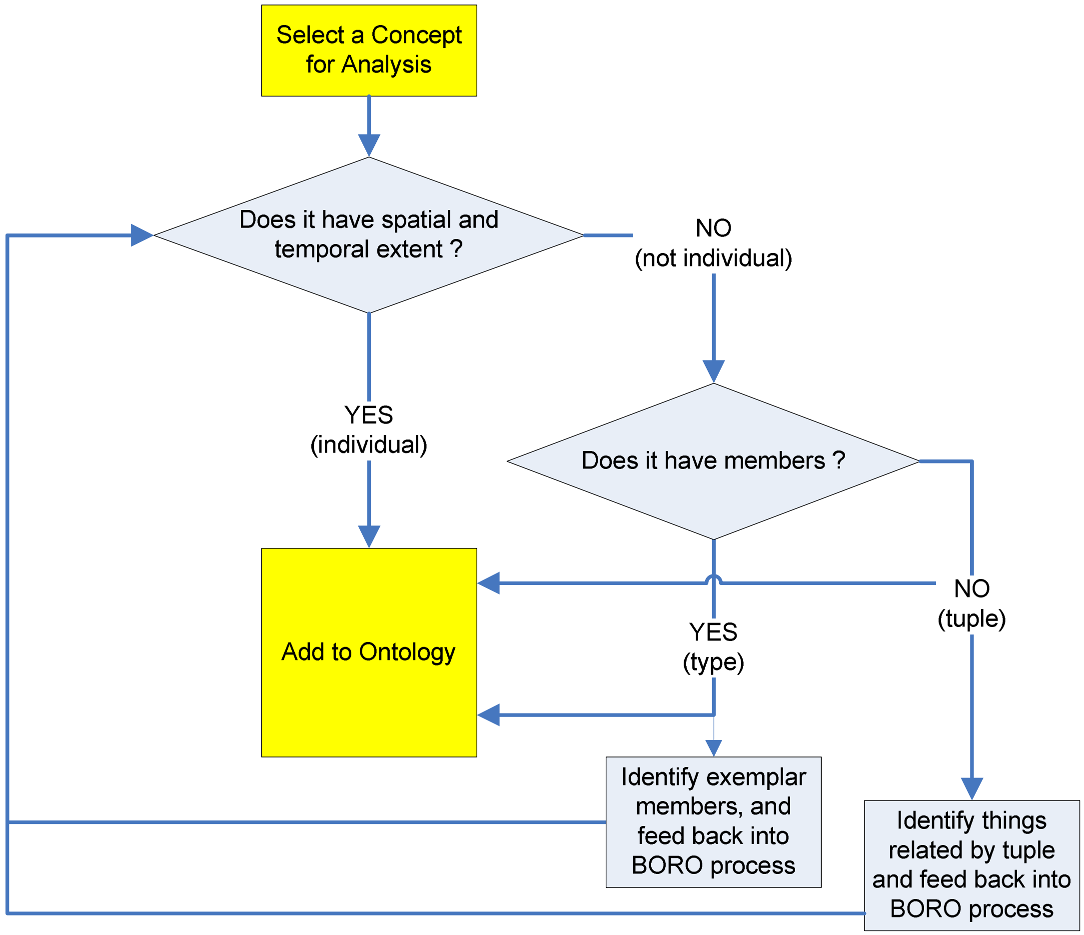


Figure 5. Top-Level BORO Analysis Decision Tree

Table 3.1-3. DM2 Model Specification Rules

Rule Name	Description
Terms and Definitions	All model and alias terms proposed for inclusion in the data dictionary shall be researched for multiple source definitions. DoD definitions shall be included. Other Federal Government, industry and academic and common definitions should also be included. The WG shall formulate a baseline definition based on the multiple sources, core process requirements, and model structural meaning. The source definitions and the rationale for the baseline definition shall be maintained in the data dictionary as well.
Aliases	Terms representing concepts that are represented in a semantically equivalent way by other terms and concepts in the model shall be maintained as aliases and shall not be introduced into the model. Multiple source definitions shall be maintained as with other model terms and a consensus definition shall be derived from the sources.
Core Process Requirement	All concepts included in the DM2 shall be necessary to support the information requirements of one or more core processes (PPBE, DAS, JCIDS, CPM, SE, OPS). All DoDAF models shall be applicable to one or more core processes. Core process information requirements shall be as explicitly or implicitly specified in current or planned DoD governance. All model terms and concepts not necessary for core process support with architectures shall be removed. All core process information requirements for architectural descriptions shall be modeled and contained in one or more DoDAF models.
Aggregation Rule	If a term representing a concept differs structurally from some other term representing some concept only in level of aggregation, it shall not be included in the model. Whole-part relationships cover the need without different names for different types of wholes and parts. The term may be included as an alias.
Subtype Rule	If a term representing a subtype concept has no structural difference from its supertype, it shall not be included in the model. Super-subtype relationships cover the need without different names for different types of supertypes and subtypes. The term may be included as an alias.
Typed Relationships	All relationships shall be typed, ultimately up to IDEAS foundation. The typing shall be determined using BORO analysis of spatio-temporal examples.
Attributes and Properties	All attribute and property relationships shall be explicit, that is, by an association class that is typed according to the Typed Relationships rule. The only exceptions are for representational exemplars.
Information Pedigree	There shall be a provision to provide pedigree (and provenance) for every piece of data IAW NCDS
Security classification marking	There shall be a provision to provide a classification marking for every piece of data and for DM2 PES XML documents overall IAW NCDS

3.1.10 DM2 Physical Exchange Specification (PES)

Unlike the CADM, the physical phase was to produce an exchange specification only, not a schema for RDBMS implementation. The reason was to provide greater access to non-RDBMS implementations (e.g., object-oriented, tools) and for the DoDAF to prescribe the minimum needed for DoD goals, i.e., to not over-prescribe. A very simple XML style was chosen that somewhat resembles the UCORE "digest" style in that it is very flat, i.e., does not use much of the structural features of XSD. The reason is that the IDEAS relationships are all explicit so there is no need for any additional ones other than for packaging. It was understood that this PES would not be structured for immediate analysis but the TWG decided that the accessibility would be a good trade-off. The PES does not preclude analysis, but merely requires that additional artifacts (the LDM) be accessed to analyze the datasets. An IDEAS-based RDFS-

OWL structure was foreseen but was not released with DoDAF 2.0 because it was judged to not have the wide accessibility required for the DoD architecture community.

3.1.11 Summary

The need for and potential benefits of an architectural description meta model have been recognized and sought after for over 15 years. Many lessons have been learned along the way but the original vision is largely intact. Indeed, new requirements for rigor and data quality have emerged. The need for coalition architectural description exchange and DoDAF 2.0's commitment to support DoD's six core processes have necessitated fresh and flexible approaches. The current meta model (DM2) is the latest in the evolution, providing accessibility at the conceptual as well as multiple physical levels – XML, RDBMS, and RDFS/OWL – and a mathematical foundation that holds promise in achieving DoD's goals for architectural descriptions.

3.2 DoDAF Glossary and Model Files

3.2.1 DoDAF glossary and model files

a. The DM2 LDM description provides the essential aspects of the standard terminology used as the basis of DoDAF 2.0. The DM2 provides the standard data lexicon definitions and the logical relationships between elements of the lexicon. The DM2 defines the common architectural description lexicon across the six major processes of the DoD. That terminology and its mapping to other widely-used terms are contained in the DoDAF Glossary. The DoDAF Glossary is maintained in Microsoft Excel and has the following structure.

Table 3.2-1: DM2 Data Dictionary Columns

Columns in DM2 Data Dictionary											
B	C	D	E	F	G	H	I	J	K	L-AB	AC-CB
Name	Definition	Aliases	Status	Source Definitions	Definition Rationale, Further work, other Comments	Examples	Association?	CDM?	Who	DM2 Data Group Applicability	DoDAF Legacy Model Applicability ("Monster Matrix")
Name of concept or relationship	*Agreed-upon definition*	*Aliases are kept in the data dictionary*	*In model, alias, external, new*	*Multiple sources maintained*	*Comments*		*Is this a relationship?*	*Is this in the Conceptual Data Model*	*Primary investigator*	*Which of the 12 DM2 submodels does this apply?*	*Which of the 52 AV-1 through DIV-3 does this apply?*

b. It is best used using:

1. Microsoft Excel data filters to see only the items of interest. This is particularly useful when examining the “monster matrix”, by filtering to the DM2 elements that are necessary or optional in a view.
2. Microsoft Excel “freeze panes” to view columns far to the right
3. Row and/or column grouping (some are already included) or hiding to see the information of interest. For instance, you may interested in the “monster matrix” but not the definitions, sources, etc.

c. The detailed model description including the detailed definitions, relationships and the lexicon mapping to the DoDAF 2.0 views (models) are available as an Enterprise Architect (SPARX) file that can be viewed using a licensed copy of Enterprise Architect or a free viewer only. Since the DM2 is based on IDEAS, not UML, to see the diagrams correctly, an IDEAS profile should be installed.

1. To download the DM2 EA file, click here.
2. To navigate to the SPARX EA-lite site, click here.
3. To navigate to the IDEAS Group site to download the IDEAS profile, click here.

3.3 DM2 and Core Process Relationships Overview

An overview of the role of the concepts modeled in the DM2 is shown in Table 3.3-1. The key to the symbols in this table are:

Table 3.3-1: Mapping of DM2 CDM Core Concepts to DoD Core Processes DoDAF Supports

DM2 CDM Core Concepts	Core Process Utilization						
	JCIDS Capability Mgmt	JCIDS Interop & Supp	DAS	PPBE	CPM	SE / SOSE	Ops Planning
Activity	●	⊙	◉	◉	⭘	◉	⭘
Agreement	⊙	◉	◉	⍰	⍰	◉	◉
Capability	●	◉	⭘	◉	●	◉	◉
Condition	⭘	⊙	○	⊙	⊙	⊙	⭘
Data	◉	⭘	⊙	○	◉	◉	⊙
DesiredEffect	●	○	◉	⍰	⭘	⊙	●
Guidance	○	◉	◉	⊙	⊙	◉	⭘

Table 3.3-1: Mapping of DM2 CDM Core Concepts to DoD Core Processes DoDAF Supports

DM2 CDM Core Concepts	Core Process Utilization						
	JCIDS Capability Mgmt	JCIDS Interop & Supp	DAS	PPBE	CPM	SE / SOSE	Ops Planning
Information	⭘	⭘	⊙	○	◉	◉	⭘
Location	⊙	⊙	○	⊙	⊙	○	⭘
Materiel	◉	○	◉	⊙	⊙	⭘	⭘
Measure	●	⭘	⭘	⊙	⭘	◉	⊙
MeasureType	●	⭘	⭘	⊙	⭘	◉	⊙
Organization	⊙	○	○	⊙	⊙	○	⭘
Performer	⭘	⭘	⭘	⭘	⭘	⭘	⭘
PersonType	◉	⊙	⊙	○	⊙	⊙	⭘
Project	⍰	⍰	⭘	⊙	◉	⊙	◉
Resource	◉	◉	⊙	○	◉	◉	⭘
Rule	⊙	⭘	◉	○	⊙	⭘	⊙
Service	⊙	◉	◉	⊙	◉	⭘	⍰
Skill	◉	⊙	⊙	○	⊙	⊙	⭘
Standard	⊙	●	◉	○	⊙	⭘	⊙
System	⊙	⭘	⭘	⭘	⭘	●	◉
Vision	⊙	⍰	⊙	○	◉	⍰	○
ArchitecturalDescription	⭘	⭘	◉	⊙	⭘	⭘	◉

LEGEND:

●	Critical role
⭘	Substantial role
⭘	Significant role

◉	Moderate role
⊙	Supporting role
○	Minor / optional role
blank	Insignificant / no role

3.4 Reifications in Architecture and DM2

Architecture descriptions such as activity models are example of architectural descriptions that reified at many levels of abstraction. In a typical development project, the architectural descriptions (contained in plans, specifications, and model-based computer-aided designs) provide increasing levels of detail as the project progresses through the normal DoD milestone process. This is what John Zachman calls "reification, the transformation of an abstract idea into an instantiation." In a related paper, Zachman goes on to say "What differentiates the Rows of the [Zachman] Framework is not levels of detail… the models in the different Rows are different models. They are the result of transformations, not decomposition." as shown in Figure 3.2.1-1.

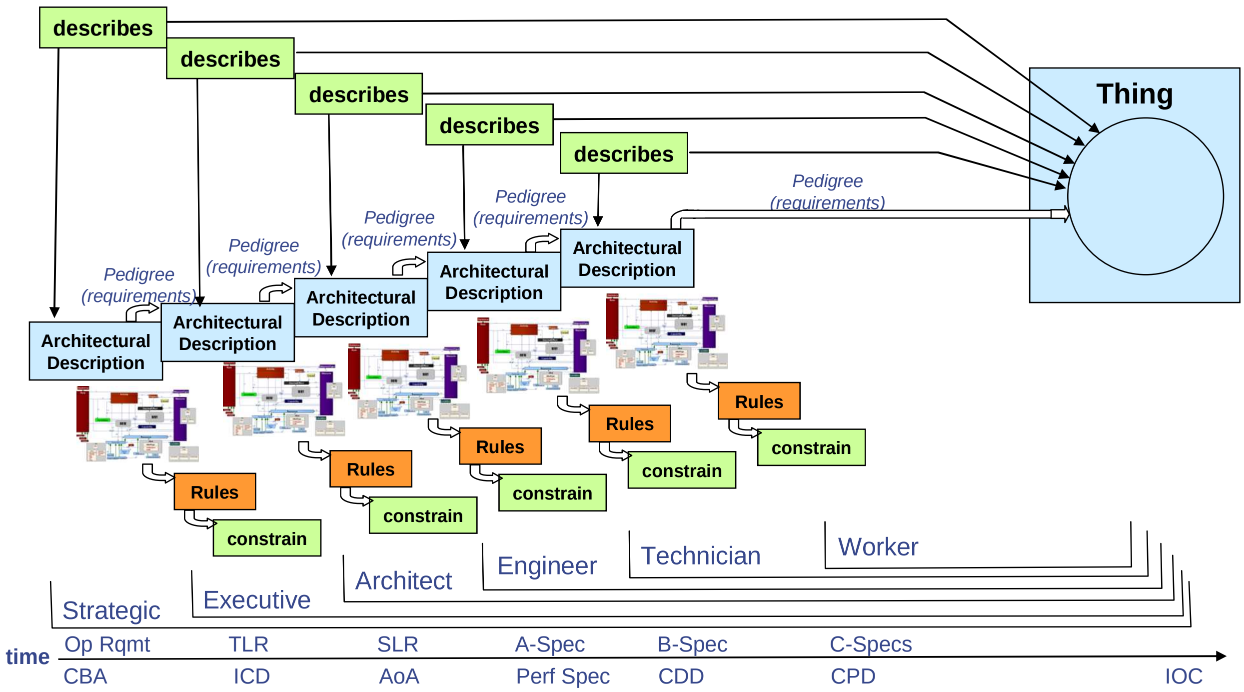


Figure 3.2.1-1: Reification of architectural descriptions at different levels

3.4.1 Data group description

Figure 3.4.1-1 shows the DoDAF meta-model diagram for the Reifications data group.

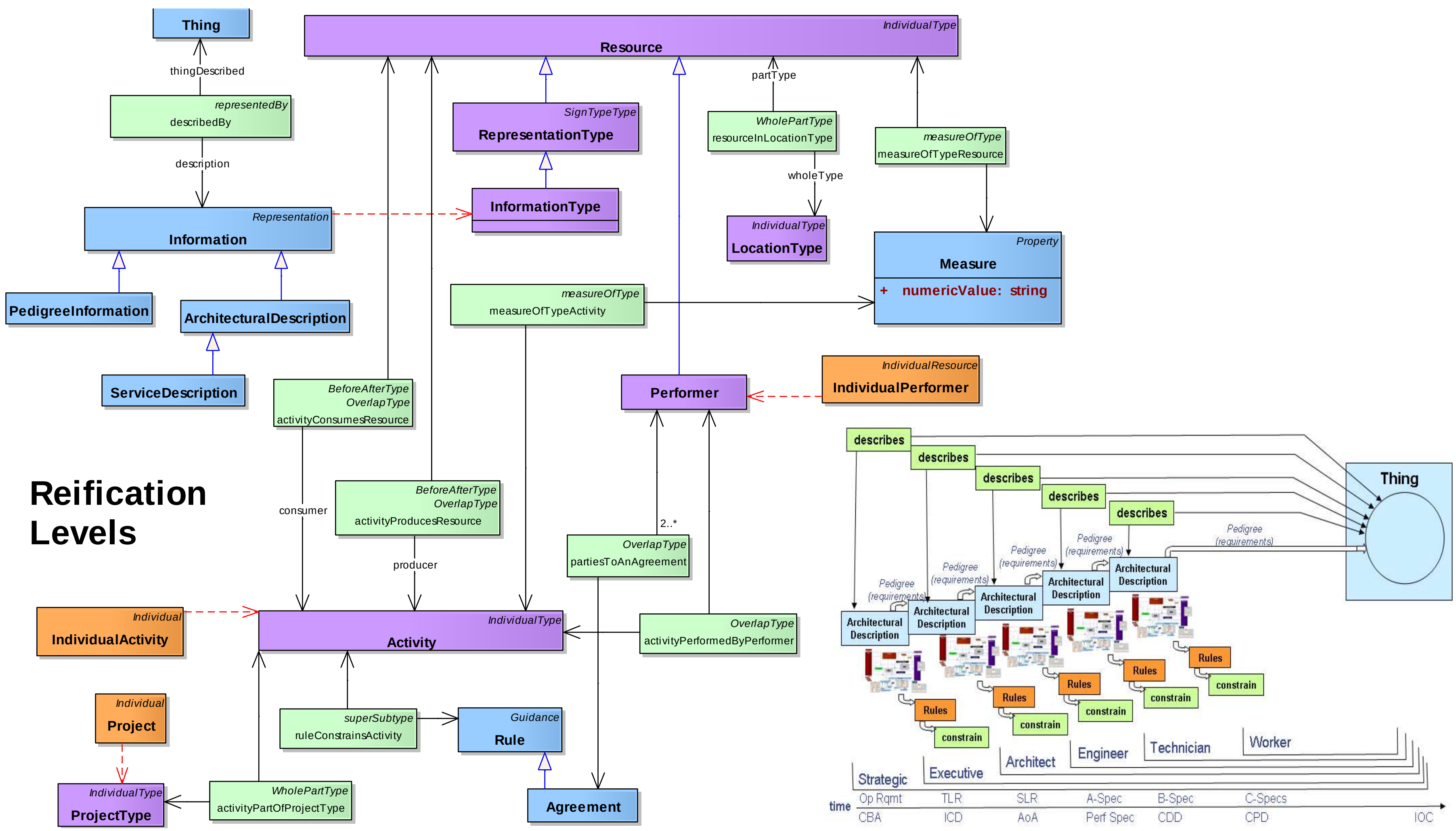


Figure 3.4.1-1: DoDAF meta-model diagram for Reifications data group

3.4.2 Use in DoD core processes

Reification may be used in these ways:

4. JCIDS:
    1. Refinement and increased levels of detail of capability and solution constraint descriptions from ICD to CPD.
5. PPBE:
    1. Refinement in project or program WBSs and cost-to-complete estimates.
6. DAS:
    1. Refinement and increasing detail of design and architectural descriptions through the milestone review process.
7. Systems engineering:
    1. Refinement and increase detail of design and architectural descriptions through the various design and development stages.
    2. Clearly described functional allocations and traceability throughout the various levels of architectural descriptions (e.g. specifications, architectural view and models).
8. Operations planning:
    1. Refinement and increasing levels of detail in tactics, techniques and procedures throughout the stages of operational plan development.
9. Capability portfolio management:
    1. Refinement and increased detail in the descriptions of the capability, performance, functionality and cost effectiveness of the portfolio.

Zachman, John. *The Zachman Framework for Enterprise Architecture*. Zachman International. 2011.

Zachman, John. *Concise Definition of the Zachman Framework*. Zachman International. 2008.

Zachman, John, *Yes, "Enterprise Architecture Is Relative" BUT It Is Not Arbitrary*. Zachman International. 2009.

4 Presentation and Visualization supporting Fit for Purpose

4.1 Presentation (Fit-for-Purpose Views) and Documents

Effective presentation of business information is necessary for architects to convey the data in the Architectural Description in a way meaningful to stakeholders. Since the purpose of the enterprise architecture discipline is to collect and store all relevant information about an enterprise, it can be assumed that the majority of information needed by an organization's decision-makers is contained somewhere in the architectural data. Presentations, or Fit-for-Purpose Views, are always dependent on the quality of the architecture information collected through the rigor of architecture methods. Many of the existing architecture methods, or DoDAF-described Models, are valuable for organizing architecture information, but less valuable for communicating that information to stakeholders. As **Figure 1.2-1** illustrates, presentation techniques pull from the architecture information store and display the data to stakeholders.

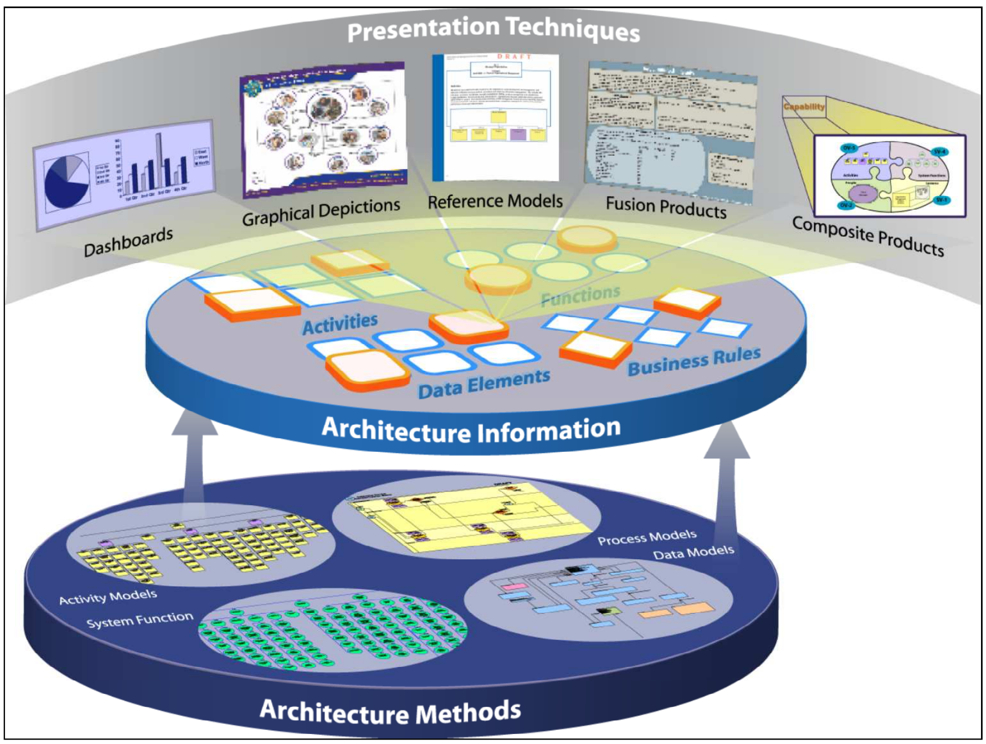


Figure 1.2-1: Relationship of Architecture Methods, Data, and Presentation Techniques

Presentation techniques allow for the communication of many complex or disparate concepts in a context that is meaningful and useful for viewers. Displaying complex information in an effective way can be difficult, but enables the communication and analysis of information. If designed well, a single presentation, a Fit-for-Purpose View, can replace 20 individual documents and display the information with purpose, geared to the targeted stakeholder. This knowledge visualization is accomplished through the use of various techniques, which are each described below.

Information is generally presented in textual documents, with associated, imbedded graphical representations. Specific presentation types are educational syllabi; instruction modules; dashboards on accomplishments or status; and graphical charts, such as pie charts, or bar charts.

It is imperative to realize that when choosing how to present data sets, there is no limit on which presentations (Fit-for-Purpose Views) to use. There are countless ways to display information to decision-makers, and it is up to the presentation developer to determine the most effective way to accomplish this task. The remainder of this document will give a base of presentations to start from, each created to serve its own unique purpose. Details are provided on five different presentation techniques that have proven to be useful in engaging various audiences, and a more comprehensive treatment of presentations will be found online in the DoDAF Journal. The five techniques are as follows:

a. **Composite Presentations**: Display multiple pieces of architecture in formats that are relevant to a specific decision-maker.

b. **Dashboards**: Integrate abstracted architecture information for a given business context.

c. **Fusion Presentations**: Display multiple pieces of architecture and incorporate disparate pieces of information that are not captured within the architecture.

d. **Graphics**: Visually represent manipulated data.

e. **Reference Models**: Capture the elements of the architecture and translate those elements into text.

The DoDAF-described Models that are available in DoDAF V2.0 are listed in **Table 1.2-1**. The list provides the possible models and is not prescriptive. The decision-maker and process owners will determine the DoDAF-described Models that are required for their purposes. The DoDAF-described Models are grouped into the following viewpoints:

- All Viewpoint (AV)
- Capability Viewpoint (CV)
- Data and Information Viewpoint (DIV)
- Operational Viewpoint (OV)
- Project Viewpoint (PV)
- Services Viewpoint (SvcV)
- Standard Viewpoint (StdV)
- Systems Viewpoint (SV)

Table 4.1-1.2-1: DoDAF V2.0 Models

Models	Descriptions
AV-1: Overview and Summary Information	Describes a Project's Visions, Goals, Objectives, Plans, Activities, Events, Conditions, Measures, Effects (Outcomes), and produced objects.

Table 4.1-1.2-1: DoDAF V2.0 Models

Models	Descriptions
AV-2: Integrated Dictionary	An architectural data repository with definitions of all terms used throughout the architectural data and presentations.
CV-1: Vision	The overall vision for transformational endeavors, which provides a strategic context for the capabilities described and a high-level scope.
CV-2: Capability Taxonomy	A hierarchy of capabilities which specifies all the capabilities that are referenced throughout one or more Architectural Descriptions.
CV-3: Capability Phasing	The planned achievement of capability at different points in time or during specific periods of time. The CV-3 shows the capability phasing in terms of the activities, conditions, desired effects, rules complied with, resource consumption and production, and measures, without regard to the performer and location solutions.
CV-4: Capability Dependencies	The dependencies between planned capabilities and the definition of logical groupings of capabilities.
CV-5: Capability to Organizational Development Mapping	The fulfillment of capability requirements shows the planned capability deployment and interconnection for a particular Capability Phase. The CV-5 shows the planned solution for the phase in terms of performers and locations and their associated concepts.
CV-6: Capability to Operational Activities Mapping	A mapping between the capabilities required and the operational activities that those capabilities support.
CV-7: Capability to Services Mapping	A mapping between the capabilities and the services that these capabilities enable.
DIV-1:Conceptual Data Model	The required high-level data concepts and their relationships.
DIV-2: Logical Data Model	The documentation of the data requirements and structural business process (activity) rules. In DoDAF V1.5, this was the OV-7.
DIV-3: Physical Data Model	The physical implementation format of the Logical Data Model entities, e.g., message formats, file structures, physical schema. In DoDAF V1.5, this was the SV-11.
OV-1: High-Level Operational Concept Graphic	The high-level graphical/textual description of the operational concept.
OV-2: Operational Resource Flow	A description of the Resource Flows exchanged between

Table 4.1-1.2-1: DoDAF V2.0 Models

Models	Descriptions
Description	operational activities.
OV-3: Operational Resource Flow Matrix	A description of the resources exchanged and the relevant attributes of the exchanges.
OV-4: Organizational Relationships Chart	The organizational context, role or other relationships among organizations.
OV-5a: Operational Activity Decomposition Tree	The capabilities and activities (operational activities) organized in a hierarchal structure.
OV-5b: Operational Activity Model	The context of capabilities and activities (operational activities) and their relationships among activities, inputs, and outputs; Additional data can show cost, performers, or other pertinent information.
OV-6a: Operational Rules Model	One of three models used to describe activity (operational activity). It identifies business rules that constrain operations.
OV-6b: State Transition Description	One of three models used to describe operational activity (activity). It identifies business process (activity) responses to events (usually, very short activities).
OV-6c: Event-Trace Description	One of three models used to describe activity (operational activity). It traces actions in a scenario or sequence of events.
PV-1: Project Portfolio Relationships	It describes the dependency relationships between the organizations and projects and the organizational structures needed to manage a portfolio of projects.
PV-2: Project Timelines	A timeline perspective on programs or projects, with the key milestones and interdependencies.
PV-3: Project to Capability Mapping	A mapping of programs and projects to capabilities to show how the specific projects and program elements help to achieve a capability.
SvcV-1 Services Context Description	The identification of services, service items, and their interconnections.
SvcV-2 Services Resource Flow Description	A description of Resource Flows exchanged between services.
SvcV-3a Systems-Services Matrix	The relationships among or between systems and services in a given Architectural Description.

Table 4.1-1.2-1: DoDAF V2.0 Models

Models	Descriptions
SvcV-3b Services-Services Matrix	The relationships among services in a given Architectural Description. It can be designed to show relationships of interest, (e.g., service-type interfaces, planned vs. existing interfaces).
SvcV-4 Services Functionality Description	The functions performed by services and the service data flows among service functions (activities).
SvcV-5 Operational Activity to Services Traceability Matrix	A mapping of services (activities) back to operational activities (activities).
SvcV-6 Services Resource Flow Matrix	It provides details of service Resource Flow elements being exchanged between services and the attributes of that exchange.
SvcV-7 Services Measures Matrix	The measures (metrics) of Services Model elements for the appropriate time frame(s).
SvcV-8 Services Evolution Description	The planned incremental steps toward migrating a suite of services to a more efficient suite or toward evolving current services to a future implementation.
SvcV-9 Services Technology & Skills Forecast	The emerging technologies, software/hardware products, and skills that are expected to be available in a given set of time frames and that will affect future service development.
SvcV-10a Services Rules Model	One of three models used to describe service functionality. It identifies constraints that are imposed on systems functionality due to some aspect of system design or implementation.
SvcV-10b Services State Transition Description	One of three models used to describe service functionality. It identifies responses of services to events.
SvcV-10c Services Event-Trace Description	One of three models used to describe service functionality. It identifies service-specific refinements of critical sequences of events described in the Operational Viewpoint.
StdV-1 Standards Profile	The listing of standards that apply to solution elements.
StdV-2 Standards Forecast	The description of emerging standards and potential impact on current solution elements, within a set of time frames.
SV-1 Systems Interface Description	The identification of systems, system items, and their interconnections.
SV-2 Systems Resource Flow Description	A description of Resource Flows exchanged between systems.

Table 4.1-1.2-1: DoDAF V2.0 Models

Models	Descriptions
SV-3 Systems-Systems Matrix	The relationships among systems in a given Architectural Description. It can be designed to show relationships of interest, (e.g., system-type interfaces, planned vs. existing interfaces).
SV-4 Systems Functionality Description	The functions (activities) performed by systems and the system data flows among system functions (activities).
SV-5a Operational Activity to Systems Function Traceability Matrix	A mapping of system functions (activities) back to operational activities (activities).
SV-5b Operational Activity to Systems Traceability Matrix	A mapping of systems back to capabilities or operational activities (activities).
SV-6 Systems Resource Flow Matrix	Provides details of system resource flow elements being exchanged between systems and the attributes of that exchange.
SV-7 Systems Measures Matrix	The measures (metrics) of Systems Model elements for the appropriate timeframe(s).
SV-8 Systems Evolution Description	The planned incremental steps toward migrating a suite of systems to a more efficient suite, or toward evolving a current system to a future implementation.
SV-9 Systems Technology & Skills Forecast	The emerging technologies, software/hardware products, and skills that are expected to be available in a given set of time frames and that will affect future system development.
SV-10a Systems Rules Model	One of three models used to describe system functionality. It identifies constraints that are imposed on systems functionality due to some aspect of system design or implementation.
SV-10b Systems State Transition Description	One of three models used to describe system functionality. It identifies responses of systems to events.
SV-10c Systems Event-Trace Description	One of three models used to describe system functionality. It identifies system-specific refinements of critical sequences of events described in the Operational Viewpoint.

Within the DoDAF Meta-model, the elements for the DoDAF-described Models are modeled with time periods (temporal extents) that can be in the future, and the models can be used to describe requirements. A requirement is a two-party agreement, between a requirer and a require-ee. An OV DoDAF-described Model could be used to describe a business process (activity) requirement while an SV DoDAF-described Model might be used to describe a system requirement.

To aid the decision-maker and process owners, the DoDAF-described Models have been categorized into the following types:

- Tabular: Models which present data arranged in rows and columns, which includes structured text as a special case.
- Structural: This category comprises diagrams describing the structural aspects of an architecture.
- Behavioral: This category comprises diagrams describing the behavioral aspects of an architecture.
- Mapping: These models provide matrix (or similar) mappings between two different types of information.
- Ontology: Models which extend the DoDAF ontology for a particular architecture.
- Pictorial: This category is for free-form pictures.
- Timeline: This category comprises diagrams describing the programmatic aspects of an architecture.

DoDAF Architectural Descriptions are expressed in the form of sets of data, expressed as DoDAF-described Models, which can be classified into categories. **Table 1.2-2** below provides a summary of how the DoDAF-described Models can be sorted using the categories above and can provide insight for the decision-maker and process owners for the DoDAF-described Models needed.

Table 1.2-2: DoDAF-Described Models Categorized by Type

Category / VP	Tabular	Structural	Behavioral	Mapping	Taxonomy	Pictorial	Timeline
All Viewpoint	AV-1				AV-2		
Capability	CV-1	CV-4		CV-6 CV-7	CV-2		CV-3 CV-5
Operational	OV-3	OV-2 OV-4	OV-6a OV-6b OV-6c		OV-5	OV-1	
System	SV-6 SV-7 SV-9	SV-1 SV-2	SV-4 SV-10a SV-10b SV-10c	SV-3 SV-5a SV-5b			SV-8
Standards	StdV-1 StdV-2						
Data and Information		DIV-1 DIV-2 DIV-3					
Service	SvcV-6 SvcV-7 SvcV-9	SvcV-1 SvcV-2	SvcV-4 SvcV-10a SvcV-10b SvcV-10c	SvcV-3a SvcV-3b SvcV-5			SvcV-8
Project		PV-1		PV-3			PV-2

Some of the DoDAF-described Models above were based on analysis of Ministry of Defence Architecture Framework (MODAF) and North Atlantic Treaty Organization (NATO) Architecture Framework (NAF) views and information requirements provided in the key process workshops [2]by the subject matter experts. In addition, analysis on the DoDAF V1.5 products was performed by the DoDAF V2.0 Presentation Technical Working Group[3]. The objective of the analysis was to determine if any product could be eliminated or if any product was created in every architecture effort. The OV-1 is the most created product at 92 percent of the projects. The SV-7 was the least created product at 5 percent. What is revealing is that there was not a product that could be deleted. The results of the survey are documented in the *DoDAF Product Development Questionnaire Analysis Report.doc* online in the DoDAF Journal.

In addition, based on the level of the architecture effort, the decision-maker and architect need to determine the DoDAF-described Models and Fit-for-Purpose Views needed. To assist,

[2] JCIDS, SE, and Operations workshops were conducted. Other key process workshops, PPBE and Defense Acquisition System (DAS), were not conducted.
[3] The Presentation Technical Working Group reported into the DoDAF Core Management Group and worked with the DoDAF Development Team. The Presentation Technical Working Group focus was on presenting architecture in meaningful ways to the decision-makers.

[Table 1.2-3](#) uses the Zachman Framework[4] with the levels of architecture overlaid for consideration by the decision-maker and architect. Table 1.2-3 is only provided as input; DoDAF is not prescribing DoDAF-described Model or Fit-for-Purpose Views or presentations.

Table 1.2-3: Zachman Framework with Levels of Architecture

Strategic Architectures apply to entire Department | Capability Architectures specific to CPM & Component Tiers | Solution Architecture: Materiel/Non-materiel | DoD EA: Strategic, Capability

	Layer	What (Data)	How (Function)	Where (Network)	Who (People)	When (Time)	Why (Motivation)
1	Scope Context Boundary **(Planner)**	List of things important to the business	List of processes the business performs	List of locations in which the business operates	List of organizations important to the business	List of events significant to the business	List of business goals/ strategies
2	Business Model Concepts **(Owner)**	e.g., Semantic or Entity-relationship Model	e.g., Business Process Model	e.g., Business Logistics System	e.g., Work Flow Model	e.g., Master Schedule	e.g., Business Plan
3	System Model Logic **(Designer)**	e.g., Logical Data Model	e.g., Application Architecture	e.g., Distributed System Architecture	e.g., Human Interface Architecture	e.g., Processing Structure	e.g., Business Rule Model
4	Technology Model Physics **(Builder)**	e.g., Physical Data Model	e.g., System Design	e.g., Technology Architecture	e.g., Presentation Architecture	e.g., Control Structure	e.g., Rule Design
5	Component Configuration **(Implementer)**	e.g., Data Definition	e.g., Program	e.g., Network Architecture	e.g., Security Architecture	e.g., Timing Definition	e.g., Rule Specification
6	Functioning Enterprise Instances **(Worker)**	e.g., Data	e.g., Function	e.g., Network	e.g., Organization	e.g., Schedule	e.g., Strategy

4.1.1 Architecture Interrogatives

A critical part of defining an architecture is answering what is known as, the set of standard interrogatives, which are the set of questions, *who*, *what*, *when*, *where*, *why*, and *how*, that facilitate collection and usage of architecture-related data. DoDAF provides a means of answering these interrogatives through the DoDAF Viewpoints and DoDAF-described Models (described further in Volume II), and the DoDAF Meta-model Data Groups, introduced in Section 9 of Volume I as the major parts of the DoDAF Conceptual Data Model (CDM).

[Table 1.2.1-1](#) is a simple matrix that presents the DoDAF Viewpoints and DoDAF-described Models as they relate to the DoDAF Meta-model Groups, and how these viewpoints, models, and groups answer the standard interrogatives. When architecture is required to support decision-making, the matrix is useful in both data collection, and decisions on how to best represent the data in DoDAF-described Models that are appropriate to the purpose for which the architecture is created.

[4] Zachman, John. Zachman Framework. © Zachman International. The Zachman Framework can be found at the Zachman International Website: http://zachmaninternational.com/index.php/the-zachman-framework/26-articles/13-the-zachman-framework-a-concise-definition

Table 1.2.1-1: Standard Interrogatives Matrix

	What (Date)	How (Function)	Where (Network)	Who (People)	When (Time)	Why (Motivation)
Viewpoint	AV, DIV	OV, SV, SvcV	OV, SV, SvcV	OV	CV, OV, PV, SV, SvcV	AV, CV, OV, StdV, SV, SvcV
DoDAF-described Models	AV-2, DIV-1, DIV-2, DIV-3	OV-5a, OV-5b, OV-6a, b, c, SV-4, SV-10a, b, c, SvcV-10a, b, c	OV-2, SV-2, SvcV-2	OV-2, OV-4	CV-2, CV-4, OV-6c, PV-2, SV-8, SvcV-8, Sv-10c, SvcV-10c	AV-1, CV-1, OV-6a, StdV-1, StdV-2, SV-10a, SvcV-10a
Meta-model group	Information and Data, Project	Activity, Capability, Service, Measures	Location	Performer	All	Rules, Goals

As an example, a decision is required on changing a logistics transaction process (a composite of activities). The process is documented (*how*), to include the measures of performance, services required, and the capability supported by the action (activity). Data required to execute the process (*what*) is collected concurrently. Included in that data collection is the location and other administrative data on the place of process execution (*where*), and the performers of the action (*who*). The time frames required (*when*) and the Rules, Goals, and Expected Results (*why*) are also determined. These interrogatives impact on measures of performance. Each of these interrogatives can be represented by either a DoDAF-described Model or a Fit-for-Purpose View defined by the architectural development team that meets agency requirements. Either way, the models and views needed are created utilizing data defined and derived from the DoDAF Meta-model.

The architecture interrogatives are overlaid on the DM2 Conceptual Data Model in **Figure 1.2.1-2:**

- The Data Description — What (DM2 generalizes to other Resources besides just Data)
- The Function Description — How (and also the Performer that performs the Function, Measures, Rules, and Conditions associated with)
- The Network Description — Where (generalized)
- The People Description — Who (DM2 includes Organizations)
- The Time Description — When
- The Motivation Description — Why (broadened to include Capability requirements)

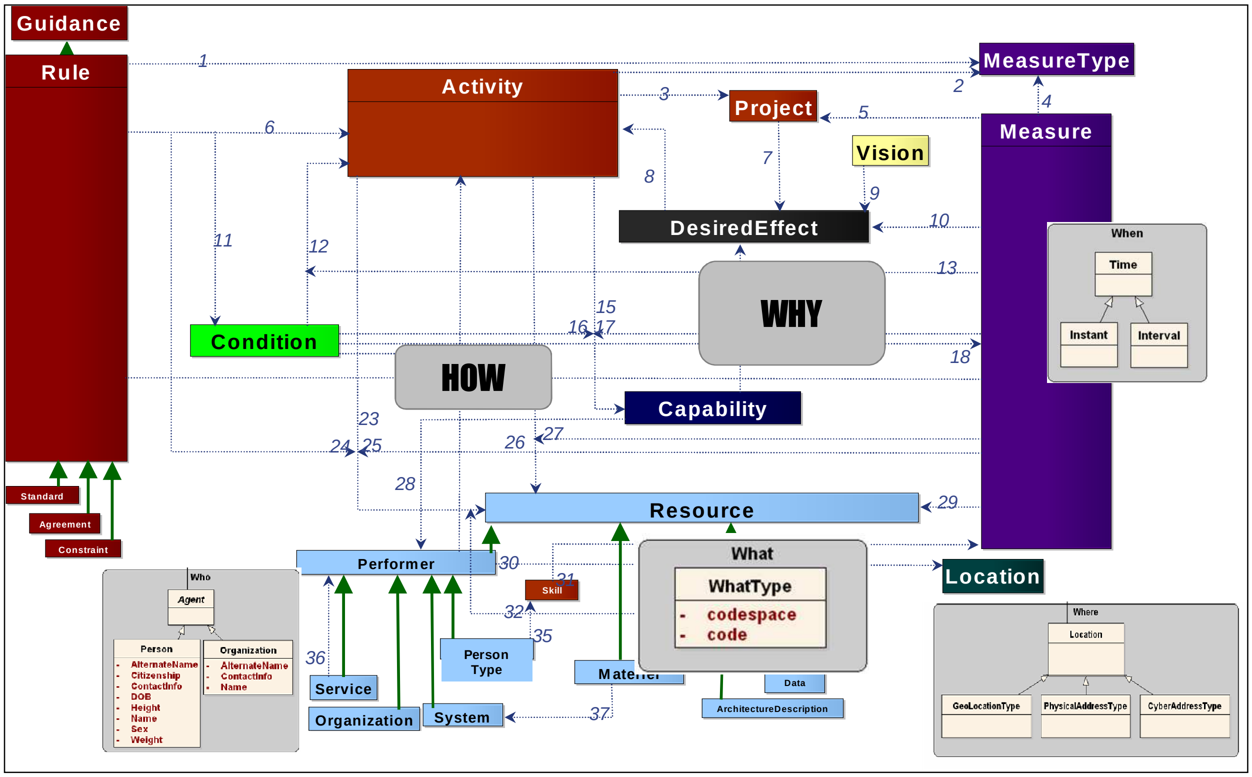


Figure 1.2.1-2: Architecture Interrogative overlay on the DM2 Conceptual Data Model

4.1.2 Architecture Modeling Primitives

Work is presently underway within the Department to ensure uniform representation for the same semantic content within architecture viewing, called Architecture Modeling Primitives. The Architecture Modeling Primitives, hereafter referred to as Primitives, will be a standard set of viewing elements and associated symbols mapped to DM2 concepts and applied to viewing techniques. Use of the Primitives to support the collection of architecture content in concert with the Physical Exchange Specification will aid in generating common understanding and improving communication. As the Primitives concepts are applied to more viewing techniques, they will be updated in the DoDAF Journal and details provided in subsequent releases of DoDAF. When creating an OV-6c in Business Process Modeling Notation (BPMN), the primitives notation may be used. DoD has created the notation and it is in the DoDAF Journal. The full range of Primitives for DoDAF-described Models, as with the current BPMN Primitives, will be coordinated for adoption by architecture tool vendors. Examples of presentations can be viewed online in the public DoDAF Journal.

4.2 Scoping Architectures to Be "Fit-for-Purpose"

Establishing the scope of architecture is critical to ensuring that its purpose and use are consistent with specific project goals and objectives. The term "Fit-for-Purpose" is used in DoDAF to describe an architecture (and its views) that is appropriately focused (i.e., responds to the stated goals and objectives of process owner, is useful in the decision-making process, and responds to internal and external stakeholder concerns. Meeting intended objectives means those actions that either directly support customer needs or improve the overall process undergoing change.

The architect is the technical expert who translates the decision-maker's requirements into a set of data that can be used by engineers to design possible solutions.

At each tier of the DoD, goals and objectives, along with corresponding issues that may exist should be addressed according to the established scope and purpose, (e.g., Departmental, Capability, SE, and Operational), as shown in the notional diagram in **Figure 2-1**.

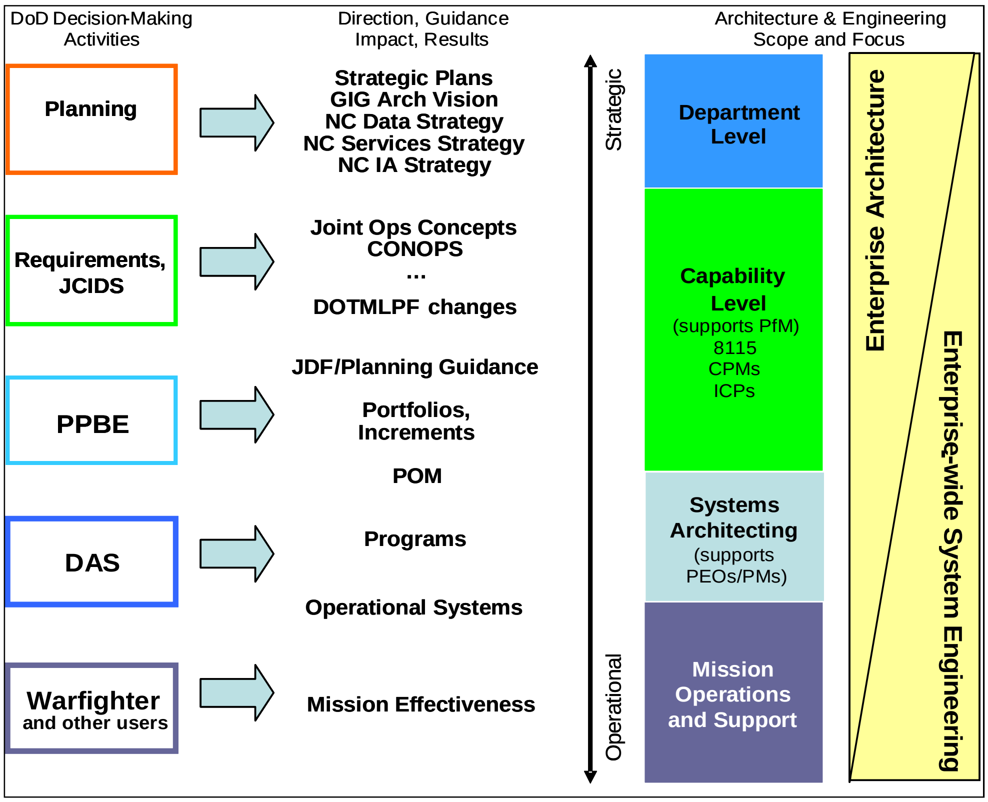


Figure 2-1: Establishing the Scope for Architecture Development

Establishing a scope for an architecture effort at any tier is similarly critical in determining the architecture boundaries (Purpose and Use expected), along with establishing the data categories needed for analysis and management decision-making. Scope also defines the key players whose input, advice, and consensus is needed to successfully architect and implement change (i.e., Stakeholders, both internal and external). Importantly, scope also determines the goals and objectives of the effort, consistent with both boundaries and stakeholders; since goals and objectives define both the purpose for architecture creation and the level of the architecture. Establishing the scope of an effort also determines the level of complexity for data collection and information presentation.

Architecture development also requires an understanding of external requirements that may influence architecture creation. An architecture developed for an internal agency purpose still needs to be mappable, and consistent with, higher level architectures, and mappable to the DoD EA. For some architecture developments, consideration must be given in data collection and graphical presentation to satisfaction of other external requirements, such as upward reporting and submission of architectural data and models for program review, funding approval, or budget review due to the sensitivity or dollar value of the proposed solution. Volume II contains guidance on data collection for specific views required by instruction, regulation, or other regulatory guidance (i.e., Exhibit 53, or Exhibit 300 submissions; OMB Segment architecture reviews, or interoperability requirements).

Architecture scoping must facilitate alignment with, and support the decision-making process and ultimately mission outcomes and objectives as shown in **Figure 2-2**. Architectural data and supporting views, created from organizing raw data into useful information, and collected into a useful viewpoint, should enable domain experts, program managers, and decision makers to utilize the architecture to locate, identify, and resolve definitions, properties, facts, constraints, inferences, and issues, both within and across architectural boundaries that are redundant, conflicting, missing, and/or obsolete. DoDAF V2.0 provides the flexibility to develop both Fit-for-Purpose Views (User-developed Views) and views from DoDAF-described Models to maximize the capability for decision-making at all levels. Figure 2-2 below shows how the development of architectures supports the management decision process. In this case, the example shows how an architecture and the use of it in analysis can facilitate the ability to determine and/or validate mission outcome.

Analysis also uncovers the effect and impact of change ("what if") when something is redefined, redeployed, deleted, moved, delayed, accelerated, or no longer funded. Having a disciplined process for architecture development in support of analytics will produce quality results, not be prone to misinterpretations, and therefore, be of high value to decision makers and mission outcomes.

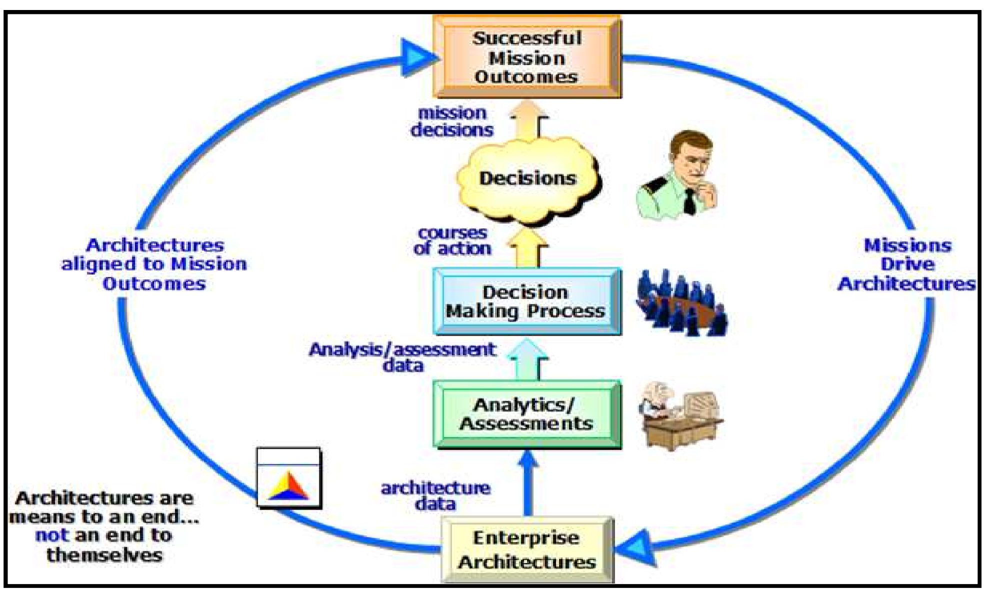


Figure 2-2: Mission Outcomes Supported by Architectures

4.3 Uses for DoDAF Viewpoint models

4.3.1 Uses of All Viewpoint models

All Viewpoint models capture the scope of an architectural description and where an architectural description fits with other architectural descriptions. All Viewpoint models are used to register architectural descriptions with the Defense of Defense Architecture Repository System (DARS).

4.3.2 Uses of Capability Viewpoint models

Capability Viewpoint models support various decision processes within the Department, including portfolio management. Since the DoD has moved toward the delivery of capabilities, these models take on a more important role. Developing an architecture that includes the relationships necessary to enable a capability thread is essential to improving usability of architectures, as well as increasing the value of federation.

In the above context, a capability thread is similar to the result of a query that would be run on a particular capability. For example, if an architecture were to include operational activities, rules, and systems, a capability thread would equate to the specific activities, rules, and systems that are linked to that particular capability. CV models are used to provide the strategic perspective and context for other architectural information.

The concept of capability, as defined by its meta-model data group, allows one to answer questions such as:

1. How does a particular capability or capabilities support the overall mission/vision?
2. What outcomes are expected to be achieved by a particular capability or set of capabilities?
3. What services are required to support a capability?
4. What is the functional scope and organizational span of a capability or set of capabilities?
5. What is our current set of capabilities that we are managing as part of a portfolio?

4.3.3 Uses of Data and Information Viewpoint models

Data and Information Viewpoint models provide means to ensure that only those information items that are important to an organization's operations and business are managed as part of the enterprise. These models are also useful tools for discussions among the various stakeholders of an architecture (e.g., decision-makers, architects, developers). These stakeholders need different levels of abstraction to support their roles within an enterprise.

The concepts, information requirements, and data items represented by DIV models are resources such as guidance and rules that shape activities and resources that are consumed and produced by activities. DIV models tie data managed within a described architecture to activities that need information for decision making. This allows information identified by DIV models to be traced to the strategic drivers of an architecture. This also allows data to be used to map services and systems where data are implemented to business operations that use that data. In particular, DIV-1 models of concepts are helpful when discussing this traceability with decision-makers and executives of an enterprise.

DIV-2 models of information requirements bridge the gap between DIV-1 models of essential concepts and DIV-3 models of the physical implementation of data. DIV-2 models, sometimes known as logical data models, introduce attributes and structural rules that form needed data structures. A DIV-2 model provides more detail about information than a corresponding DIV-1 model and communicates more facets of basic concepts to architects, systems analysts, and other stakeholders who are concerned about the information needs of decision makers. A DIV-2 model bridges the gap between architectural concepts and the physical implementation of systems. In particular, the information requirements provided by DIV-2 models are helpful in the verification and validation of services and systems.

DIV-3 models, sometimes known as physical data models, determine data schemas for the storage (e.g., database schemas), manipulation (e.g., class models), and exchange (e.g., message formats) of data consumed and produced by a described architecture. Data elements of DIV-3 models trace directly back to corresponding information requirements of DIV-2 models. In particular, XML message sets and other physical exchange specifications defined by DIV-3 models are helpful in the verification and maintenance of physical data exchange media and interfaces.

Any information and data that an organization manages through enterprise architecture should be specified by Data and Information Viewpoint models.

4.3.4 Uses of Operational Viewpoint models

The OV models may be used to describe a requirement for a “to-be” architecture in logical terms or as a simplified description of the key behavioral and information aspects of an “as-is” architecture. The OV models re-use the capabilities defined in the Capability Viewpoint and put them in the context of an operation or scenario. The OV models can be used in a number of ways, including the development of user requirements, capturing future concepts, and supporting operational planning processes.

One important way that architectural modeling supports the definition of requirements is in terms of boundary definition. Boundary definition is a process that often requires a significant degree of stakeholder engagement; the described models provided by DoDAF provide ideal support for this interactive process. The DoDAF provides support to the concept of functional scope and organizational span. When performing analysis of requirements relative to a particular capability or capabilities, it is important to know the specific functionality intended to be delivered by the capability. It is also important to know the limits of that functionality, to be able to determine necessary interfaces to activities that are parts of other capabilities and to organizations that carry out those activities. The use of OV models supports identification of the boundaries of capabilities, thus rendering the functional scope and organizational span.

Operational Viewpoint models support interoperability analyses in many ways, and they may specify user-level interoperability requirements.

Operational models can help answer questions like these:

1. What lines of business are pursued by an organization?

2. What activities support the lines of business of an organization?
3. What is the functional scope of the capability or capabilities for which I am responsible? This can be answered by a combination of information from the activity model and CV models.
4. What is the organizational span of influence of this capability or capabilities?
5. What information must be passed between capabilities?
6. What strategic drivers require that certain data are passed or tracked? This can be answered by a combination of data within the logical data model, information exchanges, activities, and CV models.
7. What activities are being supported or automated by a capability or capabilities?
8. What role does organization X play within a capability or capabilities?
9. What are the functional requirements driving a particular capability?
10. What rules are applied within a capability, and how are they applied?

4.3.5 Uses of Project Viewpoint models

Project Viewpoint models contain information that improves DoDAF's support to the portfolio management process. It is important to be able to look across portfolios (i.e., groups of investments) to ensure that all possible alternatives for a particular decision have been exhausted to make the most informed decision possible in support of the Department. Relating project information to the responsible organizations, as well as to other projects, forms a valuable architecture construct that supports PfM.

Incorporation of these models also makes the DoDAF a value-added framework to support the PPBE process. These models are especially applicable to the Programming phase of the PPBE process. It is within this phase that the Program Objective Memorandum (POM) is developed. The POM seeks to construct a balanced set of programs that respond to the guidance and priorities of the Joint Programming Guidance within fiscal constraints. When completed, the POM provides a fairly detailed and comprehensive description of the proposed programs, which can include a time-phased allocation of resources (personnel, funding, materiel, and information) by program projected into the future. The information captured within the Project models (e.g., project relationships, timelines, capabilities) can be used within the PPBE process to develop the POM. Using these models enables decision-makers to perform well-informed planning and complements the use of Capability Viewpoint models.

Project Viewpoint models can be used to answer questions such as:

1. What capabilities are delivered as part of this project?
2. Are there other projects that either affect or are affected by this project? To what portfolios do the projects or projects belong?
3. What are the important milestones relative to this project? When can I expect capabilities to be rendered by this project to be in place?

4.3.6 Uses of Service Viewpoint models

The relationship between architectural data elements across the Service Viewpoint to the Operational Viewpoint and Capability Viewpoint can be exemplified as services are procured and fielded to support the operations and capabilities of organizations. The structural and behavioral models in the Operational Viewpoint and the Service Viewpoint allow architects and stakeholders to quickly tell which functions are carried out by humans and which by services for each alternative specification and so carry out trade analysis based on risk, cost, reliability, and other factors.

Within the development process, Service Viewpoint models describe the environment in which the operational activities and resources are required to function to provide and support the development process (JCIDS) and the Defense Acquisition System or capability development within the JCAs.

The concept of service, as defined by its meta-data model data group, within section 2, allows one to answer questions such as:

How does the service support the overall mission, vision, goals, and objectives?

What outcomes are expected to be achieved by the service or family of services?

1. What systems are required to support the service?
2. What capabilities are required to support the service?
3. What is the functional scope and organizational span of a service of family of services?
4. What is the characteristic of services for our portfolio?
5. What is current set of services that we are managing as part of a portfolio?

Within the development process, Service Viewpoint models describe the design for service-based solutions to support operational requirements from the development processes (JCIDS) and the Defense Acquisition System or from capability development within the JCAs.

4.3.7 Uses of Standards Viewpoint models

The Standards Viewpoint can articulate the applicable policy, standards, guidance, constraints, and forecasts required by JCIDS, DAS, systems engineering, PPBE, operations, other process owners, and decision-makers.

4.3.8 Uses of System Viewpoint models

f. Within the development process, System Viewpoint models describe the design for system-based solutions to support or enable requirements created by the operational development processes (JCIDS) and the Defense Acquisition System.

Within the development process, System Viewpoint models describe the design for system-based solutions to support or enable requirements created by the operational development processes (JCIDS) and the Defense Acquisition System.

The concept of system, as defined by its meta-data model data group, allows one to answer questions such as:

1. How does the system support the overall mission, vision, goals, and objectives?
2. What outcomes are expected to be achieved by the system or family of systems?
3. What systems are required to support the system?
4. What capabilities are required to support the system?
5. What is the functional scope and organizational span of a system of family of systems?
6. What is the characteristic of systems for our portfolio?
7. What is current set of systems that we are managing as part of a portfolio?

5 Architecture Methodologies

This section introduces a methodology-based approach to Architectural Description development in DoD, draws on the methodology originally introduced in DoDAF V1.5, and expands on that methodology to highlight its use in a data-driven, net-centric architecture development environment. The methodology contained in this section is notional, represents best practices that have evolved over time, and can be utilized in conjunction with, or as a replacement for other methodologies, as described below.

5.1 6-Step Architecture Development Process

The high-level, 6-step architecture development process provides guidance to the architect and Architectural Description development team and emphasizes the guiding principles described in Section 3.5.1. The process is data-centric rather than product-centric (e.g., it emphasizes focus on data, and relationships among and between data, rather than DoDAF V1.0 or V1.5 products). This data-centric approach ensures concordance between views in the Architectural Description while ensuring that all essential data relationships are captured to support a wide variety of analysis tasks. The views created as a result of the architecture development process provide visual renderings of the underlying architectural data and convey information of interest from the Architectural Description needed by specific user communities or decision makers. **Figure 7.1.1-1** depicts this 6-step process.

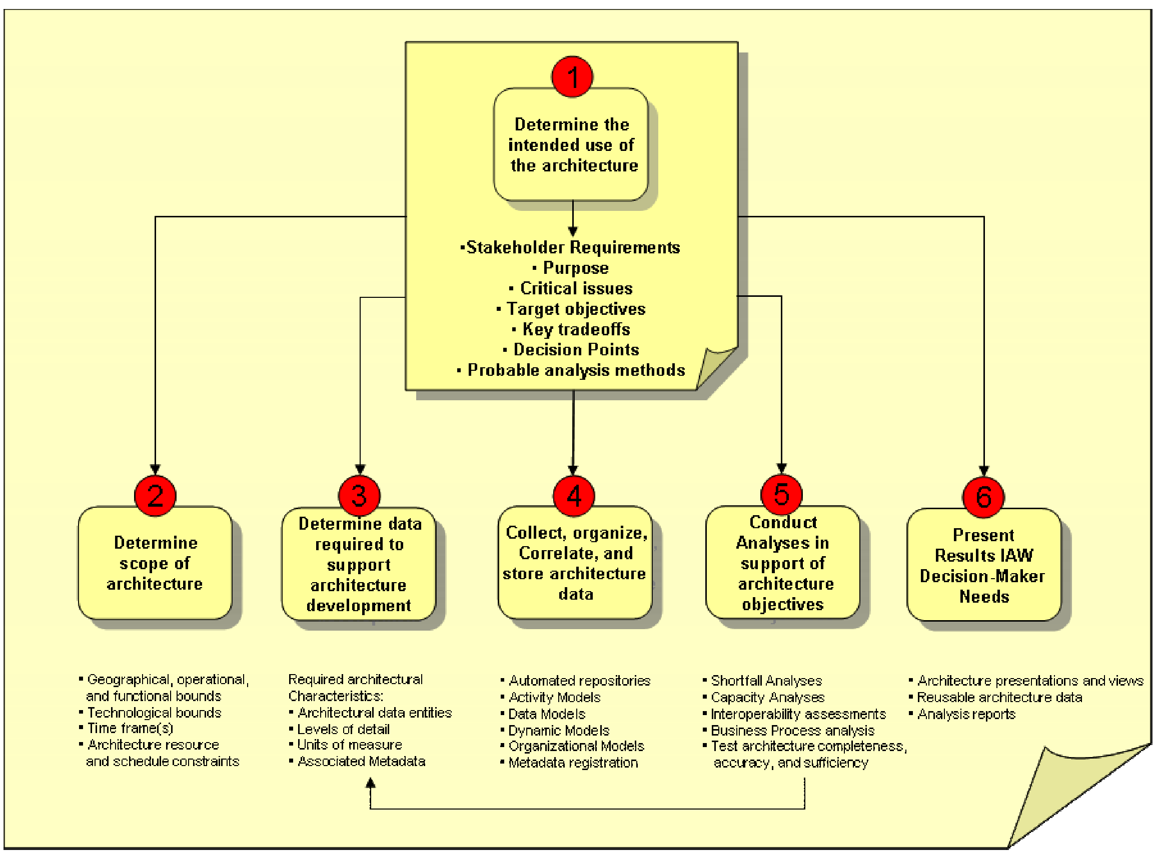


Figure 7.1.1-1: Architecture Development 6-Step Process

NOTE: It is important to note in this section that the development of Architectural Description is an iterative process and a unique one, in that every Architectural Description is:

- Different in that architecture creation serves a specific purpose, and is created from a particular viewpoint.
- Serving differing requirements, necessitating different types of views to represent the collected data.
- Representative of a ‘snapshot in time’ (e.g., the Architectural Description may represent the current view or baseline, or it may represent a desired view in some future time).
- Changeable over time as requirements become more focused or additional knowledge about a process or requirement becomes known.

The methodology described below is designed to cover the broadest possible set of circumstances, and also to focus on the most commonly used steps by the architecture community.

5.1.1 Step 1: Determine Intended Use of Architecture.

Defines the purpose and intended use of the architecture (“Fit-for-Purpose”); how the Architectural Description effort will be conducted; the methods to be used in architecture development; the data categories needed; the potential impact on others; and the process by which success of the effort will be measured in terms of performance and customer satisfaction. This information is generally provided by the process owner to support architecture development describing some aspect of their area of responsibility (process, activity, etc.).

A template for collection of high-level information relating to the purpose and scope of the Architectural Description, its glossary, and other information, has been developed for registration of that data in DARS. An electronic copy is found on the public page of DARS.

5.1.2 Step 2: Determine Scope of Architecture.

The scope defines the boundaries that establish the depth and breadth of the Architectural Description and establish the architecture’s problem set, helps define its context and defines the level of detail required for the architectural content. While many architecture development efforts are similar in their approach, each effort is also unique in that the desired results or effect may be quite different. As an example, system development efforts generally focus first on process change, and then concentrate on those automated functions supporting work processes or activities. In addition to understanding the process, discovery of these ‘system functions’ is important in deciding how to proceed with development or purchase of automation support.

Information collected for Architectural Descriptions describing services is similar to information collected for Architectural Descriptions describing systems. For describing services, Architectural Description will collect additional information concerning subscriptions, directory services, distribution channels within the organization, and supporting systems/communications web requirements.

Similar situations occur with Architectural Description development for joint operations. Joint capabilities are defined processes with expected results, and expected execution capability dates. The Architectural Descriptions supporting the development of these types of capabilities usually require the reuse of data already established by the military services and agencies, analyzed, and configured into a new or updated process that provides the desired capability. Included are the processes needed for military service and/or agency response, needed automation support, and a clear definition of both desired result and supporting performance measures (metrics). These types of data are presented in models further described in Volume II.

The important concept for this step is the clarity of scope of effort defined for the project that enables an expected result. Broad scoping or unclear definition of the problem can delay or prevent success. The process owner has the primary responsibility for ensuring that the scoping is correct, and that the project can be successfully completed.

Clarity of scope can better be determined by defining and describing the data to be used in the proposed Architectural Description in advance of the creation of views that present desired data in a format useful to managers. Early identification of needed data, particularly data about the Architectural Description itself, the subject-matter of the proposed Architectural Description, and a review of existing data from COIs, can provide a rich source for ensuring that Architectural Descriptions, when developed, are consistent with other existing Architectural Descriptions. It also ensures conformance with any data-sharing requirements within the Department or individual COIs, and conformant with the DM2 described in Section 9.

An important consideration beginning with this and each subsequent step of the architecture development process is the continual collection and recording of a consistent, harmonized, and common vocabulary. The collection of terms should continue throughout the architecture development process. As architectural data is identified to help clarify the appropriate scope of the architecture effort, vocabulary terms and definitions should be disambiguated, harmonized, and recorded in a consistent AV-2 process documented in the “DoDAF V2.0 Architecture Development Process for the DoDAF-described Models” Microsoft Project Plan.

Analysis of vocabularies across different Architectural Descriptions with similar scope may help to clarify and determine appropriate Architectural Description scope. Specific examples of data identification utilizing the AV-2 Data Dictionary construct are found in the DoDAF Journal.

5.1.3 Step 3: Determine Data Required to Support Architecture Development.

The required level of detail to be captured for each of the data entities and attributes is determined through the analysis of the process undergoing review conducted during the scoping in Step 2. This includes the data identified as needed for execution of the process, and other data required to effect change in the current process, (e.g., administrative data required by the organization to document the Architectural Description effort). These considerations establish the type of data collected in Step 4, which relate to the architectural structure, and the depth of detail required.

The initial type of architectural data content to be collected is determined by the established scope of the Architectural Description, and recorded as attributes, associations, and concepts as described in the DM2. A mapping from DM2 concepts, associations, and attributes to architecture models is provided that suggests relevant architectural views the architect may develop (using associated architecture techniques) during the more comprehensive and coherent data collection of Step 4. This step is normally completed in conjunction with Step 4, a bottom-up approach to organized data collection, and Architectural Description development typically iterates over these two steps. As initial data content is scoped, additional data scope may be suggested by the more comprehensive content of Architectural Views desired for presentation or decision-making purposes.

This step can often be simplified through reuse of data previously collected by others, but relevant to the current effort. Access to appropriate COI data and other architecture information, discoverable via DARS and the DMR, can provide information on data and other architectural views that may provide useful in a current effort.

Work is presently underway within the Department to ensure uniform representation for the same semantic content within architecture modeling, called Architecture Modeling Primitives. The Architecture Modeling Primitives, hereafter referred to as Primitives, will be a standard set of modeling elements, and associated symbols mapped to DM2 concepts and applied to modeling techniques. Using the Primitives to support the collection of architecture content and, in concert with the PES, will aid in generating common understanding and communication among architects in regard to architectural views. As the Primitives concepts are applied to more modeling techniques, they will be updated in the DoDAF Journal and details provided in subsequent releases of DoDAF. When creating an OV-6c in Business Process Modeling Notation (BPMN), the Primitives notation may be used. DoD has created the notation and it is in the DoDAF Journal. The full range of Primitives for views, as with the current BPMN Primitives, will be coordinated for adoption by architecture tool vendors.

5.1.4 Step 4: Collect, Organize, Correlate, and Store Architectural Data.

Architects typically collect and organize data through the use of architecture techniques designed to use views (e.g., activity, process, organization, and data models as views) for presentation and decision-making purposes. The architectural data should be stored in a recognized commercial or government architecture tool. Terms and definitions recorded are related to elements of the (DM2).

Designation of a data structure for the Architectural Description effort involves creation of a taxonomy to organize the collected data. This effort can be made considerably simpler by leveraging existing, registered artifacts registered in DARS of the DM2, to include data taxonomies and data sets. Each COI maintains its registered data on DARS, either directly or through a federated approach. In addition, some organizations, such as U.S. Joint Forces Command (JFCOM), have developed templates, which provide the basis of a customizable solution to common problems, or requirements, which includes datasets already described and registered in the DMR. Examples of this template-based approach are in the DoDAF Journal.

DARS provides more information that is specific, and guidance on retrieving needed data through a discovery process. Once registered data is discovered, the data can be cataloged and organized within a focused taxonomy, facilitating a means to determine what new data is required. New data is defined, registered in DARS, and incorporated into the taxonomy structure to create a complete defined list of required data. The data is arranged for upload to an automated repository, such as DARS, to permit subsequent analysis and reuse. Discovery metadata (i.e., the metadata that identifies a specific Architectural Description, its data, views, and usage) should be registered in DARS as soon as it is available to support discovery and enable federation. Architects and data managers should use the DoD EA Business Reference Model (DoD EA BRM) taxonomy elements as the starting point for their registration efforts. Additional discovery metadata, such as processes and services may be required later, and should follow the same registration process.

5.1.5 Step 5: Conduct Analyses in Support of Architecture Objectives.

Architectural data analysis determines the level of adherence to process owner requirements. This step may also identify additional process steps and data collection requirements needed to complete the Architectural Description and better facilitate its intended use. Validation applies the guiding principles, goals, and objectives to the process requirement, as defined by the process owner, along with the published performance measures (metrics), to determine the achieved level of success in the Architectural Description effort. Completion of this step prepares the Architectural Description for approval by the process owner. Changes required from the validation process, result in iteration of the architecture process (repeat steps 3 through 5 as necessary).

5.1.6 Step 6: Document Results in Accordance with Decision-Maker Needs.

The final step in the architecture development process involves creation of architectural views based on queries of the underlying data. Presenting the architectural data to varied audiences requires transforming the architectural data into meaningful presentations for decision-makers. This is facilitated by the data requirements determined in Step 3, and the data collection methods employed during Step 4.

DoDAF V2.0 provides for models and views. DoDAF-described Models are those models described in Volume II that enable an architect and development team whose data has already been defined and described consistent with the DM2. The models become views when they are populated with architectural data. These models include those previously described in earlier versions of DoDAF, along with new models incorporated from the MODAF, the NATO NAF, and TOGAF that have relevance to DoD architecture development efforts.

Fit-for-Purpose Views are user-defined views that an architect and development team can create to provide information necessary for decision-making in a format customarily used in an agency. These views should be developed consistent with the DM2, but can be in formats (e.g., dashboards, charts, graphical representations) that are normally used in an agency for briefing and decision purposes. An Architectural Description development effort can result in an Architectural Description that is a combination of DoDAF-described Models and Fit-for-Purpose Views.

DoDAF does not require specific models or views, but suggests that local organizational presentation types that can utilize DoDAF-created data are preferred for management presentation. A number of available architecture tools support the creation of views described in this step. The PES provides the format for data sharing.

NOTE: While DoDAF does not require specific models or views in an architecture, several JCS and DoD publications do require specific views in response to their stated requirements. Managers and architects, in deciding what views are created in an architecture development effort, must consider those specific requirements to ensure that the architecture developed is useful in satisfying those requirements.

5.2 Methodology Based Approach to Architecture

Several methodologies, with supporting tools, techniques, and notations (i.e., a set of written symbols used to represent something such as activity, decisions, systems, applications, interfaces) exist for developing Architectural Descriptions. While DoDAF does not promote a specific approach, the DoDAF provides the rules, standard entities, and relationships for developing Architectural Descriptions in a semantically consistent and interoperable fashion. The DoDAF V2.0 CDM and LDM, described in Volumes 1 and 2, along with the PES in Volume III, have been designed to facilitate adoption of DoDAF by a wide range of toolsets and techniques. The DM2 should be used as the principal reference for creating the data structures in toolsets to ensure both interoperability and reuse capabilities. An achievable level of commonality among the notations is possible when basing architecture development on the DoDAF V2.0 CDM and LDM.

NOTE: Several commercial toolsets that are commonly used to develop architecture views still use the terms 'model' of 'diagram' to describe those views. Within this chapter, we continue to use the terms 'model' and 'diagram', as they are used by toolset vendors, to avoid confusion. However, a model or diagram created by a toolset, using an appropriate notation, and included in a set of views in a DoD architecture should be understood as a 'view' within DoDAF.

The two most common techniques—the SADT Approach and the OOAD Approach—are discussed briefly below. Examples of the notation supporting these techniques are presented in examples contained within Volume II. Either of these techniques can be used with the methodology described above, or by others, such as MODAF, NAF, TOGAF, or other Government or commercial offerings.

The Webster's II New College Dictionary 2001 defines methodology as (1) the system of principles, procedures, and practices applied to a particular branch of knowledge, and, (2) the branch of logic dealing with the general principles of the formation of knowledge. Generally speaking, knowledge is gained through the acquisition of, and effective use of information organized from data for a particular purpose.

An architecture development methodology specifies how to derive relevant information about an enterprise's processes and business or operational requirements, and how to organize and model that information. Architecture methods describe consistent and efficient ways to collect data, organize the data in a particular grouping or structure, and store collected data for later presentation and use in decision-making processes. A methodology also provides a means for replicating the steps taken to create an Architectural Description for a specific purpose later, by another person or team with the expectation of achieving similar results.

In turn, through utilization of a method, it is possible to compare Architectural Descriptions created under the same, or similar methods, evaluate how disparate Architectural Descriptions can be linked to provide a higher-level picture of a process or capability, and to analyze the impact of future change. These analyses can include:

- Static Analyses – which could include capability audit, interoperability analysis, or functional analysis. These analyses are often performed using simple analysis tools such as paper-based comparisons and database queries.
- Dynamic Analyses – sometimes referred to as executable models, these analyses typically examine the temporal, spatial, or other performance aspects of a system through dynamic simulations. For example, these analyses might be used to assess the latency of time sensitive targeting systems or conduct traffic analyses on deployed tactical networks under a variety of loading scenarios.
- Experimentation – the use of tactical capability requirements, such as the Coalition Warrior Interoperability Demonstration (CWID), sponsored annually by the JCS, and various battle labs to provide the ability to conduct human-in-the-loop simulations of operational activities. Differing degrees of live versus simulated systems can be deployed during these experiments and there is a high degree of control over the experiment variables. These can be used for a variety of purposes.

The 6-step architecture development process described below is a generic, time-tested method, which can be utilized, in a wide range of architectural requirements through relatively simple adaptation. The examples described within the steps provide information on customization of the generic method for use in major departmental functions and operations.

NOTE: The methodology described in this section is applicable to development of SOA-based architectures. The steps described in the methodology, together with the requirements of the toolset, techniques and notation desired, should be considered together when defining a SOA. Volume II provides specific models that are useful for services-specific data collection, and presentation models and documents that describe services.

If another method is desired, then utilization of the information contained in this Volume, Volume II, Architectural Data and Models, and Volume III, the DM2 PES, provide the information needed for use in developing an Architectural Description. When utilizing another method, reference to the notional methodology can ensure adherence to the principles described in DoDAF V2.0, to maximize the potential for reuse of essential data, and also to ensure conformance with DoDAF V2.0.

5.2.1 Accommodating Multiple Methods for Implementation

DoDAF V2.0 is designed to be flexible in development of Architectural Descriptions supporting all tiers, capabilities, component-level views, and specific functional or operational requirements. The method described within the Framework is generic, and can be used in conjunction with other frameworks, tools, or techniques to achieve the desired result. Specifically, the conceptual model supporting DoDAF

V2.0 can be used to develop both relational and object-oriented (OO) databases in a wide variety of formats; supports both the structured analysis and Object-oriented analysis and design modeling techniques and their specific notations; and continues to support previous versions of this framework.

Many Architectural Descriptions are created utilizing data from Architectural Descriptions developed previously under another framework (i.e., MODAF, NAF, TOGAF). It is also possible, through data mapping, to link that data to the DoDAF V2.0 conceptual and LDMs, since the data models supporting these frameworks are based on either the predecessor C4ISR Framework or DoDAF V1.0.

5.2.2 Structured Technique Overview

Architectural Descriptions developed under a structured analysis-driven approach are process-oriented and characterized by hierarchical process decomposition. Historically, structured models generally used in DoD originated from the Integration Definition Language developed by the U.S. Air Force, and later used to develop the Integration Definition for Activity Modeling (IDEF0) [IDEF0 1993] Standards and the Federal Information Processing Standard (FIPS) published by the National Institutes for Standards & Technology (NIST). This technique evolved from an earlier, also process-driven approach, SADT, developed for the U.S. Air Force Materiel Command. More recently, architecture development using structured methods has also included those utilizing the BPMN, developed by the Business Process Management Initiative, and currently managed by the Object Management Group (OMG).

5.2.3 Process Data Flow

A process flow diagram (PFD) is a graphical representation of the flow of data through a process. With a process flow diagram, users are able to visualize how the process will operate, what the process will accomplish, and how the process is executed normally. Process flow diagrams can be used to provide the end user with a physical idea of the resulting actions that occur on data input, and how their actions ultimately have an effect upon the structure of the whole process. Process flow diagrams also define desired or required system-level functions—the level and type of automation desired to improve the time, efficiency, and results of executing a process.

5.2.4 Process Task-Dependency Diagram

Process Task Dependency (PTD) Diagrams lay out clearly the step-by-step flow of a process by tracking the flow of material, information or a service through all its steps in a logical or required order. The PTD diagram assists an unfamiliar audience to picture the steps of a process and clarifies misconceptions about how the process actually operates, while providing a reference for the handling of corrective action or process improvement. Task-sequence notations work especially well for uninterruptible processes, meaning a set of steps that exhibits clear dependencies, doesn’t execute until explicitly triggered, and normally continues until it achieves a clear exit criterion. Such processes are generally low-level and detailed, and useful for:

- Defining detailed performance measures (metrics) and measures capture.
- Establishing an information base for executable architecture/process simulation.
- Defining automation functional requirements.

5.2.5 Entity-Relation Model

The Entity-Relation Model describes the structure of an architecture domain's system data types and the business process rules that govern the system data. It provides a definition of architectural domain data types, their attributes or characteristics, and their interrelationships.

5.2.6 Object-Oriented Technique Overview

Object-oriented architectural views are created utilizing the Unified Modeling Language (UML) architecture technique and notation, together with the DoDAF logical and PES data structures. This technique describes the operational need, places data (objects, or 'performers' in the DoDAF data structure) in the context of its use, and provides a traceable foundation for system and software design. It is based on the concepts of data abstraction and inheritance from a service-oriented view. The object-oriented technique provides an orderly arrangement of the parts of the business organization and includes a style and method of design through its highly developed notation style.

5.2.7 Process – Activity Diagram, Object-Sequence Diagram

An activity diagram is frequently used in conjunction with a process flow diagram that describes the sequence and other attributes (i.e., timing) of the activities. A process flow diagram further captures the precedence and causality relations between situations and events. In object modeling, activity diagrams address the dynamic view of the system. They are especially important in modeling the function of a system and emphasize the flow of control among objects. An object diagram shows a set of objects (i.e., performers) and their relationships. Object diagrams represent static snapshots of instances of things found in class diagrams.

5.2.8 Data – Object Class Diagram

Class diagrams offer all the UML elements needed to produce entity-relationship diagrams. Class diagrams consist of classes, interfaces, collaborations, dependency, generalization, association, and realization relationships. The attributes of these classes can be expanded to include associations and cardinality [Booch, 1999]. In terms of support to DoDAF V1.5, classes that appear in an OV-7 (The DIV-3 in DoDAF V2.0) class diagram correlate to OV-3 information elements and OV-5 inputs and outputs. The OV-7 class diagram is a separate diagram from the class diagrams that may be developed for other products.

5.2.9 System (Component, Package, Deployment) Diagram

DoDAF V2.0 provides extensive architectural support for the SE process. As the process of developing the system architecture moves from the high-level concept (e.g., system interface description, system overview diagram) to more detailed views, it becomes useful to create multiple models so that specialized views ("Fit-for-Purpose") of the Architectural Description can be depicted. Three important diagrams (Fit-for-Purpose Views) are 1) the Component Model, which focuses on functional features of the system; 2) the Package Diagram, which focuses on grouping of components for specific purposes; and 3) the Deployment/Operational Model, which focuses on the physical runtime infrastructure on which functional components will be deployed.

The value of using multiple models arises from the fact that each of these models begins to call upon different skills and knowledge sets as the level of detail increases. Since these diagrams/ models are dependent upon each other, they cannot be created in complete isolation. The architecting process thus becomes an iterative process, defining the data for each portion, then evaluating how the data portion fits with other data portions, and making revisions that optimize the data. This can enable the generation of dependent diagrams which are accurate.

5.2.10 Component Model and Package Diagram

A Component Model, which can be a Systems Engineering Fit-for-Purpose View, describes the hierarchy of functional components, their responsibilities, static relationships, and the way components collaborate to deliver required functionality. For Section 7.2 only, a *component* is a relatively independent part of an IT System and is characterized by its responsibilities, and the interfaces it offers. Components can be decomposed into smaller components or aggregated into larger components. Some components already exist, but it may be necessary to build or buy others. A component can be a collection of classes, a program (e.g., one that performs event notification), a part of a product, or a hardware device with embedded functional characteristics (e.g., a Personal Digital Assistant [PDA]). Some are primarily concerned with data storage. A more comprehensive treatment of Component Models is found in the DoDAF Journal.

5.2.11 Deployment/Operational Model

The Operational Model, another potential Systems Engineering Fit-for-Purpose View, describes the operation of the IT system, as illustrated below in Figure 7.2.2-1. The Operational Model is derived primarily from the operational requirements placed on the e-business application. Like the Component Model, the Operational Model is typically developed through a series of progressively more detailed elaborations (i.e., Conceptual, Specified, and Physical). Also like the Component model, at each level of elaboration there may be a need to create more than one view of the Operational Model so that no single view becomes overloaded by attempting to convey too much information. A more comprehensive treatment of the Deployment/Operational Model is contained in the DoDAF Journal.

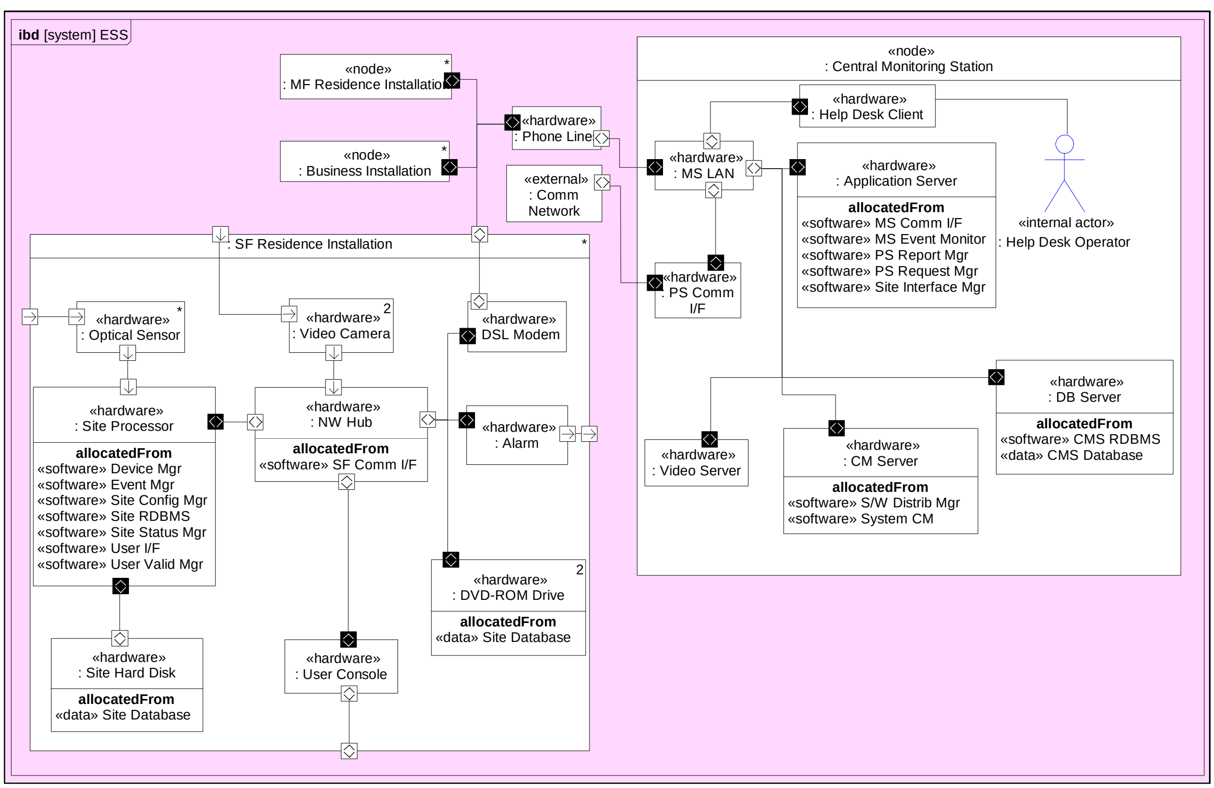


Figure 7.2.2-1: Deployment/Operational Model

5.3 C.A.R.P. and the DoDAF 6-Step Architecture Development Process

DoDAF v2.0 provides a high-level, six-step architecture development process. The six basic steps described by this process are:

1. Determine Intended Use of the Architecture
2. Determine Scope of 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

The C.A.R.P approach for AV-2 development fits neatly into this six-step architecture development process. This relationship is illustrated in the figure below. The important concept for all steps of this architecture development process is the establishment of an initial AV-2 up-front, which then drives the continual collection, recording, and reuse of a consistent harmonized vocabulary.

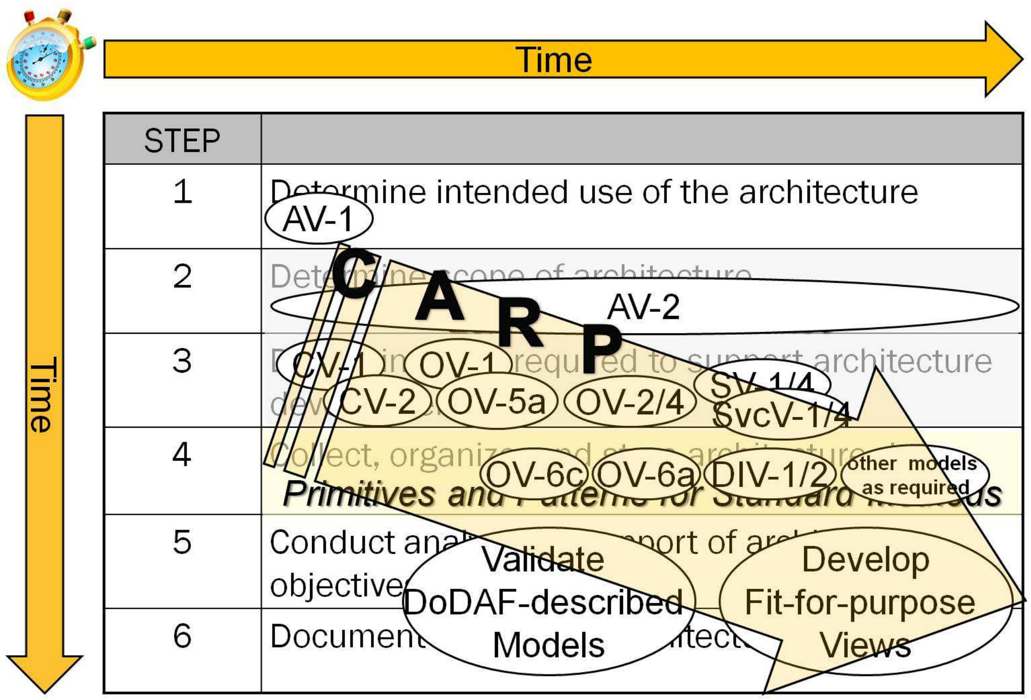


Figure 2: DoDAF Architecture Development Roadmap

5.3.1 Steps 1 and 2: Establish AV-2 Baseline

Steps 1 and 2 of the six-step process are the beginning activities for architecture development and characterize the intended use, purpose, and scope of the architecture effort. This information is generally provided by the architecture owner describing some aspect of their area of responsibility (process, activity, etc.) undergoing review, and is intended to insure the resulting architecture is "Fit for Purpose".

5.3.1.1 Start at the Beginning

6 Collection of glossary terms and definitions begins at Step 1 and architecture development process. As initial architecture data is appropriate scope of the architecture effort, vocabulary terms and harmonized and recorded in a consistent format in the AV-2 (See paragraph 1,

Transitioning from DoDAF 1.5 to DoDAF 2.0).

6.1.1 Steps 3 and 4: Proceed with C.A.R.P.

Steps 3 and 4 of the six-step process are the core activities in developing the architecture models and views, and thus produce the bulk of the terms and definitions required for the AV-2. Step 3 is a 'top-down' approach to data and vocabulary identification guided by controlled vocabularies within the C.A.R.P. method, while Step 4 is a more 'bottom-up' approach for data capture usually based on architecture methods and tools focused on development of specific DoDAF-described models. Architecture development typically iterates over these two steps. Terms and definitions recorded in the AV-2 should be related to elements of the DM2. In turn, these DM2 elements are associated with other architecture models that suggest additional data content to be collected and recorded.

6.1.1.1 Central Points in the DM2

7 Architects typically collect and organize data through the use of architecture models, e.g. activity, process, organization and data guided by controlled vocabulary terms and definitions that are consistent format in the AV-2 (See paragraph 1,

Transitioning from DoDAF 1.5 to DoDAF 2.0). The starting points are central key elements of the DM2 prescribed by the C.A.R.P. method as described in Section 2.1. Additional guidance for the AV-2 representation of these central DM2 elements is provided in Appendix D.

7.1.1 Steps 5 and 6: Validate and Iterate

Steps 5 and 6 of the six-step process test the architecture for completeness, accuracy, and sufficiency. Decision points related to including an architecture view, model, or even a term and definition are based on the intended use, purpose, and scope of the architecture effort determined in the first steps of development.

7.1.1.1 Appropriate Completeness and Coverage

Architectures that conform to DoDAF consist of multiple models, covering different aspects of the system that is being described. These descriptions reflect DM2 concepts that occur in one or more DoDAF-described models. Specific terms are defined in the AV-2 and classified according to the DM2 concept. The AV-2 should be assessed for adequate coverage of appropriate DM2 concepts and completeness against project requirements.

7.1.2 C.A.R.P. Process Guide Sheet

1. Define Capabilities	• *Identify overall objectives of the system.* • *What are the goals of the system?* • *What are the major design constraints?* • *What is the major functionality to be offered by the resulting system?*	AV-1 Overview and Summary Information: The capabilities identified in this step should occur in the AV-1 architectural description document. Initial basis for CV-1, CV-2, CV-3, CV-4. Can be used later on to define CV-5, CV-6, and CV-7.
2. Define Activities	• *Identify the major processes of the system that are needed to provide the desired capabilities.* • *Break the major processes into those activities necessary to achieve the objectives of each process.* • *Describe Activities in "Verb-Object" format (e.g.: write report).* • *Avoid unspecific verbs such as "manage" or "oversee."* • *Do not use "and" in activity labels: Break complex activities into individual steps.*	CV-6: Linkage between Activities and the Capabilities that they support OV-5a: Operational Activity Decomposition Tree The results of this step become the activities in a hierarchical functional decomposition diagram OV-6c Event-Trace Description: The results of this step become the activities in an eventual process model Constraints among the activities can be used as the basis for OV-6a (Operational Business Rules)
3. Define Resources	• *Identify the major objects and data elements (entities) of the system.* • *Identify the relationships*	DIV-1/2: Data Model The results of this step become classes/tables in an eventual conceptual data model, which

	among the resources (Structural Business Rules).	forms the basis for DIV-1 and DIV-2 DoDAF – described models. OV-2/OV-3: Operational Resource Flow
4. Define Performers	• *Revisit the list of resources identified in step 2 and identify those that actively contribute toward the completion of activities or the achievement of an objective.*	OV-4: Organizational Relationship Chart OV-6c Event-Trace Description: The result of this step defines the swimlanes in an eventual process model. S9vc)V-4 System (Service) Functionality Description

8 DoDAF Product Development Guidance and Best Practices

8.1 Planning for Architecture Development

Planning an Architectural Description effort involves more than selection of a method for development. The Architectural Description effort starts with the identification of a requirement, problem, or desired change by the process owner – the senior official responsible for the overall operation of the functional, tactical, component or JCA. The process owner selects a team leader and team members who will actively participate in the Architectural Description effort. That team may have a varying membership, generally including an enterprise architect, and subject matter experts in the process area undergoing analysis and potential change, and will refine the process owner's vision and/or initial requirement into a project through development of an appropriate Architectural Description, as shown in the steps in Section 6.1.1, and in Section 10, Architecture Planning.

Managers and decision-makers are generally not technicians or information architects. They do, however, have a vital part in the decisions that need to be made early in the planning process to define the types of views they need to support their involvement in the decision-making process. Organizations differ in the type of presentation materials they prefer (i.e., dashboards, charts, tables) and these preferences need to be accommodated during Architectural Description development. Toolsets should be selected that have the capability to provide these management views and products, along with the ability to collect and organize data consistent with the DM2 to facilitate reuse. A detailed discussion of toolset requirements and capabilities is contained in the DoDAF Journal.

8.2 Architecture Lifecycle and Architecture Governance

Architectural Description development is only one phase of an overall architecture lifecycle, similar to other process maturity and change lifecycles. One such lifecycle, the Architecture Governance, Implementation, and Maturity Cycle, shown in **Figure 7.1.3-1** below, is described in detail in the DoDAF Journal. This lifecycle relies on the commonly used Plan-Do-Check-Act (PDCA) governance method.

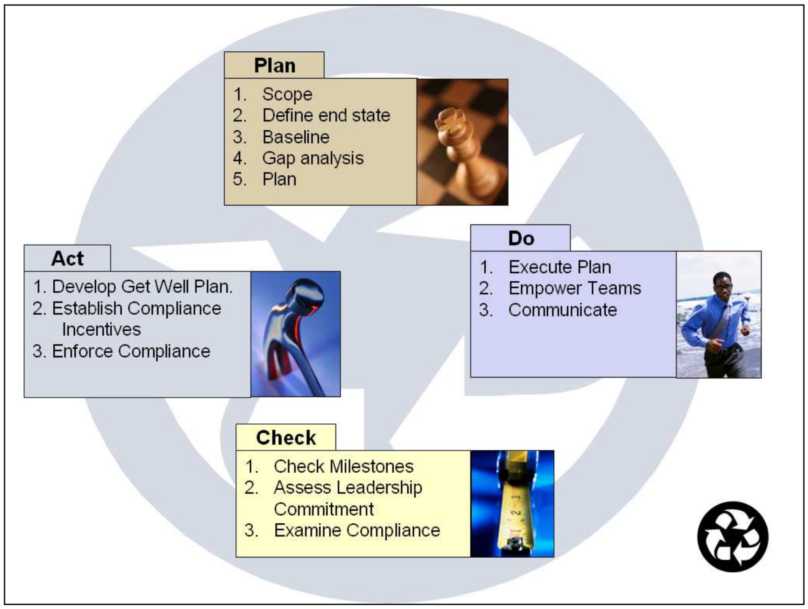


Figure 7.1.3-1: Plan, Do, Check, and Act (PDCA) Cycle

8.3 Developing, Maintaining and Managing Architectures

8.3.1 Developing Architectures

Careful scoping and organization by managers of the architecture development effort focuses on areas of change indicated by policy or contract in support of the stated goals and objectives. A data-centric, rather than product-centric, architecture framework ensures concordance across architectural views (i.e., that data in one view is the same in another view when talking about the same exact thing , such as an activity), enables the federation of all pertinent architecture information, and provides full referential integrity (that data in one view is the same in another view when talking about the same exact thing , such as an activity) through the underlying data to support a wide variety of analysis tasks. Logical consistency of the data thus becomes a critical 'property' of architectures of all types as described more fully below. The objective of achieving concordance across the architectural view must be included in architecture planning and development actions.

DoDAF V2.0 describes two major types of architectures that contribute to the DoD Enterprise Architecture, the **Enterprise-level architecture** and the **Solution Architecture**. Each of these architectures serves a specific purpose, as described briefly below, and in more detail in Section 4 of Volume I:

g. Enterprise Architectures: A strategic information asset base, which defines the mission, the information necessary to perform the mission, the technologies necessary to perform the mission, and the transitional processes for implementing new technologies in response to changing mission needs. EA includes a baseline architecture, a target architecture, and a sequencing plan.5

[5] *Office of Management and Budget (OMB) Circular-A-130, Management of Federal Information Resources*, February 8, 1996. Executive Office of the President, Office of Management and Budget. The current version can be found at: http://www.whitehouse.gov/omb/circulars/a130/a130trans4.html#2

h. Solution Architectures: A framework or structure that portrays the relationships among all the elements of something that answers a problem.6 This architecture type is not a part of the DoD Enterprise Architecture, but is used to define a particular project to create, update, revise, or delete established activities in the Department. Solution architecture may be developed to update or extend one or more of the other architecture types. A Solution Architecture is the most common type of architecture developed in the Department. Solution architectures include, but are not limited to, those SOA-based architectures developed in support of specific data and other services solutions.

Instances of Enterprise Architectures include Capability, Segment, Mission Thread, and Strategic Architectures. They are not types of Architecture.

Version 1.0 and 1.5 of the DoDAF used the term 'product' or 'products' to describe the visualizations of architecture data. In this volume, the term 'DoDAF-described Model' is generally used, unless there is a specific reference to the products of earlier versions. For DoDAF-described Models that have been populated or created with architectural data, the term 'Views' is used. The term "Fit-for-Purpose Views" is used when DoDAF described models are customized or combined for the decision-maker's need.

The Models described in DoDAF, including those that are legacy views from previous versions of the Framework, are provided as pre-defined examples that can be used when developing presentations of architecture data. DoDAF does not prescribe any particular models, but instead concentrates on data as the necessary ingredient for architecture development. If an activity model is created, a necessary set of data for the activity model is required. Key process owners will decide what architectural data is required, generally through DoDAF-described Models or Fit-for-Purpose Views. However, other regulations and instructions from both DoD and CJCS have particular presentation view requirements. These views are supported by DoDAF V2.0, and should be consulted for specific view requirements. The architectural data described in DoDAF V2.0 can support many model and view requirements and the regulations and instructions should be consulted for specific model and view requirements.

In general, architecture data and derived information can be collected, organized, and stored by a wide range of tools developed by commercial sources. Creation of various views using these architecture tools is the typical way an enterprise architect initially captures and represents important architectural data.

Both DoDAF-described Models and Fit-for-Purpose Views (e.g., dashboards, composite, or fusion presentations) created as a part of the architecture development process, which visually render the underlying architectural data, act to facilitate decisions.

8.3.2 Maintaining and Managing Architectures

Embedding architecture development process in routine planning and decision-making institutionalizes the practice and makes the maintenance of architectural data, views, and viewpoints more automatic. Architectures are maintained and managed within the Department

[6] Derived from Joint Pub 1-02 and Merriam-Webster.com.

through *tiered accountability*. Tiered accountability is the distribution of authority and responsibility for development, maintenance, CM, and reporting of architectures, architecture policy, tools, and related architecture artifacts to all four distinct tiers within the DoD. DoDAF V2.0 supports four tiers: Department, JCA, Component, and Solution (i.e., program or project-level solutions development). These tiers support the federated approach for architecture development and maintenance.

8.4 DoDAF Development Guidelines

DoDAF v2.0 provides comprehensive and practical guidance for the creation of Architectural Descriptions that provide added value for decision-making at the level of the DoD they are produced. To this end, the framework offers guiding principles in the development of Architectural Descriptions that transcend the tier, level, or purpose of the architecture development, and a logical method for executing the development of Architectural Descriptions for supporting critical decisions within key DoD management and change management processes. The Framework also offers flexibility in approach, toolset utilization, and techniques such as structured analysis, object-oriented, and service-oriented.

8.4.1 Guiding Principles

Guiding principles are high-level concepts, which provide a general roadmap for success in developing Architectural Descriptions under DoDAF v2.0. The principles are:

a) Architectural Descriptions should clearly support the stated objective(s) ("Fit-for-Purpose"). The framework offers general direction in the development of Architectural Descriptions so that they can support critical decisions within key DoD management and change management processes. While DoDAF v2.0 describes a number of models, based on collected data, diligent scoping of a project and any guiding regulations, instructions, or standard procedures will determine the specific visualization requirements for a particular architectural effort.

b) Architectural Descriptions should be simple and straightforward, but still achieve their stated purpose. Architectural descriptions should reflect the level of complexity defined by the purpose for their creation. Scoping of a project, as described in Section 7.0 Methodologies, will ensure that the resulting architectural data and derived information, and the views created are consistent with their original purpose.

c) Architectural Descriptions should facilitate, not impede, communications in decision processes and execution. Creation of Architectural Descriptions is meant to support decision processes and facilitate improvement of procedures and/or technology in the enterprise. Collection of architectural data and creation of views supports the decision-making process, and provides a record to explain critical choices to technical and non-technical managerial staff.

d) Architectural Descriptions should be relatable, comparable, and capable of facilitating cross-architecture analysis. Most Architectural Descriptions, except perhaps those at the highest levels of DoD or an organization, relate on their boundaries to other external processes and operations. When several processes and/or operations are evaluated, compared, or cross-referenced, it should be clear how, where, and why data passes among them in similar form.

e) Architectural Descriptions should articulate how data interoperability is achieved among federated Architectural Descriptions. To enable federation, the framework will provide structures to ensure that horizontal touch-points can be compared for consistency across Architectural Description boundaries. Other mechanisms will ensure that higher tiers have access to data from lower tiers in a form that supports their decision needs. DoDAF utilizes the DM2, and particularly the PES described in Volume III, as a resource for interoperability. A key element in ensuring interoperability is the effort taken to plan for integration of data across views, Architectural Description boundaries, and is consistent between tiers.
f) Architectural Descriptions should be data centric and tool-agnostic. The framework assists in the design of structures that meet specific needs depending on the priorities of individual organizations. In particular, the framework calls for the development of integrated, searchable, structured architectural data sets that support analysis targeted to critical decisions. To that end, multiple toolsets, with varying internal rules, techniques, notations, and methods may be used, consistent with the PES.

g) Architectural data should be organized, reusable, and decomposed sufficiently for use by architectural development teams and decision support analysis teams. Collecting and organizing architectural data for use in decision processes should not be ‘over done’, that is the depth and breadth of data collected should be sufficient to capture the major processes actions, and not be so broad that the original intent of the architecture project becomes clouded. Whenever possible, data common to other Architectural Descriptions should be used. New data should be created utilizing the structures described in Volumes 2 and 3 so that, when stored in the DoD Metadata Registry (DMR), it becomes discoverable to others with similar requirements.

h) Development of Architectural Descriptions should be guided by the principles and practices of net-centricity to facilitate and support the Net-Centric Strategy of the Department. Development of Architectural Descriptions should ensure that Architectural Descriptions are developed adhere to net-centric principles, as outlined in the Net-Centric Strategy, and clearly delineate data that must be shared across and between systems or services described in the Architectural Description.

NOTE: It is recognized that not all Architectural Descriptions or architectural data developed by DoD are related to net-centric operations or net-centricity; however, for the majority of Architectural Descriptions developed under the DoDAF, net-centricity is a critical design consideration.

Architectural guiding principles enable and facilitate validation and verification activities that will determine the success of the project, and the ability of the resulting Architectural Descriptions to serve the purpose for which it was created. Guiding principles support the more specific goals and objectives of a project as a roadmap.

8.4.2 Multiple Techniques and Toolsets, Including Structured and Object Oriented Analysis

The framework allows architects to select techniques and toolsets to meet specific needs. While the framework provides examples of the application of both Structured Analysis and Design (SADT) and Object-Oriented Analysis & Design (OOAD) techniques, it mandates neither. The framework explicitly permits any technique that meets the needs of the organization, provides the appropriate architectural data, adheres to the architectural data requirements of parent tiers described further in Section 3, and is capable of producing data that can be shared in a federated environment. A brief section on essential toolset attributes desirable for creation of Architectural Descriptions utilizing DoDAF are contained below in Section 3.5.3.

8.4.3 Essential Toolset Attributes

While DoDAF is toolset agnostic, allowing architects, and Architectural Description development teams to utilize any toolset they desire to create Architectural Descriptions, there are some basic attributes of a toolset needed to ensure that Architectural Descriptions, once registered, are discoverable, sharable, and their data useful to others with similar or derived needs in their own Architectural Description development. These attributes are:

1. Capable of utilizing the PES described in Volume III to collect, organize, store, and share architectural data.

2. Capable of eXtensible Markup Language (XML) data transfer to/from the DMR, and other resources, such as the DoD Architecture Registry System (DARS) for registering architectural data.

8.4.4 Architecture Resources

A number of architecture resources exist which serve as sources for guidelines that should be consulted while building architectural views. Some of these architecture resources are briefly described Table 5.1-1, with their architectural uses, and their URLs. Additional information is contained in the individual URLs. Some architecture resources require Secret Internet Protocol Router Network (SIPRNET) access.

Table 8.4-1: Architecture Resources

Resource	Description	Architecture Use	URL
Department of Defense Information Enterprise Architecture (DoD IEA)	Defines the key principles, rules, constraints and best practices to which applicable DoD programs, regardless of Component or portfolio, must adhere in order to enable agile, collaborative net-centric operations.	The DoD IEA provides the guidelines and rules that the architect must keep in mind in the architecture development effort.	http://www.defenselink.mil/cio-nii/cio/diea/

Resource	Description	Architecture Use	URL
DoD Architecture Registry System (DARS)	DARS is the DoD registry and repository of segment and solution architectures comprising the federated DoD enterprise architecture.	To discover architectures that exist, or may be in development. Depending on the purpose and scope, an architect may search and discover Architectures that overlap the scope and purpose of the architecture effort. To register metadata about architectures that are being developed, or currently exist.	https://dars1.army.mil
DoD Information Technology Portfolio Repository (DITPR)	The official unclassified DoD data source for Federal Information Security Management Act (FISMA), E-Authentication, Portfolio Management, Privacy Impact Assessments, the inventory of MC/ME/MS systems, and the registry for systems under DoDI 5000.2.	The Systems metadata from the Architecture can be used to populate DITPR with new or updated information. DITPR can also populate the architecture's Systems metadata, particularly on systems that interface with systems described in the architecture, but are not part of the scope of the architecture.	https://www.dadms.navy.mil/
DoD Information Technology Standards and Profile Registry (DISR)	Online repository for a minimal set of primarily commercial IT standards.	The DISR can be used to populate the Standards models (StdV-1 and StdV-2) of the Architecture. Conversely, the Standards Models can identify additional or new standards that need to be added to DISR.	https://disronline.disa.mil

Resource	Description	Architecture Use	URL
Joint C4I Program Assessment Tool (JCPAT)	Formally assess systems and capabilities documents (Initial Capabilities Document, Capability Development Document, and Capability Production Document) for Joint Staff interoperability requirements certification and is the ITS/NSS Lifecycle Repository and the archives.	The ICD, CDD, and CPD contain architecture information. As the architecture development progresses, the collected architecture information can be extracted and reported in the ICD, CDD, and the CPD. In addition, the architecture information can be within with the Enhanced-Information Support Plan (E-ISP) tool, a part of the JCPAT toolset.	http://jcpat.ncr.disa.smil.mil/JECOweb.nsf
Joint Common System Function List (JCSFL)	A common lexicon of systems/service functionality supporting joint capability. The JCSFL is provided for mapping functions to supported activities and the systems or services that host them. Chairman of the Joint Chiefs of Staff Instruction (CJCSI) 6212.01E prescribes the JCSFL for use in developing a common vocabulary for architecture development.	Use the taxonomy to align or extend system functions within the architecture being developed	https://us.ar.y.mil/suite/page/419489
Knowledge Management/ Decision Support (KM/DS)	The KM/DS tool will be used by DoD components to submit documents and comments for O-6 and flag reviews, search for historical information, and track the status of documents.	Supporting the JCIDS approval process, the documents that are necessary for Milestone Decisions have architecture information. As the architecture development progresses, the collected architecture information can be extracted and reported in the required documents.	https://jrockmds1.js.smil.mil/guestjrcz/gbase.guesthom.

Resource	Description	Architecture Use	URL
Metadata Registry	The DoD Metadata Registry and Clearinghouse provides software developers access to data technologies to support DoD mission applications. Through the Metadata Registry and Clearinghouse, software developers can access registered XML data and metadata components, database segments, and reference data tables and related metadata information	The Resource Flows and Physical Schemas from the Architecture can be used to populate the Metadata Registry.	http://metadata.dod.mil
Naval Architecture Elements Reference Guide (NAERG)	A standard terms of reference for the Navy and Marine Corps. The Architecture Elements represent the critical taxonomies requiring concurrence and standardization for an integrated architecture. They comprise the lexicon for the three views of the architecture framework, the operational (OV), system (SV) and technical standards (TV) views.	The use of the critical taxonomies is a step to ensuring integration of systems within a system of systems and alignment of information technology (IT) functionality to mission and operational needs. The data contained in each element of the Architecture list shall be used for overall architecture framework development, programmatic research, development, and acquisition activities, and related integration and interoperability and capability assessments. It will be updated through review periods to support DoN Program Objective Memorandum (POM) efforts and to reflect changes mandated by DoD, technology improvements, and other factors.	https://stalwart.spawar.navy.mil/naerg/

Resource	Description	Architecture Use	URL
Service Registry	The Service Registry provides enterprise-wide insight, control and leverage of an organization's services. It captures service descriptions and makes them discoverable from a centrally managed, reliable, and searchable location.	The Services metadata from the Architecture effort can be used to populate the Service Registry in the process of developing the solution.	*http://metadata.dod.mil*, Select the "NCES Service Discovery" button
Universal Joint Task List (UJTL)	The Universal Joint Task List from the Chairman of the Joint Chiefs of Staff Manual 3500.04C (CJCSM) serves as a common language and common reference system for joint force commanders, combat support agencies, operational planners, combat developers, and trainers to communicate mission requirements. It is the basic language for development of a joint mission essential task list (JMETL) or agency mission essential task list (AMETL) that identifies required capabilities for mission success.	Use the taxonomy to align or extend operational activities within the architecture being developed.	http://www.dtic.mil/doctrine/jel/cjcsd/cjcsm/m350004c.pdf

8.5 Addressing Security Issues in DoDAF-Conformant Architecture Development

Security continues to be a critical concern within the DoD, and Architectural Description development efforts at any level need to ensure that appropriate security concerns are addressed clearly, so that any decisions made that rely on the Architectural Descriptions are valid and useful. Security concerns are routinely addressed through the risk assessment process described in Section 10 of Volume I, and Appendix C of Volume II.

Each of the individual models described in detail in Volume II provides the architect and development team with a set of data for collecting, documenting, and maintaining security data. These data support physical, procedural, communications security (COMSEC), Transient Electromagnetic Pulse Emanation Standard (TEMPEST), and Information Security (INFOSEC) concerns. DM2 incorporates the Intelligence Community Information Security Marking (IC ISM) standard for classification markings of architecture information.

Capabilities are subject to a variety of threats to the integrity, availability, and confidentiality of their operation. These threats range from failures of equipment, attempts to gain unauthorized access to their services and data, to sabotage of their functions. Security engineering is concerned with identifying the potential threats to a capability, and then, using a risk management approach, devising a set of measures which reduce the known and potential vulnerabilities to an acceptable level. In general, the measures that can be applied fall into the following categories:

- **Physical –** measures such as guards, guard dogs, fences, locks, sensors, including Closed Circuit Television, strong rooms, armor, weapons systems, etc.
- **Procedural –** the specification of procedures, including vetting (which tests that personnel have a sufficient level of integrity and trust to be given responsibility to access and use a capability's services and data) that will reduce the likelihood of vulnerabilities being exploited.
- **Communication Security (COMSEC) –** using encryption and other techniques to ensure that data transmission is available at sufficient bandwidth, that the traffic pattern and content of data in transit are indecipherable to a third party who might intercept the data, and that its integrity is protected.
- **Transient Electromagnetic Pulse Emanation Standard (TEMPEST) –** measures to ensure that the electromagnetic transmissions from equipment can't be intercepted to derive information about the equipment's operation and the data it processes.
- **Information Security (INFOSEC) –** ensuring the integrity, availability and confidentiality of data and IT-based services.

In general, the measures employed to protect a capability will have undesirable impacts on all of the capability's lines of development, and in particular on it's deploy ability, usability and procurement and maintenance costs. It is therefore desirable to minimize the strength of the measures to be employed in a fashion commensurate with the value of the assets being protected. This requires a risk-managed approach based on the assessment of the likely threats posed to the asset. A risk assessment approach considers the following characteristics:

- Environment – The level of hostility of the environment the asset is being deployed to.
- Asset Value – this is denoted by a protective marking which indicates the impact of the loss or disclosure of the asset would have on the effective operation of the government and its departments of state.
- Criticality – an assessment of the criticality of the asset to enabling the government to undertake its activities.
- Personnel Clearance – a measure of the degree of trust that the government is willing to put in the personnel that will have (direct or indirect) access to the asset.

The aim of this guidance for representing security considerations is to enable sufficient information to be recorded for interested parties (accreditors, security advisors, users, system managers) to understand the potential security exposure of capabilities so that security can be managed effectively throughout the life of a capability.

The **Table C-1** below shows the DoDAF scheme for assigning security characteristics and protective measures to elements of DoDAF. There is not a specific security viewpoint in DoDAF; security information can be shown on models using annotations and call–outs. The

DoDAF Meta-Model contains the concepts, associations, and attributes for capturing and representing security characteristics in a consistent way between models. Table B-1, DM2 Concepts, Associations, and Attributes Mapping to DoDAF-described Models indicates the security elements within the DM2.

Table C-1: DoDAF Viewpoints and Concept Mapped to Security Characteristics and Protective Measures

Viewpoint	Concept	Security Characteristics	Protective Measures	Notes
Capability	Capability requirement	Security Marking Criticality Environment User Security Profile		The security characteristics of capability requirements provide the security envelope for the capability for a particular timeframe.
Operational	Location	User Security Profile Environment		The User Security Profile is the lowest clearance of the users within a location, facility, or organization. The environment identifies the most hostile conditions for the location, facility, or organization.
	Operational Activity	Security Marking Criticality		The security marking identifies the highest security marking of information that will be processed by a Operational Activity and the Criticality measures the impact on government operations with the disruption of the operational activity.
	Resource Flow	Security Marking		The security marking identifies the highest security marking that will be exchanged in a Resource Flow.

Table C-1: DoDAF Viewpoints and Concept Mapped to Security Characteristics and Protective Measures

Viewpoint	Concept	Security Characteristics	Protective Measures	Notes
	Organization	User Security Profile Environment		The minimum clearances of members of the organization, post, base, fort.
System	Capability Taxonomy	Security Marking Criticality Environment User Security Profile		The security characteristics of a capability taxonomy are to be derived from the constituent systems.
	System	Security Marking Criticality Environment User Security Profile	Physical TEMPEST COMSEC	The environment of a system is derived from the Physical Asset to which is deployed. The User Security Profile is derived from the Organization which uses the system, its Criticality and Security Marking from its Functions.
	Physical Asset	Environment	Physical TEMPEST	The environment identifies the worst environment to which the Physical Asset will be deployed.
	Function	Security Marking Criticality	INFOSEC Procedural	The Security Marking identifies the maximum security marking of the data the Function will process and the criticality represents the degree of harm to government operations if disrupted.

Table C-1: DoDAF Viewpoints and Concept Mapped to Security Characteristics and Protective Measures

Viewpoint	Concept	Security Characteristics	Protective Measures	Notes
	System Resource Flow	Security Marking	COMSEC	The Security Marking represents the maximum security marking of the Resource Flow.
	Performer and Function	User Security Profile	Procedural	The User Security Profile is the lowest clearance of the user performing the function. This should be derived from Organizations who perform the Function, if the information exists.
Service	Capability Taxonomy	Security Marking Criticality Environment User Security Profile		The security characteristics of a capability taxonomy are to be derived from the constituent services.
	Service	Security Marking Criticality Environment User Security Profile	Physical TEMPEST COMSEC	The environment of a service is derived from the Physical Asset to which is deployed. The User Security Profile is derived from the Organization which uses the service, its Criticality and Security Marking from its Functions.
	Physical Asset	Environment	Physical TEMPEST	The environment identifies the worst environment to which the Physical Asset will be deployed.

Table C-1: DoDAF Viewpoints and Concept Mapped to Security Characteristics and Protective Measures

Viewpoint	Concept	Security Characteristics	Protective Measures	Notes
	Function	Security Marking Criticality	INFOSEC Procedural	The Security Marking identifies the maximum security marking of the data the Function will process and the criticality represents the degree of harm to government operations if disrupted.
	System Resource Flow	Security Marking	COMSEC	The Security Marking represents the maximum security marking of the Resource Flow.
	Performer and Function	User Security Profile	Procedural	The User Security Profile is the lowest clearance of the user performing the function. This should be derived from Organizations who perform the Function, if the information exists.
Standards	Performer	Security Marking	INFOSEC Procedural	The Security Marking identifies the security standard for the data the Function will process and the criticality represents the degree of harm to government operations if there is unauthorized access.

8.6 AV-2 Guidance

8.6.1 Architectures and the DoDAF Meta Model (DM2)

Data-Centric architectures intend to provide an integrated base of architecture data organized to effectively provide information to users in support of query, analysis, decision making, and business intelligence. Data-Centric architectures represent a fundamental shift from traditional architecture approaches focused on separate standard presentations of different data in different formats intended for different stakeholders.

In DoDAF V2.0, "The major emphasis on architecture development has changed from a product-centric process to a data-centric process designed to provide decision-making data organized as information for the manager."

- "DoDAF V2.0 focuses on architectural "data", rather than on developing individual "products" as described in previous versions."
- "Products have been replaced by views that represent specific types of presentation for architectural data and derived information."
- "Visualizing architectural data is accomplished through models (e.g., the products described in previous versions of DoDAF)."
- "When data is collected and presented as a "filled-in" model, the result is called a view."
- "Fit-for-Purpose Views are user-defined views of a subset of architectural data created for some specific purpose (i.e., "Fit-for-Purpose")."

DoDAF V2.0 defines an underlying DoDAF Meta Model (DM2) of concepts, attribute and associations. In DoDAF V2.0, "The DM2 provides a high-level view of the data normally collected, organized, and maintained in an Architectural Description effort." that is meant to "Establish and define the constrained vocabulary for description and discourse about DoDAF models (formerly "products") and their usage in the 6 core processes".

A controlled vocabulary drives development of integrated and federated architectures. By focusing on a core set of common concepts across all architectures, data-centric architectures encourage use of a common term for the same concept. This results in integrated architecture where the same concept has the same name and definition across the multiple DoDAF-described and "fit for purpose" models and views.

8.6.2 Capturing Architecture Data

In DoDAF V2.0, "In general, data can be collected, organized, and stored by a wide range of architecture tools developed by commercial sources." Many different methods may be used to capture architecture data. Ultimately this data becomes labels for rendering core DM2 concepts within a particular architecture method/technique. The same DM2 concept could appear differently in different architectural methods/techniques. See Table 2-1.

Table 8-1: Different Architecture Methods and Data

	Example Architecture Methods

AV-2 Term	DoDAF v2.0 Meta Model Concept	Organization Hierarchy	Activity Decomposition	BPMN Process Model	UML Class Diagram	…
Intermediate C2	Performer	Organization	N/A	Swimlane	Class	…
Coordinate CAS	Activity	N/A	Node	Task	N/A	…

In this example the term 'Intermediate C2' is an instance of the DM2 concept 'performer' depicted as an organization in an organization hierarchy (OV-4), a swimlane in a BPMN process model (OV-6c) and a class in a UML Class Diagram (DIV-1). The term 'Coordinate CAS', an instance of the DM2 concept 'activity', is rendered as a task in a BPMN process model (OV-6c) and a node in an Operational Activity Decomposition (OV-5a).

8.6.3 Guidelines for DoDAF AV-2 Design and Development

The DoDAF V2.0 AV-2 Integrated Dictionary is defined as "An architectural data repository with definitions of all terms used throughout the architectural data and presentations." Its purpose is to serve as a common vocabulary and consistent terminology reference for DoDAF described models and views, derived 'fit for purpose' views and other architectures. The AV-2 provides clear definitions for specific terms used in the specific types of presentation views for architecture model developers and users.

An initial version of the AV-2 Integrated Dictionary should be developed at the beginning of any DoDAF architecture project to gain clarity over the scope, objectives and constraints of the architecture and to precisely define key terms. This provides a baseline to be refined and expanded in an iterative fashion throughout the architecture development process. The end result is a controlled vocabulary harmonized across the architecture that drives development of the various DoDAF described and 'fit for purpose' views.

AV-2 terms should be related to the architecture concept they represent in the DoDAF Meta Model (DM2). Linking a term to its DM2 concept enables users to identify which DoDAF-described Models may (or should) contain reference to this term, and supports discovery and re-use of common terms rather than inventing another synonym.

A vocabulary-driven DM2-based approach to development of DoDAF V2.0 Architectures implies that:

- AV-2 definition leads the architecture development effort, providing a clear, common, controlled vocabulary for architects to use as they develop the architecture models and views.
- The AV-2 can provide a validation instrument for the architecture based on explicit relationships in the AV-2 between defined terms and DM2 concepts. DoDAF V2.0 provides a mapping of DM2 concepts to DoDAF-described models. Linking this mapping to the AV-2 can support assessment of architecture coverage and completeness.

- AV-2 terms and relationships should be stored in a format and repository with data management capabilities that allows persistency and linkage for future reference and reuse, and supports reasoning over the terms, definitions and relationships among concepts.

The current state of AV-2 construction in DoDAF development results in architectures that often require additional effort to integrate the various models and are difficult to federate with other DoDAF architectures:

- The AV-2 is typically a derived view that does not lead development efforts. The AV-2 dictionary is commonly derived from existing architecture models and typically generated 'after the fact', i.e. after other architecture views are completed. This implies that architects must harmonize and merge different vocabularies from different DoDAF models.
- The identification of individual terms in an AV-2 with the concepts they represent in the DM2 is typically not explicit. This implies that an architect cannot analyze the coverage or check the completeness of architecture concepts against a list of mandatory views of the architecture or discover linkages between architecture models.
- The AV-2 is typically a simple table structure that neglects data management and reasoning capabilities provided by other representations, such as cross-referencing of terms, extensions of pre-populated views, and reuse of common definitions. This also means that a user cannot browse or reason about relationships among terms, and increases the risk that terminology conflicts go unnoticed and create ambiguities and inaccuracies in the resulting architecture.

8.6.4 The C.A.R.P. Method: Central Points in the DM2

The C.A.R.P method is intended to produce a baseline controlled vocabulary focused on key concepts central to any DoDAF architecture. The initial AV-2 documents a baseline to be refined and expanded in an iterative fashion throughout the architecture development process. See also Appendix A and C.

To support the major objective of architecture alignment and federation, core AV-2 content for any DoDAF architecture should be focused on a central common foundation of concepts relevant to any domain. Starting points corresponding to central key concepts defined in the DoDAF V2.0 DM2:

- Capability ("why"): "The ability to achieve a desired effect under specified standards of performance and specified conditions through combinations of ways (guidance and rules) and means (resources) to perform a specified set of activities." Capabilities describe the desired functionality an architecture is defined to support. This may correspond to high-level value streams, objectives, and goals of an organization. In DoDAF V2.0, "A capability is distinguished from other collections of activities and resources by (a) the explicit presence of a performer who is capable of responsibility and who envisions a desired effect, (b) explicit statement and measures of such desired effects, (c) and explicit consideration of conditions under which activities entailed by a capability may be successfully carried out.", and so a fully described capability requires additional related descriptions of C.A.R.P. concepts, and may also include additional DM2 concepts at more detailed levels of description.
- Activity ("how"): "Work, not specific to a single organization, weapon system or individual that transforms inputs (Resources) into outputs (Resources) or changes their state. " Activities transform resources to achieve an objective/provide a capability. Activities at various levels of detail describe the processes, sub-processes, and tasks carried out to actively support a capability to realize a desired effect. All exchanges and flows of resources are due to producing

or consuming activities. Resource flows are activity-based, not performer-based, because a performer cannot produce or consume a resource other than by carrying out an activity. That is, a performer can only give or get a resource by carrying out an activity.

- Resource ("what"): "Data, Information, Performers, Materiel, or Personnel Types that are produced or consumed." Resources are consumed, transformed and produced by activities in order to do work. By definition, resources may be types of equipment, apparatus or supplies (Material), Information, or more specifically the representation of information in a formalized manner suitable for communication, interpretation, or processing (Data). Note that a resource may also be a type of geospatial extent whose boundaries are by declaration or agreement by political parties – i.e. GeoPoliticalExtentType.
- Performer ("who"): "Any entity - human, automated, or any aggregation of human and/or automated - that performs an activity and provides a capability." A performer may be answerable or accountable for the action it performs (PerformerCapableOfResponsibility), specifically:
    - A person in a role (PersonRole) such as the roles described by the Amy's military occupational specialties (MOS) , a specific person role (IndividualPersonRole) such as a particular MOS at a particular billet, a type of organization (OrganizationType) or a specific organization (Organization) that has a mission.

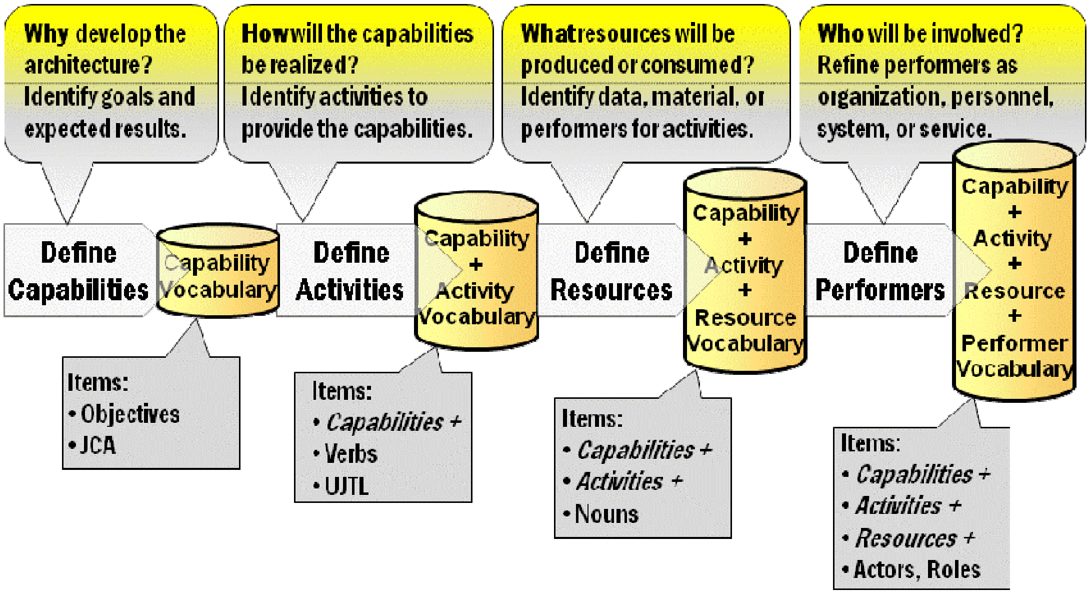
  A Performer may also be a type of system (System) in the general sense of any assemblage of components—machine and human—that accomplish a function, or a type of service (Service) including software services and business services. In DoDAF V2.0, "The performance of an activity by a performer occurs in physical space and time. That is, at some place and time, the activity is conducted. This is referred to as a spatial-temporal overlap, simply meaning that the activity and performer overlap in space and time. There are two ways in which a performer spatial-temporally overlaps an activity:

    - In the act of performing the activity. This relationship is sometimes called assignedTo for the purposes of traceability.
    - The other way is as part of a larger process (aggregated activities). This is sometimes called allocatedTo and forms the initial stages of system or activity decomposition. Allocated performers are assigned to activities in the initial stages of defining performers."

  Note that performers not only perform activities, but are also a type of resource that may be produced, consumed, or transformed by an activity.

These central concepts in the DM2 model suggest a logical, incremental approach to developing an initial AV-2 integrated dictionary, as illustrated in Figure 2-1. As discussed, this is an iterative process that should be initially exercised at the beginning of any DoDAF architecture project to provide a baseline, then continually refined and expanded in an iterative fashion throughout the architecture development process.

Figure 8.6-1: C.A.R.P. in the DM2



Since DoDAF prescribes a mapping from DM2 elements to architecture models, an initial set of identified DM2 elements suggests additional architecture models relevant to those concepts that the architect may develop for more comprehensive data collection using some formal modeling method.

8.6.5 Implications for Other DoDAF Described Models

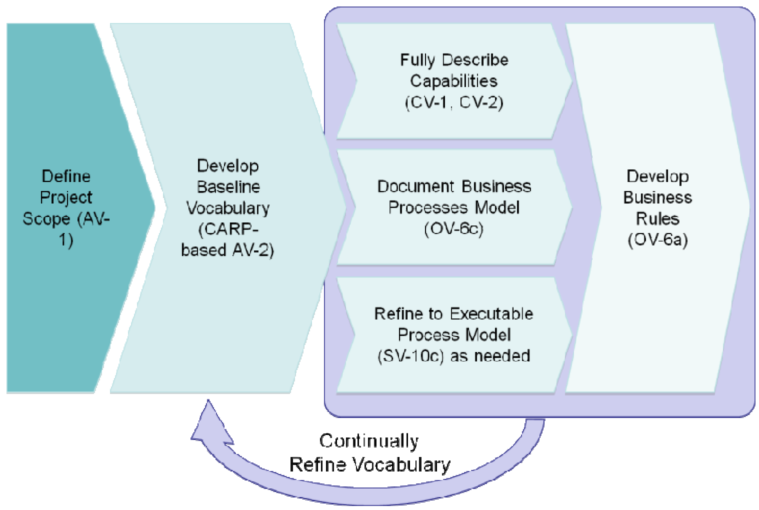
The content of C.A.R.P. maps directly to and influences several core DoDAF-described models:

- A *Capability* is represented in the DoDAF V2.0 CV-1 Capability Effects model which "identifies and describes effects caused by capabilities within a described architecture and specifies measures for these effects. This model emphasizes the desired effects of capabilities provided by a described architecture." A set of related *Capabilities* is essentially a DoDAF V2.0 CV-2 Capability Hierarchies model, which "emphasizes relations among capabilities and among the parts of capabilities." Note that a CV-2 model may show the parts that make up a whole capability or capabilities that are subtypes of other capabilities. CV-2 'whole-part' decompositions are typically more useful to support acquisition planning and investment review.
- A vocabulary of *Activities* may correspond to the OV-5a Operational Activity Decomposition, providing a taxonomy of terms and definitions for the activities necessary to deliver an intended capability. These will be used as the labels for tasks and sub-processes in the detailed DoDAF V2.0 OV-6c business process model.
- Together with terms and definitions for related *Resources* and *Performers,* BPMN collaboration models identify critical information and resource exchanges between performers that perform tasks within and among business processes, which may be represented in DoDAF V2.0 OV-2, OV-3, DIV-1, and at greater levels of detail support executable systems and services with the DoDAF V2.0 SV-4, SV-6 and SV-10c models.
- Furthermore, analysis of applicable Laws, Regulations, and Policies (LRP) and their specification in relation to the concepts defined in a robust *C.A.R.P vocabulary* provides the basis for a DoDAF V2.0 OV-6a Operational Rules Model that defines business rules in terms of the domain vocabulary and

exhibits key traceability from the originating LRPs to constraints on specific elements of the architectural description.

The C.A.R.P. method suggests a simple progression of model development for vocabulary-driven enterprise architecture focused on a minimal set of DoDAF-described models as illustrated in Figure 2-2. This procedure starts by capturing basic overview and summary information represented in the AV-1 DoDAF-described Model describing the vision, goals and scope of the project, then develops of a baseline controlled vocabulary driven by the C.A.R.P. method. See also Appendix D.

Figure 8.6-2: Model Development Procedure



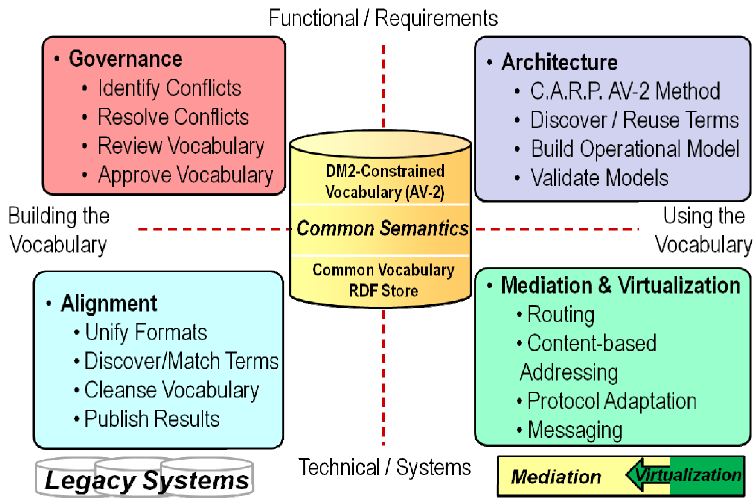
The baseline AV-2 provides an initial vocabulary used to ‘bootstrap’ and drive the development of subsequent DoDAF-described models as required. The AV-2 vocabulary is continually refined throughout the model development process as needed. New terms and definitions required to fully represent a DoDAF-described model are added to the AV-2, while terms and definitions already in the baseline AV-2 may be refined and clarified with a better understanding of the concepts achieved through their use in the more detailed DoDAF-described models.

8.6.6 Federated Vocabulary and AV-2 Development

In DoDAF V2.0, the DM2 “also serves as a roadmap for the reuse of data under the federated approach to architecture development and management.” The DoD is migrating to the concept of a set of federated architectures and vocabularies, where individual vocabularies and architectures based on Communities of Interest (COI) capture, define and maintain terms and architectural descriptions specific to the particular domain. As illustrated in Figure 3-1, development of a common vocabulary within a COI requires:

- focused consistent new vocabulary development for architecture, e.g. C.A.R.P. method
- legacy vocabulary (system) alignment:
- Vocabulary comparison/mapping/mediation
- domain-level governance: and conflict resolution

Figure 8.6-3: Building and Using a Common Vocabulary



Terms and definitions necessary to describe an architecture are identified and recorded in the AV-2. This can often be simplified through reuse of data previously collected by others that is relevant to the current effort. Access to appropriate COI data and other architecture information, discoverable via DARS and the DoD Metadata Registry (DMR) can allow architects to discover and reuse data and other architecture artifacts that prove useful in a current effort and ultimately support the federation of related architectures.

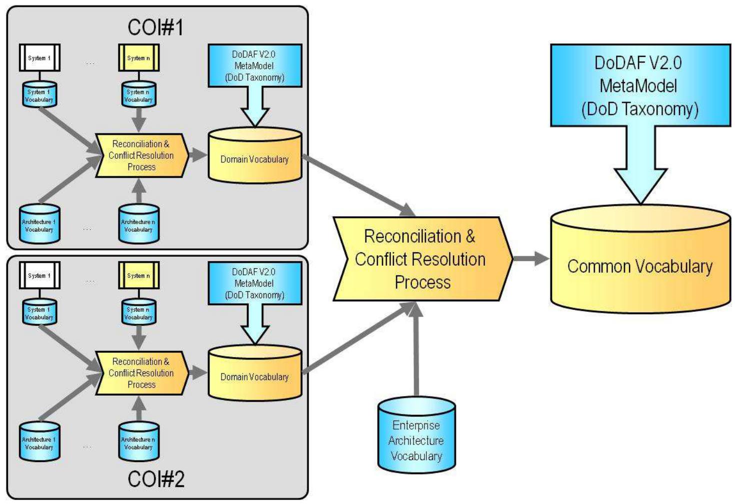
Analysis of common vocabulary across different architectures with similar scope also helps to clarify and determine appropriate architecture scope, and ultimately support the goal of architecture federation. For example, all CV-2 DoDAF-described models should be rooted in or linked to the authoritative source Joint Capability Area (JCA) taxonomy to support federation and interoperability. Likewise, all OV-5 DoDAF-described models should be rooted in or linked to appropriate authoritative sources such as the Universal Joint Task List (UJTL) to support federation, interoperability and reuse.

Across COI boundaries, broader more enterprise-level common vocabularies capture and harmonize terms common across community domains. These vocabularies result from the resolution of a common intersection of concepts, terms, and definitions in the individual COI vocabularies. Development of an enterprise-wide common 'core' vocabulary requires:

- focused consistent new vocabulary development for architecture, e.g. C.A.R.P. method
- cross-COI vocabulary alignment: Vocabulary comparison/mapping/harmonization/mediation
- enterprise-level governance, in coordination with the COI domain-level governance

Figure 3-2 illustrates federated vocabulary development and enterprise-level governance across multiple COIs, each with domain-level governance for both architecture vocabulary development for new systems and existing vocabulary alignment for legacy systems.

Figure 8.6-4: Federated Vocabulary Development



8.6.7 AV-2 Registration and Discovery

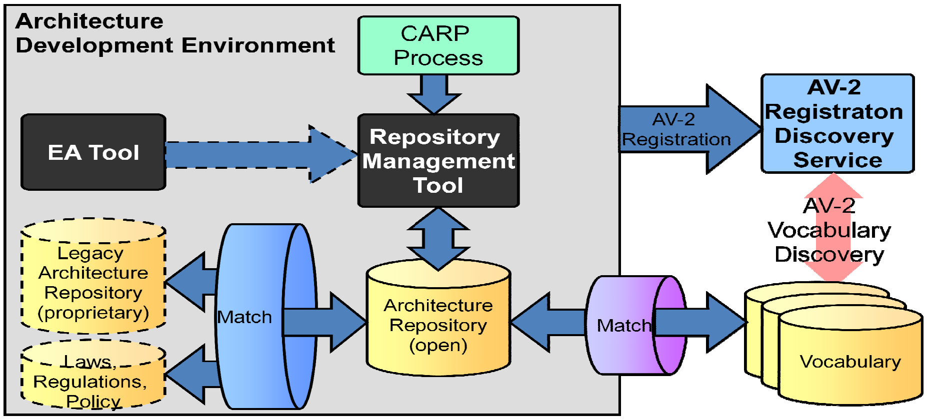
DoDAF architectures with consistent and harmonized AV-2 Integrated Dictionaries are necessary for effective architecture federation, interoperability, and reuse across the enterprise. The development of DoDAF architectures with consistent AV-2s requires that architects have ready access to the approved AV-2 Integrated Dictionaries from other related architectures for comparison and potential reuse, as well as access to any authoritative common 'core' vocabularies which may be mandated.

While the DoD Architecture Registry System (DARS) provides a central point for registration and discovery of architecture AV-1 Overview and Summary Information, there is currently no consistent way to discover and access the AV-2 Integrated Dictionary associated with a registered architecture. Net-centric principles (discovery, accessibility, understandability, and trust) suggest requirements for a DoD AV-2 Registration / Discovery Service that allows users to search and download dictionaries of architecture terms and definitions (AV-2s) and provide:

- Discovery metadata, e.g. creator, publisher, and version
- Level of approval and source of authority, e.g. approval authority, approval status, and date
- Associated Architecture AV-1 Information
- Access to authoritative common 'core' vocabularies, e.g. JCA

Figure 3-3 illustrates an example Architecture development environment (e.g. BEA, JACAE …) interacting with a notional service for registration and discovery of architecture AV-2 Integrated Dictionaries.

Figure 8.6-5: AV-2 Registration and Discovery



8.6.8 General Process for AV-2 Development

The generic process for the development of an AV-2 consists of seven steps and incorporates the C.A.R.P. vocabulary 'bootstrapping' method. The process should be initiated after the initial outline of the architecture has been developed, i.e. AV-1 and OV-1 exist, and is consistent with the one described in the "DoDAF Architecture Development Process for the Models" Microsoft Project Plan.

An AV-2 consists of defined terms and derived terms. Defined terms are those specified at the outset of an architecture project, while derived terms emerge during the development of subsequent architecture views. The purpose of this process is to ensure a sufficient set of defined terms at the beginning of an architecture project, and to allow for subsequent expansion and extension of this initial set of terms. It is an iterative process that accompanies the development of other architecture views.

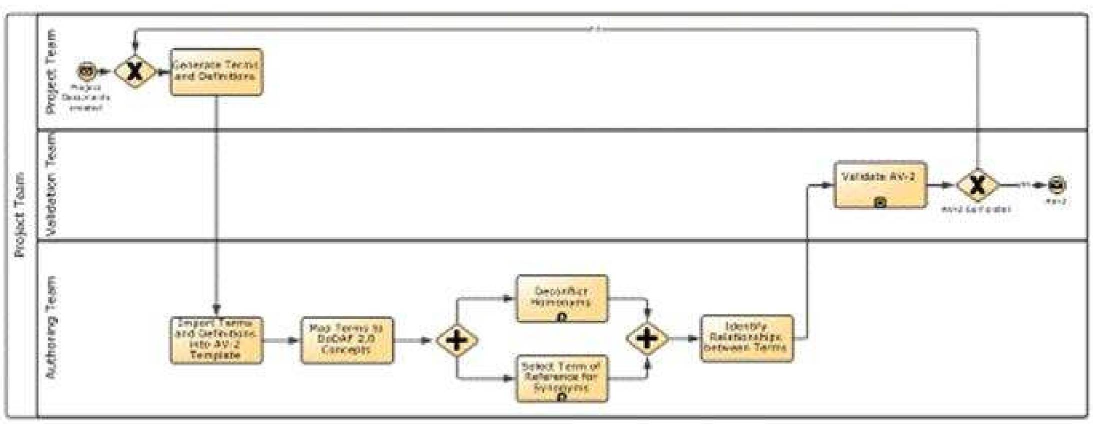


Figure 3: General AV-2 Development Process

8.6.8.1 Generate Terms and Definitions

During this step the key terms are gathered from domain subject matter experts (SMEs) and a set of definitions is created. An initial vocabulary baseline should be established using the C.A.R.P. method to 'bootstrap' an architecture development effort as a first step in defining scope – corresponding to Step 2 of the DoDAF six step development process. At the very start of the architecture development effort, these terms and definitions are typically derived from the AV-1 and related documents, and includes the definition of mandatory architecture components required by project sponsors and architecture users. As development of the architecture progresses, additional terms and definitions are identified and documented during the creation of other architecture models and views and this process repeats until the required completeness, coverage, and level of detail is achieved.

8.6.8.2 Import Terms and Definitions into AV-2 Template

Development of the AV-2 is currently supported by a simple template allowing the architect to relate each term and definition to a DoDAF Meta Model (DM2) concept. The AV-2 should initially be focused on what the target architecture should be capable of achieving, not how this functionality should be rendered.

8.6.8.3 Map Terms to DoDAF 2.0 Concepts

During this step the existing terms are mapped against the DM2 concepts. The starting point should be the key elements of the DM2: Capabilities, Resources, Activities, and Performers. Note the DM2 contains many additional elements which will be defined and refined in later development cycles.

8.6.8.4 Deconflict Homonyms

In order to disambiguate term homonyms the architect should either change one of the homonym terms, or add a suffix that specifies the context of the related definition (e.g. tank[army] vs. tank[air force])

8.6.8.5 Set Term of Reference for Synonyms

In case of multiple terms that relate to the same definition the architect should determine one term of reference. Additional terms can be explicitly listed as synonyms, but should not be listed as terms in their own right.

8.6.8.6 Define Relationships between Terms

Dependencies between terms (such as generalization/specialization and whole/part relationships) should be documented in this step.

8.6.8.7 Evaluate AV-2 Completeness and Coverage

The final step of the development process tests the AV-2 for coverage of the DoDAF Meta Model and completeness against project requirements. If the AV-2 is found to be incomplete a new round of revisions is initiated, otherwise the result of the process is the finished AV-2.

In many cases the content of the AV-2 will emerge throughout an architecture design project. The first occurrence of an AV-2 term will thus be in a particular model that represents a view of the underlying architecture. Similar to the bottom-up validation approach it is possible to trace the model construct containing the term to the underlying DoDAF Meta Model, and determine from there which other model types should be populated with this term. The figure below shows this validation process formalized in BPMN.

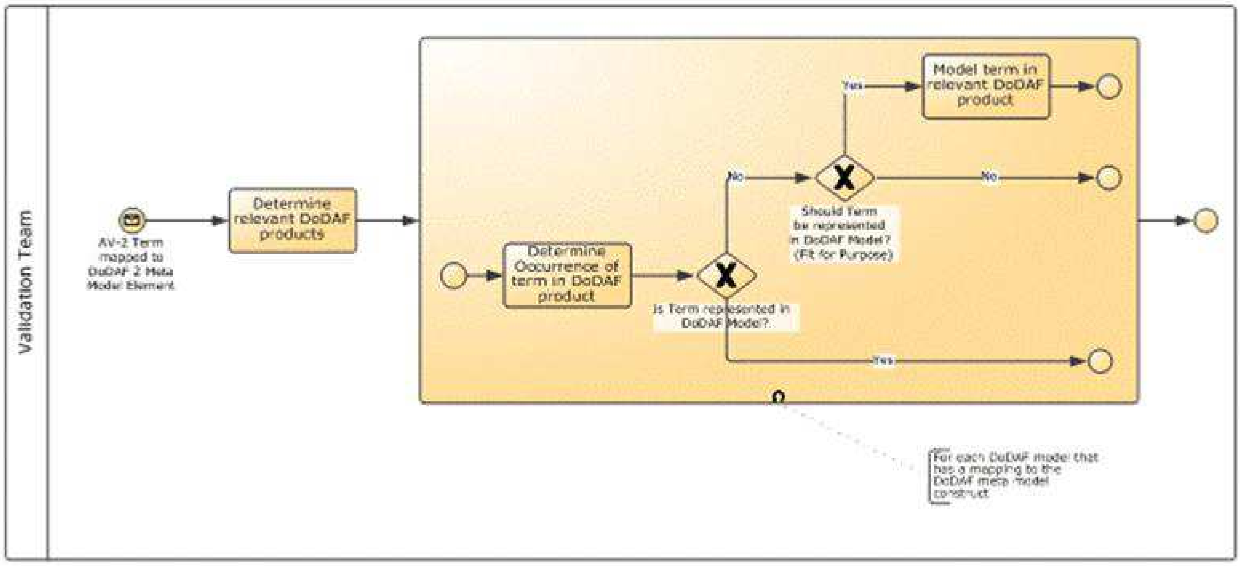


Figure 4: General AV-2 Validation Process

8.6.9 Template for AV-2 Development

In order to support the development of AV-2 views an Excel template is provided, as illustrated in the figure below. This template can be used for data capture. Given the DoDAF mapping of Meta Model concepts to architecture models that contain them, the template can help identify the set of architecture models within which the defined term is relevant. By mapping the terms in an AV-2 to the concepts of the underlying DoDAF Meta Model it is possible to trace the relationship between a term and the different architecture models in which this term occurs. In the future this template could be replaced by a web-based form that is linked to a database for easier storage, manipulation and rendering of AV-2 content or integrated as part of an architecture tool.

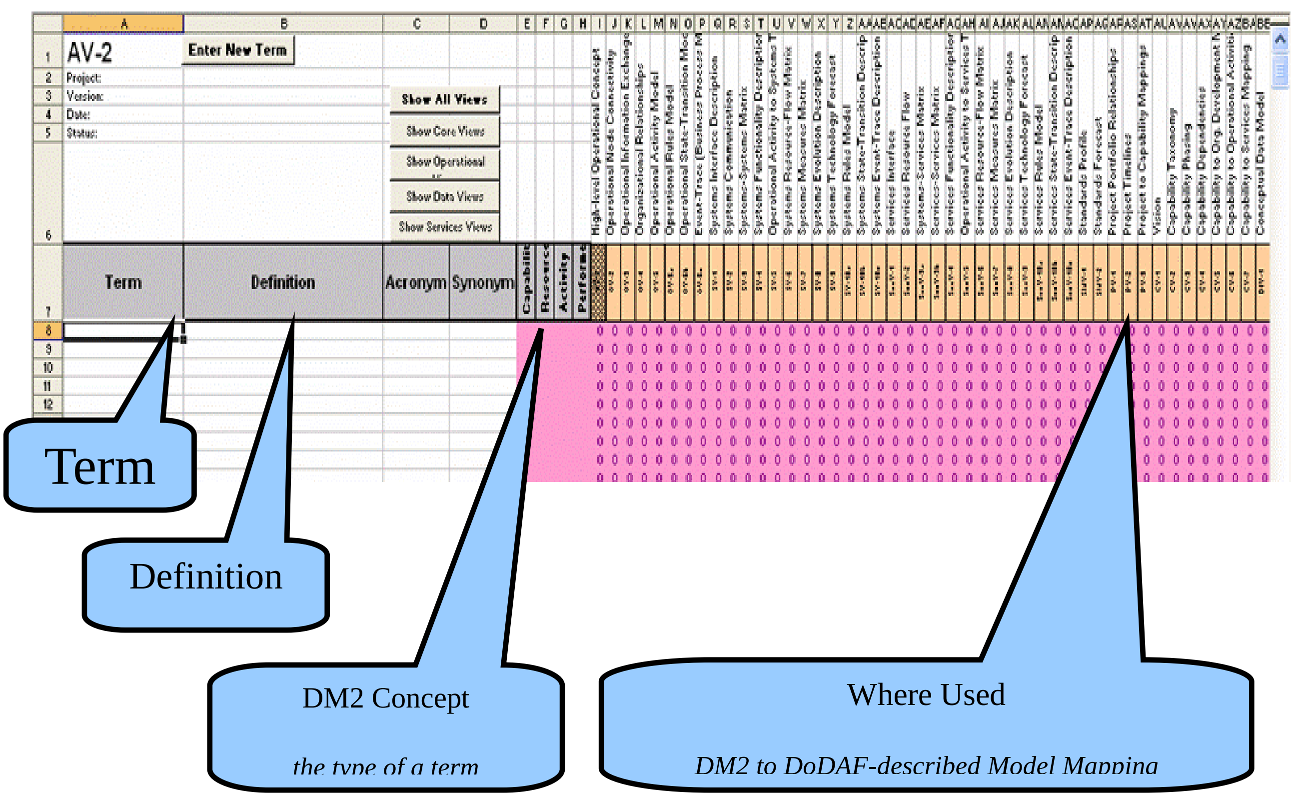


Figure 5: Example AV-2 Development Template

8.7 Guidelines for the Description of Business Rules in DoD Enterprise Architecture

8.7.1 Introduction

Enterprise Architecture in the DoD Business Mission Area (BMA) is designed to support informed decision-making and analysis based on cross-domain End-to-End (E2E) business processes, as opposed to focusing on individual business system investments. The Architecture helps to ensure IT investment decisions are made with clear, contextual understandings of the positive and/or negative impacts to the Department of Defense. It also intends to help realize the benefits of business process re-engineering, portfolio management, and interoperability.

Enterprise Architecture in the BMA is therefore focused on accurate, consistent and comparable descriptions of ***business process models***, which represent desired ***capabilities*** of the BMA, ***activities*** required to achieve those capabilities, ***resources*** (data and information assets) that are produced and consumed by those activities, and the ***performers*** (organizations, person roles, systems, and services) responsible to carry out the activities. The BMA has adopted established DoD, national and international industry consensus standards for these aspects of the architectural description data. [7]

Associated with these EA descriptions are ***rules*** that exist to enforce laws, regulations, and policy intended to guide and constrain the ***behavior*** of the process or performers as well as the ***structure*** of the organization and its data and information assets. This document discusses the role and standard representations of these rules in DoD BMA architecture

[7] See 'Guidelines for the Design and Development of Operational Activity Sequences (DoDAF OV-6c) using BPMN' and 'Guidelines for Development of an Integrated Dictionary for Enterprise Architecture (DoDAF AV-2)'

8.7.2 Enterprise Architecture: Model Structure and Domain Content

Enterprise Architecture in DoD is described by various views on the enterprise as defined in the DoD Architecture Framework (DoDAF). A DoDAF view is expressed through the data population of one or more DoDAF-described models. These views capture both *model structure* and *domain content*. This basic distinction between model structure and domain content is illustrated in Figure 6.

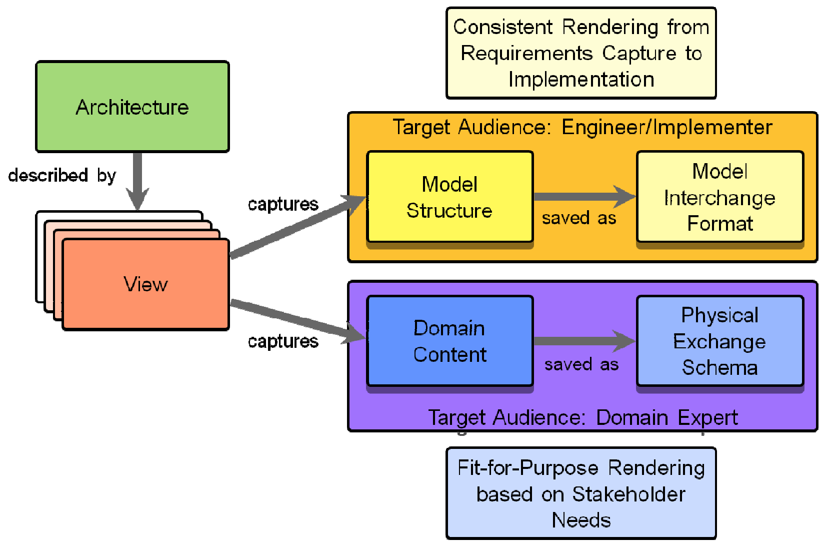


Figure 6 Architecture Structure and Content

Model structure refers to the arrangement and layout of the formal model elements, such as boxes and arrows in a process model. This includes both *logical structure* reflecting the semantics of the model topology required for implementation (i.e. what connects to what) and *physical structure* required for model interchange (i.e. the physical coordinates of elements in a diagram). Examples of model structure include network topology and process flow. Model structure is of primary interest to engineers and implementers who require consistency from requirements capture to implementation to enable round-trip engineering.

Domain content refers to the subject matter concepts represented in a model, such as the specific name labels associated with activities in a process model. This content contributes to and must be consistent with an established enterprise vocabulary. Domain content is of primary interest to domain and subject matter experts who require a clear and consistent vocabulary to support various stakeholders based on their particular information and decision-making needs.

Domain content captured using standard modeling techniques for one view may be re-purposed and rendered in different standard views –comprising an *integrated architecture*. For example, a specific organization captured in a DoDAF Organizational Relationships (OV-4) model may be the same organization acting as a performer in a DoDAF Operational Activity Sequences (OV-6c) model. Domain content may also be combined, or 'mashed up', with additional content captured in other standard models to provide specialized 'fit-for-purpose' presentations of the domain to support particular stakeholder needs.

8.7.3 Domain Content: Vocabulary Concepts and Rule Constraints

Domain content in architecture is saved as a vocabulary. In the BMA, architecture vocabulary is represented as an *ontology,* which defines terms in the domain (i.e. *concepts*) and relationships among them (i.e. *facts*), with related representation of constraints – the associated rules. This distinction between related concepts and associated constraints (i.e. *rules*) is illustrated in Figure 7.

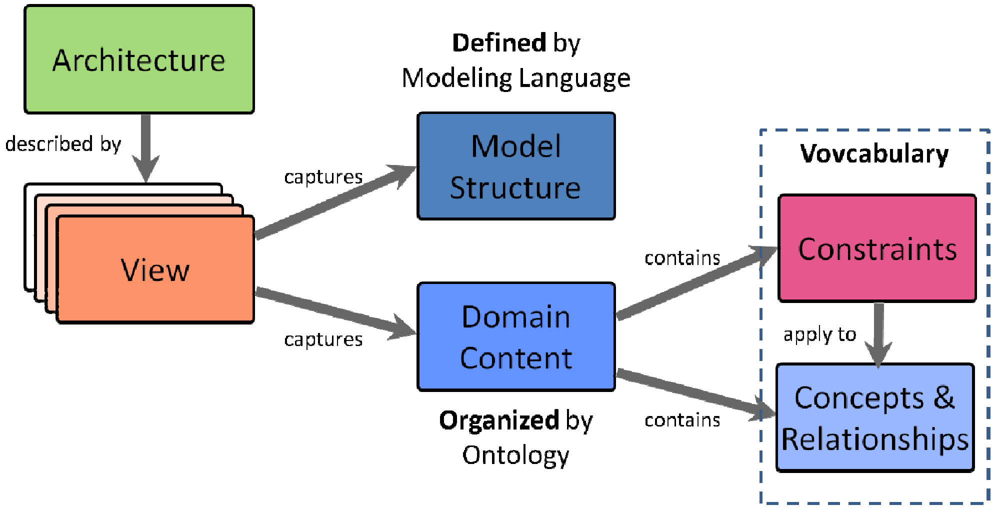


Figure 7 Domain Content as Concepts, Relationships and Constraints

Rules are in general constraints that govern. A rule is more formally defined as "a proposition that is a claim of obligation or of necessity"; a *Business Rule* is further defined as "a rule that is under business jurisdiction." A rule's being "under business jurisdiction" means that it is under the jurisdiction of the semantic community that it governs or guides - that the community can opt to change or discard the rule. Laws of physics may be relevant to a company (or other community); legislation and regulations may be imposed on it; external standards and best practices may be adopted. These things are not business rules from the company's perspective, since it does not have the authority to change them. The company will decide how to react to laws and regulations, and will create business rules to ensure compliance with them. Similarly, it will create business rules to ensure that standards or best practices are implemented as intended.[8]

8.7.4 Business Rules and Rule Types

Business rules govern aspects of an enterprise related to 'what' the enterprise intends to accomplish, and typically reflect statements of law, regulation, policy, standards, or best practices in terms of the Business Analyst and Subject Matter Expert. The intent of business rules is to capture the essence of the business in business terms, and to describe and automate aspects of the business function in a declarative instead of a procedural way (What not How).[9]

---

[8] OMG Available Specification, Semantics of Business Vocabulary and Business Rules (SBVR), v1.0
[9] Jan Vanthienen, Ruling the Business: About Business Rules and Decision Tables

The Object Management Group (OMG) specifies a basic procedure model to define business vocabulary and rules in their industry standard Semantics of Business Vocabulary and Business Rules (SBVR), as illustrated in Figure 8.

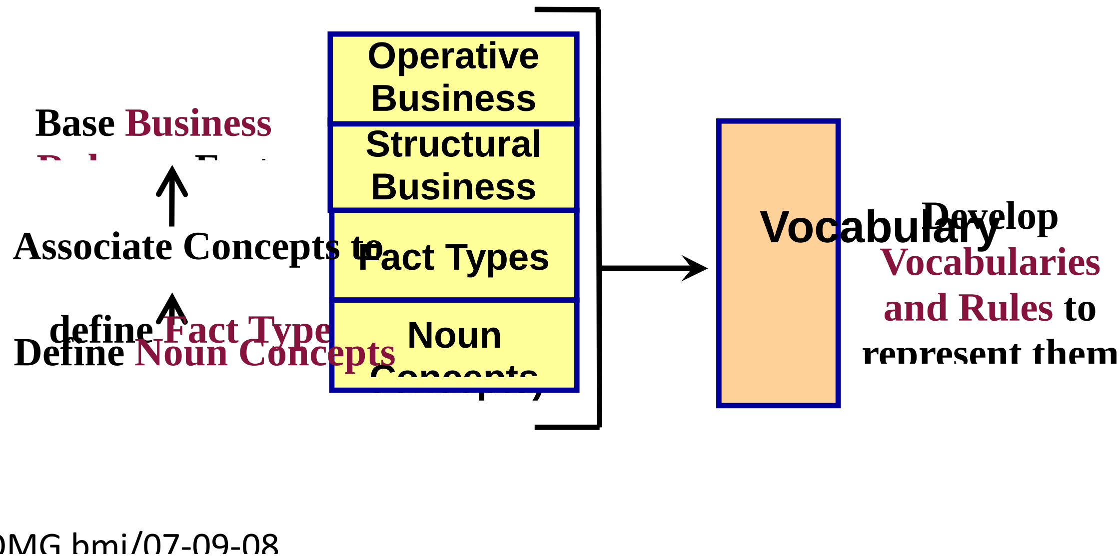


Figure 8 OMG model for Business Vocabulary and Rules

The main concepts are initially defined as nouns (i.e. *terms*), then relationships among these concepts are added as verbs (i.e. *facts*). Next, *structural rules* are defined to represent the constraints on concept properties and facts (cardinalities, property value constraints etc.). Finally *operative (*or *behavioral) rules* are defined to represent who may do what based on the facts. All these combined (nouns, verbs, structural and behavioral rules) together define the architecture vocabulary.

8.7.5 Structural and Behavioral Rules

A business rule has been generally defined as "… a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behavior of the business."[10] Inherent in this definition are two types of business rules: "… to assert business ***structure*** or to control or influence the ***behavior*** of the business."

Structural rules describe constraints that govern structure and inter-relationship of data. Such rules express criteria for correct decisions, derivations, or business computations. Structural rules, also called 'definitional' rules, supplement the definitions of concepts and govern either property (attribute) value constraints or the relationship between concepts as defined in Architecture models, e.g. classification or concept properties such as cardinality.

- Structural rules may be described in terms of the facts (relationships) that relate concepts (terms) together. To say that a customer can place an order is a structural business rule. Structural rules can be documented as natural language sentences or as relationships, attributes, and generalization structures in a graphical model. For example, "TRANSCOM is one of the Unified Combatant Commands" is a structural rule related to organizational structure.

[10] Business Rules Group, Defining Business Rules ~ What Are They Really?

- Structural rules may also operate on the properties (attributes) of concepts. For example, "Field 25 (corresponding to some concept property) must contain a number between 1 and 10 if field 24 (corresponding to some other property) has the value 'Y'.
- Structural rules may also describe *derivations* that indicate how the population of a fact may be derived from the populations of one or more facts or how a type of individual may be defined in terms of other types of individuals and facts.

Behavioral (or operative) rules are statements of a constraint or condition that limits or controls the actions of the enterprise. Behavioral rules, also called 'action assertions' or simply 'constraints', address issues related to quality of service or decision logic associated with a particular concept, e.g. sequencing or branching among activities in process models. Behavioral rules are especially important for business process models and realizing model-driven execution. Various classes of behavioral rules are identified:

- *Conditions:* a condition is essentially a "test" that helps application determine whether to perform certain actions or test other action assertions. An example of a condition can be, "has a customer been late or delinquent for any installment payment?"
- *Integrity constraints:* an Integrity constraint is an assertion that must always be true. An integrity constraint specifies conditions for valid state. For example, an integrity constraint would prohibit creating a sales order without a valid customer id or approving a loan without an acceptable credit score. Integrity constraints are also referred as validation rules.
- *Authorization:* an authorization typically specifies permissions to perform certain actions. Typically it is used to define security and access control rules. For example, "only a branch manager can approve an order of more than $100,000".

8.7.6 Business Rules and Technical Rules

Within DoD BMA Architecture, we use the term '*Business Rule*' to refer to a design-time operational rule in terms of the business analyst that corresponds to a high-level law, regulation or policy. Business rules constrain the 'what' in enterprise architecture and work on operational concepts described in logical models of the Operational Viewpoint (OV). Business rules may be either structural or behavioral. DoD BMA architecture promotes the use of the OMG standard SBVR 1.0 for formal structured English representation of the 'what' business rules.

We use the term '*Technical Rule*' (or 'Production Rule') to refer to a representation of those same constraints on run time system/service models of execution (and executable) processes. Technical rules constrain the 'how' aspects of the enterprise. These are sometimes called 'system/service' rules, and generally express how business rules are realized within the information technology.

Typically the 'how' is the instantiation of the 'what; just as 'what' business rules may be either structural or behavioral, their expression as 'how' technical rules correspondingly may also constrain either structure or behavior. In both cases, a technical rule may be implemented either explicitly in the appropriate model (behavioral or structural) or external to that model in a format fit for execution in a suitable rules engine as illustrated in Figure 9.

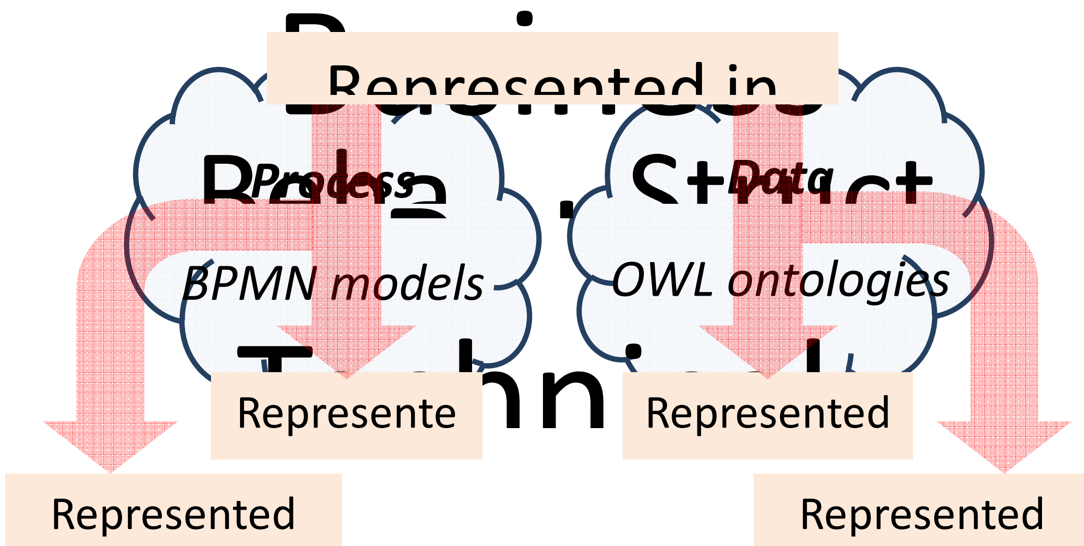


Figure 9 Business and Technical Rules

For example, DoD BMA Architecture uses the OMG standard BPMN 2.0 at both operational and executable levels for representing formal process models of behavior upon which behavioral rules operate. While business rules are consistently represented in SBVR, the corresponding technical rules may either be represented explicitly in the BPMN model, such as the decision logic of a gateway dictating a ‘split’ of control flow in the process, or they may be represented external to the model in some formal language (e.g. decision tables) for execution in a BPMS or Rules Engine.

Likewise, consider structural rules defined to represent the constraints on concepts, properties and facts. In DoD BMA Architecture, terms and facts are formally captured in vocabularies as ontologies defined by Communities of Interest (COIs) within domains using the W3C standards OWL 2.0 and RDF. So for example, the structural business rule “Each Contract must have at least one Contract Line item” can be implemented as a technical rule in different ways. One way is to constrain a relationship (fact) between two concepts (terms) with the appropriate cardinality and optionality representing the referential integrity constraints. Alternatively, one may define an explicit rule external to the structural model (e.g. in a dialect of the W3C standard RIF) to be executed on the model by an appropriate rules engine.

8.7.7 Rule Traceability

In DoD, business rules are driven by statements of law, regulation, and policy expressed as guidance by various Directives, Instructions, Manuals, and Standards. *Rule traceability* is the important ability in Enterprise Architecture to formally track a governance constraint from its original law, regulation, or policy through its various representations in the different perspectives of Architectures and levels of Implementation (see Figure 10).

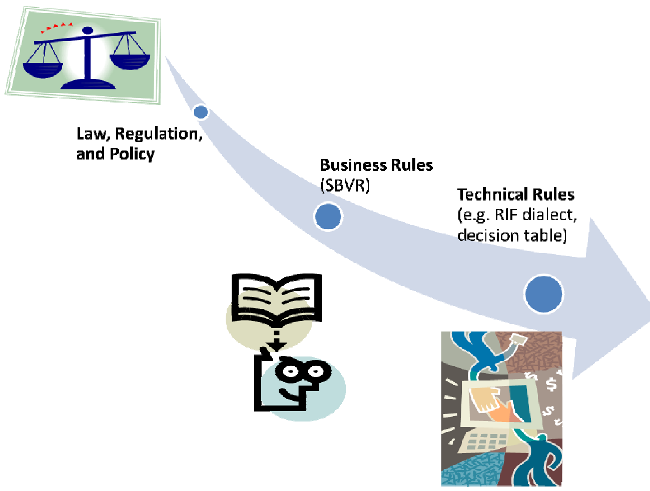


Figure 10 Rule Traceability

8.7.8 Rules, Architecture and Standards

Formal representations of business rules typically exist within a context of other concepts such as vision, goal, objective, strategy, and tactic as related to operational business processes and their associated activities, resources, and performers. Example representations of the context and role of operational business rules are the OMG Business Motivation Model (BMM) 1.1 and the DoD Architecture Framework (DoDAF) V2.0 Meta Model (DM2). The DM2 defines concepts corresponding to the types of information content required for DoD architectures and models. This formal representation defines a vocabulary of projects, desired effects, and capabilities as they relate to operational 'business' processes and their activities, resources, and performers. Rules in the DM2 V2.03 Conceptual Data Model are called out in Figure 11 below.

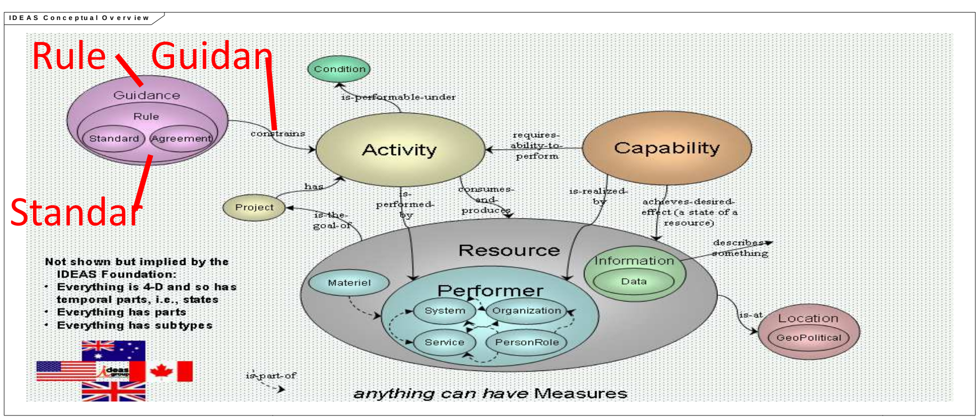


Figure 11 DoDAF MetaModel and Rules

Rules exist at different levels and apply to different domains across various perspectives in an enterprise. They are represented in different Viewpoints of the Architecture description, specifically the Operational Viewpoint (OV) and System/Service Viewpoints (SV/SvcV), as illustrated in Figure 12.

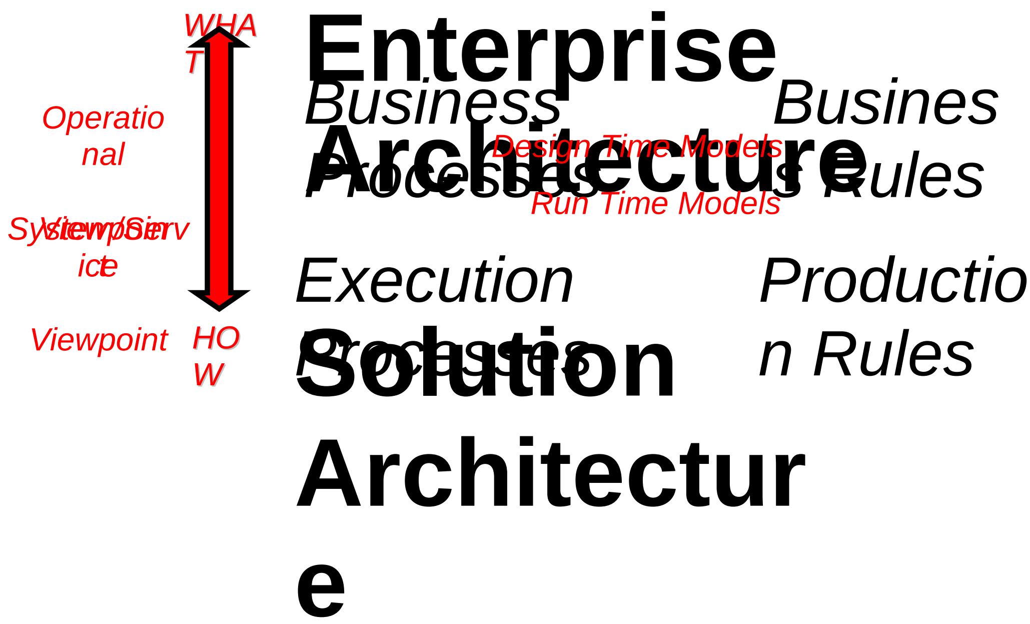


Figure 12 Perspectives on Processes, Models, and Rules

Business rules act on the activities and vocabulary of business processes and data models describing the design-time perspective of the Operational Viewpoint of Enterprise Architecture. In contrast, technical (production) rules act on system/service level activities and vocabulary described in run-time models of the System/Service Viewpoint of federated Solution Architectures.

8.7.9 Standards for DoDAF Models

DoDAF-described operational viewpoint OV-6a models represent applicable laws, regulations, and policy in business terms using the vocabulary of business requirements owners. OMG standard SBVR 1.0 is appropriate for formal structured English representation of business rules at the operational level.

DoDAF-described system/service viewpoint SV/SvcV-10a models the implementation of that policy in terms of the system/service vocabulary of run-time execution domain. W3C standard RIF 1.0 may be an appropriate choice for representing the implementation of business rules in solution architectures.

The DoDAF OV-6c and SV/SvcV-10c both represent process models, the former from the operational business 'what' perspective and the latter from the run-time execution 'how' perspective. OMG standard BPMN 2.0 (specifically the analytic conformance sub-class) is appropriate for representing process models at both operational and executable levels in DoD architecture. These various DoDAF models and Industry Consensus Standards are illustrated in Figure 13.

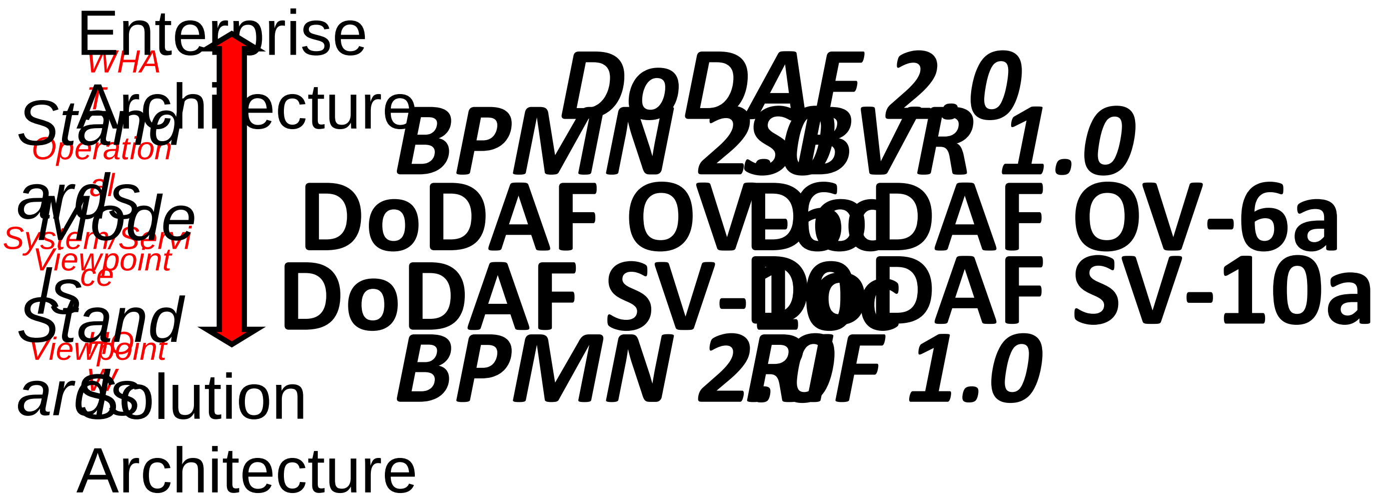


Figure 13 Standards and DoDAF Models

SBVR is not machine-executable; to get to that level we need a more formal representation. The translation between SBVR and executable rule representations such as RIF or BPMS decision tables can be done through various tools. SBVR does not specify any executable rules language; a specific language (e.g. RIF dialect) could be selected to help ensure consistency across DoD. Also an automated business rules writing toolset could facilitate the capture of structured English (SBVR) to represent applicable laws, regulations, and policy and also help in the automatic transformation from SBVR to an appropriate executable rules language.

8.7.10 Rules, Vocabularies and Domain Federation

Rules must be expressed using well-defined terms from the vocabulary or their application domain. Domains (e.g. HR, FM) must specify standard vocabularies in order to consistently describe architecture content within their scope. Communities of Interest (COIs) within domains should formally define vocabularies (ontologies) using the W3C standard OWL 2.0.

Different domain architectures are federated by appropriate mapping of domain concepts, either/both directly or through use of a common shared domain (ontology) that represents high-level concepts shared by the different domains as illustrated in Figure 14.

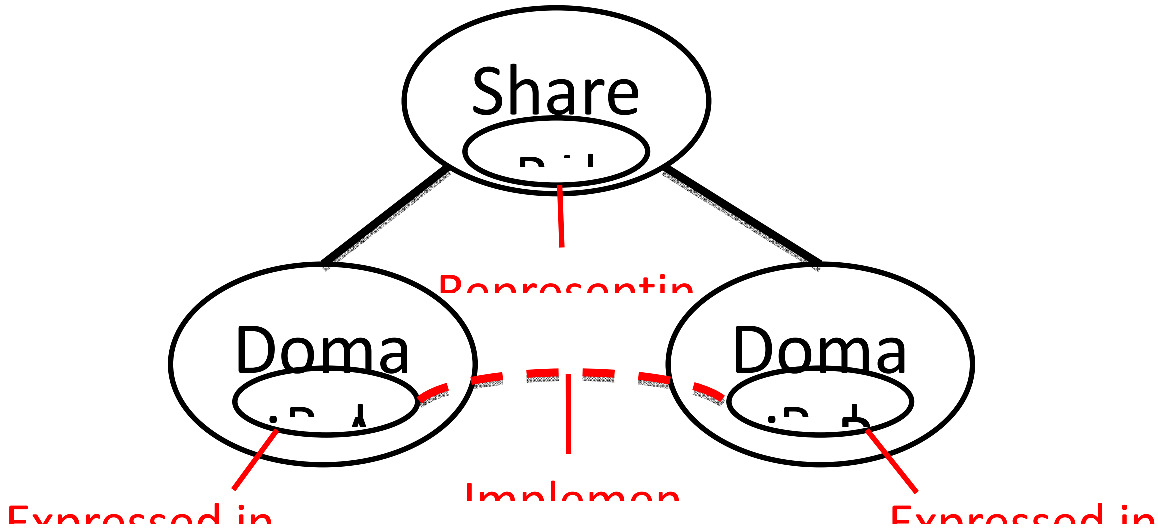


Figure 14 Rules across Domains

8.7.11 Rules, Process, and Model Driven Implementation

Model Driven Implementation (MDI) is a methodological approach designed to achieve architectural round-tripping through the use of Semantic Technology as part of its modeling, solution generation and runtime phases. Architectural round-tripping is translation of information from models to executable code, and the propagation of changes at the executable level back to the conceptual models.

As shown in Figure 15, MDI is a three step approach with the following structure:

1. Model the business capability to be developed in terms of baseline vocabulary, rules, and end-to-end processes,
2. Data: Extend the common vocabulary with semantic content, additional terms and rules, and
3. Implement the desired capability and deploy as a business/data service.

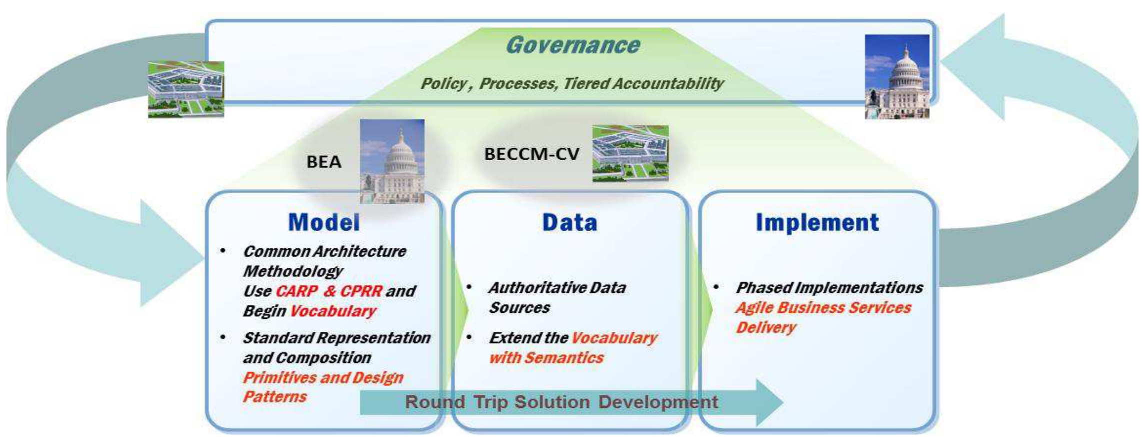


Figure 15 Model Data Implement

The linkage between process and rules is a critical component of the MDI method. Process and Rules are linked via use of common or linked vocabulary concepts defined as semantic architecture content as illustrated in Figure 16.

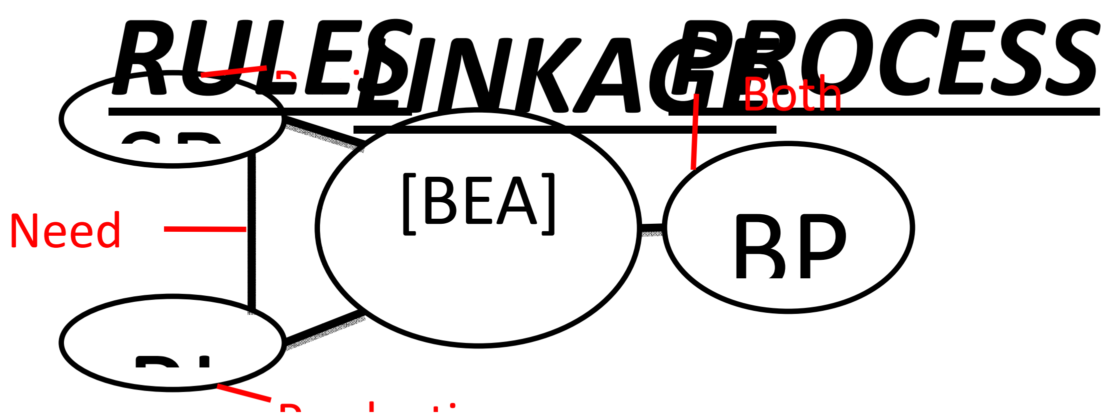


Figure 16 Linking Rules with Process

There are two options for invoking run-time executable 'production' rules from a run-time executable rendition of a business process: through Java software calls via the Java run-time API for Rule engines (JSR 94) and/or through web-service invocation managed by the BPMS, as illustrated in Figure 17.

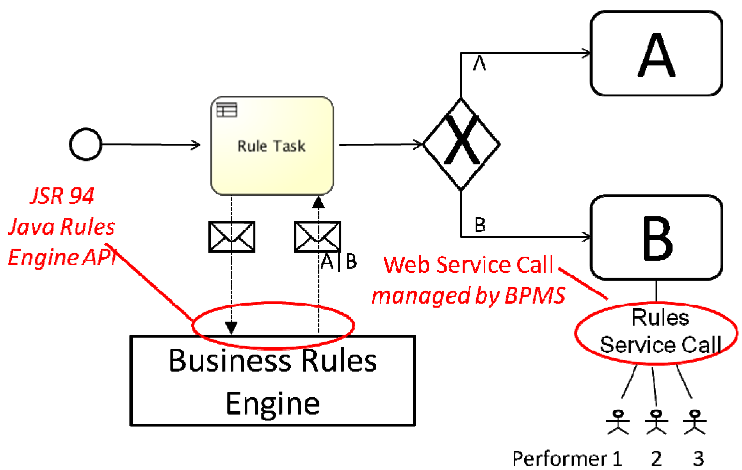


Figure 17 Rule Invocation from executable BPMN

8.8 Creating Solution Architectures as SV-10c Models in DoDAF Using BPMN 2.0

In the DoDAF, different views are identified within the architectural framework for containing and relating various architecture objects. The operational view of event traces in a business process is represented as an OV-6c construct, which per Department guidance is to be done as a Business Process Model and Notation (BPMN) model. The Deputy Chief Management Office (DCMO) of the DoD has promulgated modeling guidance and training on how to create OV-6c models using the notational subset of the BPMN v2.0 specification known as the Analytic Conformance Class. This guidance consists of normative modeling approaches based on BPMN 2.0 primitives and patterns defined by the DCMO. Application of these approaches is required of artifacts created for the Business Enterprise Architecture (BEA), which has led to numerous key Business Process Areas (BPAs) being modeled as OV-6c views across the various end-to-end processes of the DoD's Business Mission Area (BMA).

There are many counterpart system views to operational views in the DoDAF, including a system view of event traces in a business system that is supporting one or more business processes. This counterpart is known as the SV-10c view, and it can be used to define the solution architecture of a business system that exists or is proposed that purports to provide such automation support. While an SV-10c is not technically one of the DoDAF views required for defining the BEA, it nonetheless should be done for legacy systems or proposed replacement systems – whether they are custom-built, commercial-off-the-shelf (COTS), government-of-the-shelf (GOTS), or some combination thereof – that claim or assert automation support for one or more business processes <u>and</u> conformance with the governing architecture for the one or more business processes. Alternatively expressed, an SV-10c should be used to show conformance of a business system with a corresponding OV-6c's requirements and architecture.

This DoDAF Journal article has been written to explain how this can be done using current DCMO BPMN 2.0 modeling guidance with some additional guidance related to modeling system behaviors. A principal import of this proposed approach is that BPMN 2.0 should be used to describe the behavior of systems rather than using other modeling languages such as Sequence Diagrams (SDs) from the Unified Modeling Language (UML). With its additional notational elements and behavioral semantics, BPMN 2.0 is far more capable of describing system behaviors than UML SDs. Furthermore, with BPMN 2.0's XML serialization, it is now possible to attribute the modeled elements in a way that richly fills out the XML export of a BPMN 2.0 model. This export can be translated into Resource Description Framework (RDF) format using the BPMN 2.0 ontology created by the DCMO and expressed in Web Ontology Language (OWL) format. Such a semantic expression provides the basis for directly measuring the conformance of a business system with governing architectures similarly expressed in OWL (by way of targeted queries of the architecture data using the query language for semantic data, known as SPARQL).

8.8.1 Modeling Challenges for Solution Architectures

Historically, there have been numerous problems associated with the definition of solution architectures as a means of determining conformance with governing architectures.

- Differences in spans of control – the controlling execution context in which operational or system behaviors occur – between the system's view of the functionality it is supposed to deliver versus the organization's view of the same functionality. For example, an organization's view is process-centric and may recognize more than one span of control, whereas an Enterprise Resource Planning (ERP) system assumes it is the only span of control.
- Models created to describe business systems and to define business system functionality have typically proved lossy with respect to corresponding predecessor models created to describe business processes and to define business process requirements. True model round-tripping – being able to seamlessly trace back and forth from the two different types of models – has

proved elusive and technically challenging. For example, shifting from using BPMN 2.0 (or comparable language) to describe a process to using UML SDs (or comparable language) to describe a system has made the transition from analysis-to-design-to-development problematic and reverse engineering impractical.

- Even using BPMN 2.0 can prove tricky since the modeler is able to represent the system as either a performer corresponding to a swim lane (wherein modeled elements are to be executed by the system), a completely abstracted performer based on certain task types (wherein the system has only indirect representation in the model), or a distinct participant separate from the process pool that is represented as a system pool.

There are also the explicit differences between operational and system views within the DoDAF. These differences are highlighted in Table 1 below.

• **OV-6c** ***(Operational Event Trace)***	• **SV-10c** ***(System Event Trace)***
– Focus: • Describes operational activity • Provides time-ordered examination of resource flows as a result of a particular scenario • Operational thread defined as a set of operational activities, with sequence and timing attributes, and associated resources needed	– Focus: • Describes system activity • Provides time-ordered examination of interactions between functional resources • System thread is a sequence of functions and data interfaces that honor information needs of participating resources
– Intended Usage: • Analysis of operational events • Behavioral analysis • Identification of non-functional user requirements • Operational test scenarios	– Intended Usage: • Analysis of resource events impacting operation • Behavioral analysis • Identification of non-functional system requirements
http://dodcio.defense.gov/dodaf20/dodaf20_ov6c.aspx	http://dodcio.defense.gov/dodaf20/dodaf20_sv10c.aspx

Table 1 – OV-6c vs. SV-10c in DoDAF

The manifestation of these view-driven differences has generally been revealed in confusion and variation in detail for models that purport to describe processes vs. models that purport to describe systems that support those processes. If the latter is simply a more detailed representation of the former, then traceability between the two is difficult to achieve and to preserve. On the other hand, if the latter is a different a model altogether, then comparability is effectively compromised along with the capability to ascertain architectural conformance. Fortunately, there is a way out of this conundrum.

In the corollary OV-6c guidance provided by the DCMO, various modeling views of the same process space are called out:

- Milestone View – describes major phases of a business process as a simple and “happy path” sequence from start to finish, using only a Start Event, an End Event, and collapsed Subprocesses, with Sequence Flows connecting them all together

- Handoffs View – describes the handoffs from one performer to the next in the process sequence, which is either across Lanes in the same Pool with Sequence Flows or between Pools (and spans of control) via Message Flows
- Decisions View – describes the business logic behind control flow and assignment flow as expressed with diverging/converging data-based or event-based Gateways and Sequence Flows
- Procedures View – describes in rich detail the procedure-level behavior of a process, which is achieved through the use of more advanced Flow Nodes and behavioral semantics of BPMN 2.0.

These different views are intended to provide the modeler with guidance on how to iterate a model from initial cuts to a final version, and to support different "fit-for-purpose" needs of the model to speak to different constituencies and stakeholders.

It is possible to call out an additional view, a System View, that allows an organization's process-centric view to be preserved as context for defining the System View. This System View will likely require more detail than the Procedures View, but it can still be done using the Analytic Conformance Class elements. However, some additional model element attributes – taken from the Common Execution Conformance Class – are required to fully outfit the BPMN model for a System View, along with some corresponding modeling guidance.

8.8.2 Introducing the System Pool

In a certain sense, a business system mirrors the one or more business processes that it is supporting through automation. It does this by implementing a function that is mediated by a human performer, is reacted to by a human performer, or is the result of something acted upon by a human performer. In software engineering, this is known as an inversion of control or responsibility, where the business system implements at run-time the functionality defined by the system's interactions with abstracted elements outside the system context. Viewed in this way, a business system represented apart from but in relation to the supported business processes implements a type of delegation pattern that lets the modeled process elements "delegate" the implementation to one or more modeled system elements.

The concept of the System Pool was introduced contemporaneously with the BPMN 2.0 spec via the BPMN 2.0 By Example document (see http://www.omg.org/spec/BPMN/2.0/). Using an Incident Management example, the authors described how the concept of the System Pool could be introduced into a BPMN 2.0 collaboration model as the automated support for the business processes in the collaboration, and that this usage did not replace the collaboration but extended it. The principal import of this approach is that a model preservation strategy – as opposed to a model round-tripping one – is now possible with BPMN. The Incident Management example is adapted and explained below, which is presented as a BPMN collaboration between various participants.

In the Incident Management process collaboration, a problem with a separate (and not shown) business system has generated the "incident" being reported, which is presented to a set of participants charged with fixing the problem. There is a customer facing participant that fields the request, which then hands it off as an opened "trouble ticket" for the Tier 1 Team to attempt to troubleshoot the problem. If the Tier 1 Team cannot fix it, then it is handed off to the Tier 2 Team to fix it. If it cannot be fixed contemporaneously, then the fix is added to the Backlog System for incorporation into the next release of the system that originally suffered from the problem. Into this mix is introduced the Trouble Ticket System as an additional participant in the collaboration that mediates the problem resolution activities.

The first step is to begin with an OV-6c view of the business processes. This is achieved via a BPMN collaboration diagram that calls out the major steps and performers with at least a Handoffs View or Decisions View, or perhaps even a Procedures View, level of detail. This is shown in Figure 1 below.

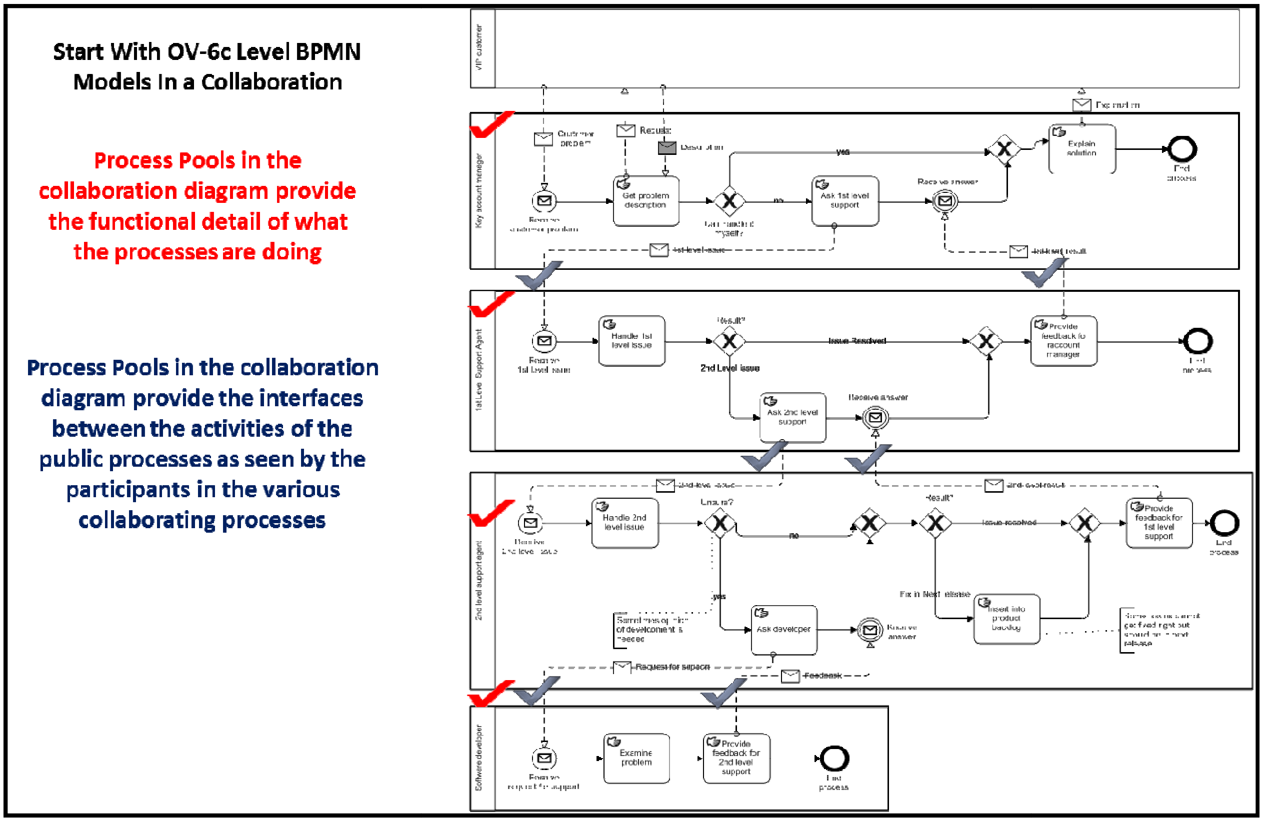


Figure 1 – Process Collaboration as the OV-6c View

(Note that the tasks initially shown in the model are all of the Manual Task type, which is what the original treatment of this approach used. The reason for that usage was that the model was ultimately intended to be executed by an enactment engine, which according to the BPMN 2.0 spec meant that any non-automated task had to be shown as a Manual Task type. However, an OV-6c model is not meant to be executed in this sense, so in practice it is appropriate instead to use an Abstract Task type (no marker in the box) for most activities or Service Task and Business Rule Task types (as needed) to indicate the intended abstractions. Consequently, the reader should equate the use of the Manual Task type in these models to the use of the Abstract Task type that would typically be the case for OV-6c models.)

The second step is to introduce the System Pool, which is initially represented as a collapsed pool (wherein the inner details of the system's design are not visually represented). As with any external span of control, the other participants in the collaboration interact with the System Pool via Message

Flows. These Message Flows into and out of the System Pool represent communicated information proceeding along the interfaces that exist between the other Participants and the System Pool. This configuration is presented in Figure 2 below.

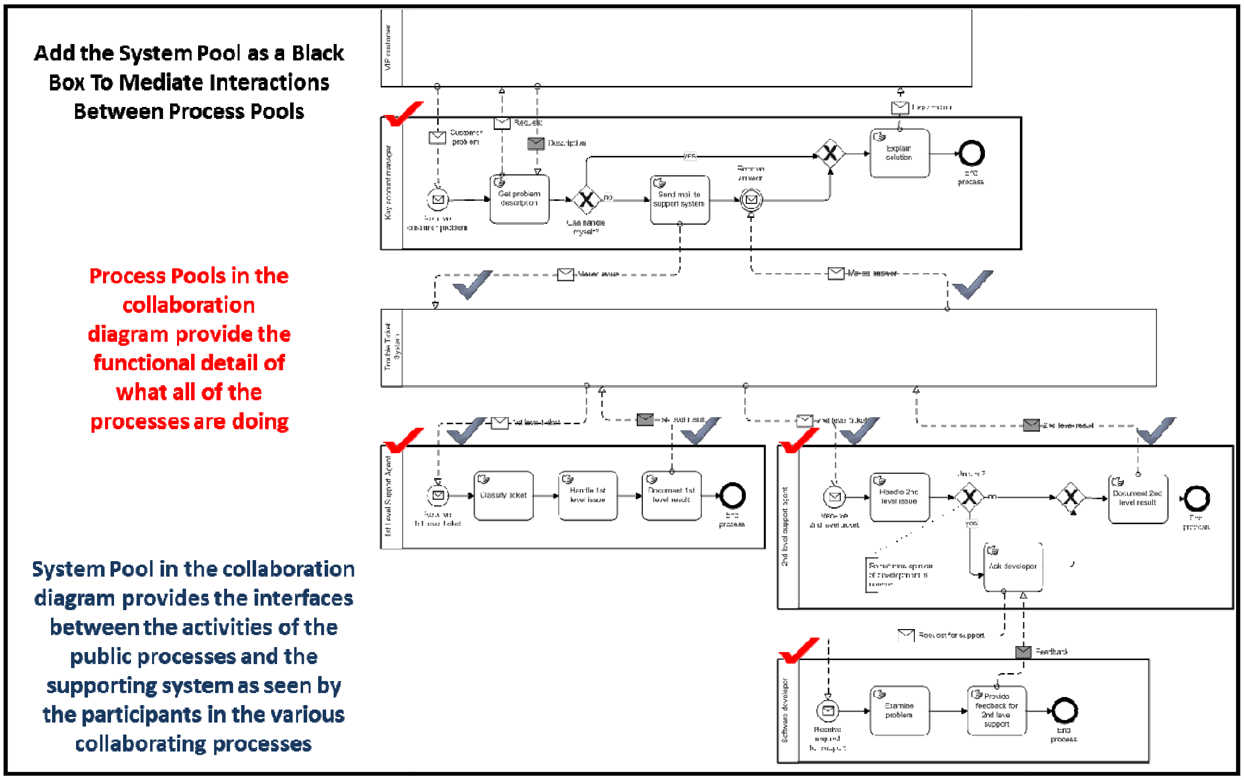


Figure 2 – Introduction of the System Pool

The third step is to then fill out the System Pool with the behavioral detail that describes the functionality of the system. Where in this functionality there is data communicated to or by the system from or to another participant in the collaboration, then Message Flows are connected at the element level on both sides of the communication. The purpose of the Message Flows is thus now clearly revealed as representing the human-to-system (user screens) or system-to-system interfaces (headless interactions via services) that exist within the Trouble Ticket system. This is illustrated in Figure 3 below.

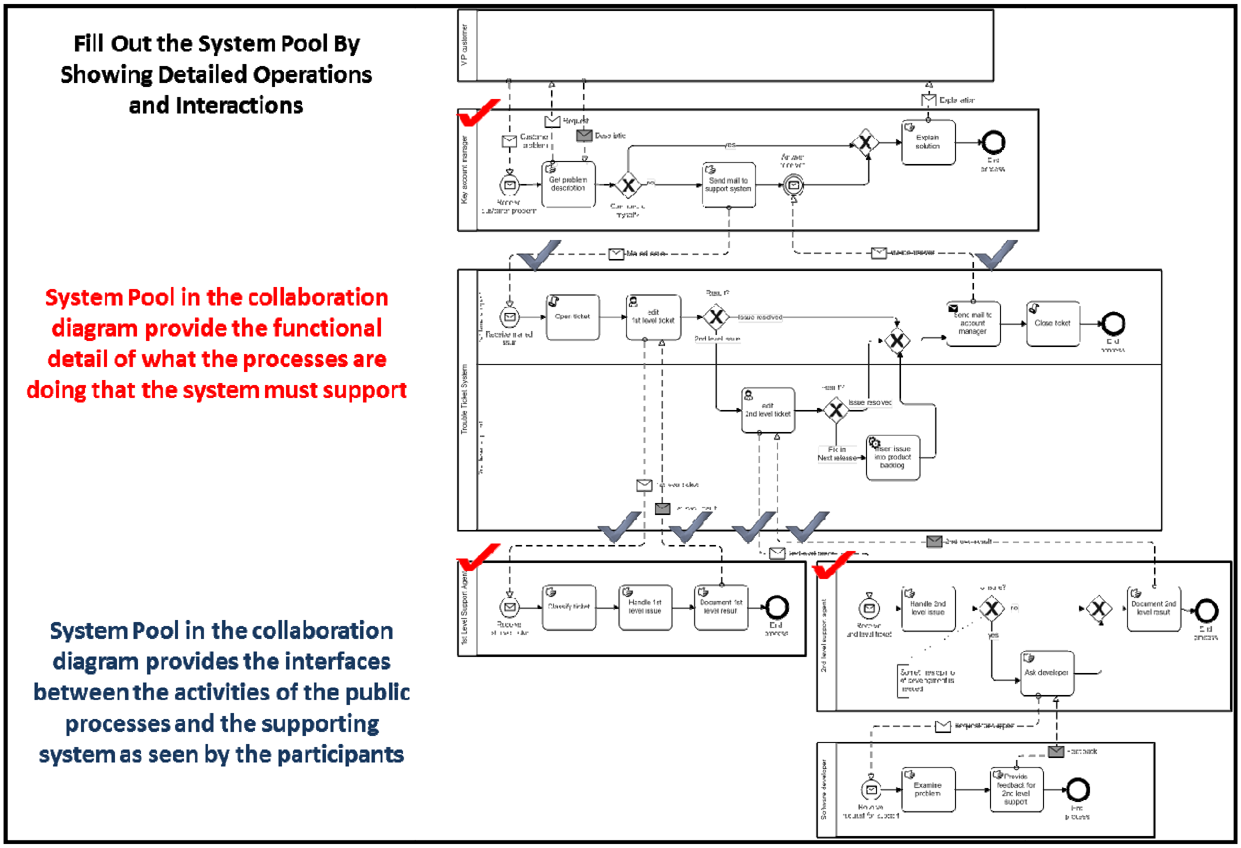


Figure 3 – Description of the System Pool

The final step is to then collapse the other Pools, leaving only the System Pool and the various Message Flows in the process collaboration. At this point, the Message Flows represent the input/output (I/O) requirements of the Trouble Ticket System that must be satisfied. As long as these are honored by the construct in the System Pool, the internal structure of the System Pool can be whatever the modeler uses for the design. Once that internal structure is finalized, the other Pools can be dropped off altogether and the System Pool itself can be handed off to a development team to build out based on the information contained in this system design statement. This is illustrated in Figure 4 below.

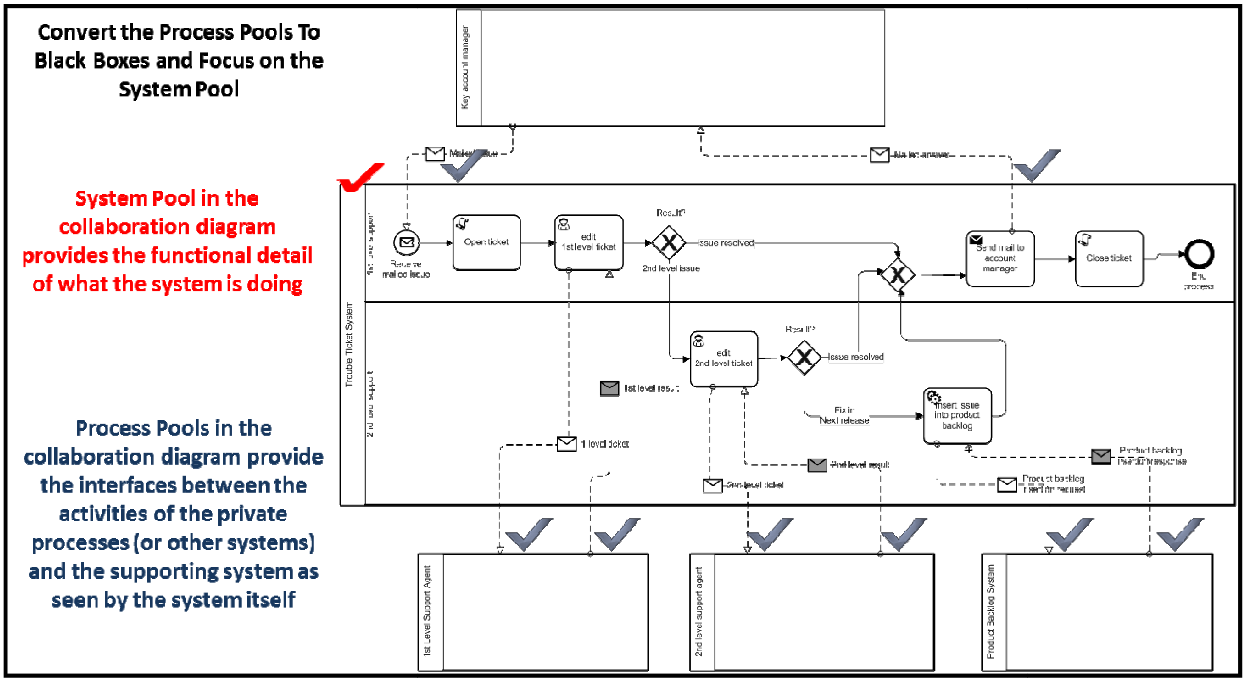


Figure 4 – Focusing on the System Pool

The System Pool can represent any type of automated solution, including an ERP system. Multiple System Pools can be similarly introduced to show how multiple business systems support one or more business processes (as opposed to a single system like an ERP supporting multiple processes).

Thus represented, the process model now has the information about the supporting system that is needed to enable measurement of its satisfaction of requirements, expressed as the informational requirements and output of the other participants. In addition, to the extent that these Message Flows map to Information Exchange Elements as architecture objects defined within an architecture (e.g., the BEA) created per an architectural framework (e.g., the DoDAF), then data as defined schemas or classes are thus mapped to the system that consumes or produces them. This mapping is similarly valid with regard to Data Objects represented in the System Pool, meaning they too can be mapped to defined schemas or classes as transient data consumed, created, or transformed by aspects of the system. The benefit to this mapping approach is that that the system being designed automatically inherits the same set of schemas or classes already defined for (and thus applicable to) the modeled business processes.

Designing the System

Additional modeling guidance beyond what is required of the OV-6c modeling is needed to ensure that the system design is sufficiently descriptive and unambiguous in its intended behavior such that a developer could build and/or configure from the design's corresponding executable constructs. As part of that guidance, it is useful to recognize the numerous analogs to system concepts that are present in BPMN 2.0. This mapping is presented in Table 2 below.

System Concept	Equivalent BPMN Representation
Execution context and framework	• Pool defines the controlling framework for the system • Lanes segment system resource (performer) responsibilities • System interfaces are realized as boundary-crossing moments with other participants
Control flow and routing logic	• Sequence flow and conditional expression logic define order of execution • Gateways and branches define inbound and outbound execution paths
Procedural and modular logic	• Service task, send task, receive task, and business rules task define specific and germane operations that are executed via invocation • User task defines an operation that is executed via one or more user interfaces • Script task defines an operation that manipulates data that is executed internally • Subprocess and Call Activity define a set of operations that are executed separately • Start events and end events define the start and end conditions for execution • Intermediate events define operations based on event type that are executed • Boundary events add operations to the attaching activities that are executed
Data I/O declaration	• Data objects as input or output with corresponding data sets as input or output, respectively, by way of corresponding data associations • Messages and signals define data input or output as data payloads
Integration interfaces (APIs)	• Service task, send task, receive task, business rules task, and user task define service interfaces to abstracted and/or external components • Message event and signal event define service interfaces with other systems

Table 2 – Mapping BPMN 2.0 Concepts To System Concepts

Given this level of equivalency, modeling guidance is critically needed about how to use BPMN 2.0 concepts to represent system-level behaviors. Part of this guidance requires deep understanding and application of first principles of modeling, which can be broken down into two core concepts:

- Decomposition
  - By performer, which ensures separation by those entities responsible for performing atomic tasks (i.e., tasks responsible for a single outcome based on initial conditions)
  - By functional purity, which ensures that atomic tasks are homogenous with respect to executed operations and/or to processed data
  - By state transitions, which ensures that all meaningful changes in the central process object moving through the process sequence are the direct result of atomic tasks
- Abstraction
  - By using a subprocess (a Collapsed Subprocess or Embedded Subprocess), which encapsulates more detailed behavior in a self-contained child process that is performed within the context of the parent process
  - By using abstraction (principally the Service Task or the Business Rule Task), which displaces business logic away from the core process being represented and towards a separate participant as the provider of requested services (a Service Provider or Business Rules Engine, respectively)

- By using reusable components (a Call Activity for a Global Task or a Global Process), which displaces business logic away from the core process being represented and towards a standalone and separate construct that can be invoked.

Through the appropriate application of these first principles of process modeling, the appropriate granularity of modeled systems can be achieved. This granularity is an emergent property that results from the interaction of the degree of coupling between modeled elements (with dependent operations and data) and the degree of cohesion across modeled elements (that fits them together into a whole). These concepts are explained in Figure 5 below.

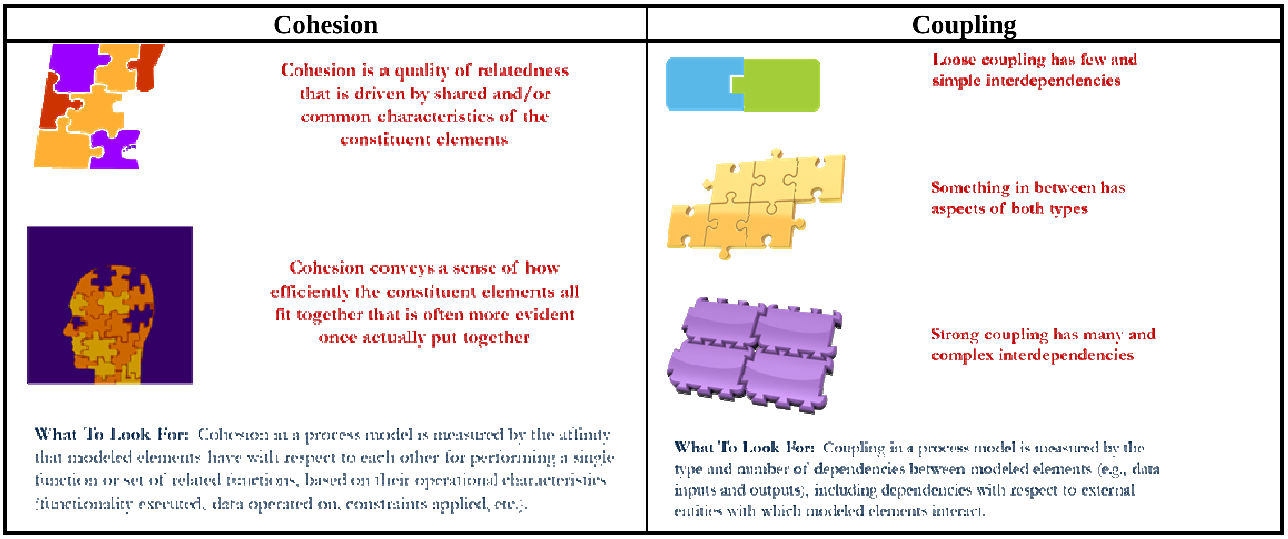


Figure 5 – Cohesion and Coupling as Elements of Granularity

The desired granularity can be achieved through the use of the following guidance with respect to tasks and events, which amounts to making sure that each task or event maps to a single operation. This is summarized in Figure 6 below.

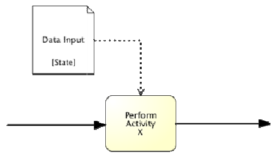
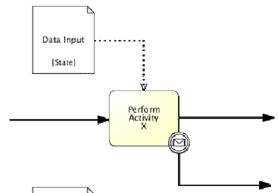
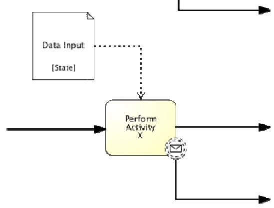
Element	Granularity Guidance
Tasks	 Any activity in an SV10c Level model that: - Is a task: should execute one operation for the activity within the process sequence, commensurate with its task type and any data input and data output associations - Is a subprocess: should effect the passing of control from the parent process to the child (and back) as its one operation to be executed for the activity within the process sequence, commensurate with its subprocess type and any data input and data output associations - Is a call activity: should execute call invocation as its one operation to be executed for the calling activity  An aborting boundary event attached to an activity in an SV10c Level model extends the functionality of that activity but the action of the event constitutes a single operation for execution within the process sequence, commensurate with any data output associations  A non-aborting boundary event attached to an activity in an SV10c Level model extends the functionality of that activity but the action of the event constitutes a single operation for execution within the process sequence, commensurate with any data output associations
Events	 A catching or starting event in an SV10c Level model constitutes a single operation for execution within the process sequence, commensurate with any data output associations  A throwing or ending event in an SV10c Level model constitutes a single operation for execution within the process sequence, commensurate with any data input associations Note: This covers all event types except for the timer event, which is by definition a single operation for execution with the process sequence (i.e., a temporal condition is acknowledged)

Figure 6 – Granularity Guidance By Flow Element Type

The summary effect of the application of the granularity guidance is easily summed up to the following:

- Granularity of the Process, Process Object, and Process Value should all generally match
- Modeled elements at higher levels should exhibit higher cohesion and lower coupling
- Modeled elements at lower levels should exhibit lower cohesion and higher coupling
- Level of granularity should be roughly equivalent across a process level.

The resulting granularity should represent an executable construct or, alternatively expressed, a construct that is sufficiently unambiguous such that a developer can build from it or an enactment engine can be configured to implement it.

8.8.2.1 System Design Patterns

As a system design, there are certain patterns that are optimal in terms of realizing consistent processing of transactions and preserving the integrity of those transactions during processing. One such pattern is ACID (defined below), which is the ideal nature of an individual task at the parent level of system designed in BPMN 2.0. Another pattern is BASE (also defined below), which can be the nature of implementing components that should be encapsulated into lower-order expressions (Subprocesses or Call Activities) or abstracted out (using a Service Task or a Business Rule Task). For example, the implementation of a User Task can consist of several screens (participating in a series of related navigational flows) that exhibit BASE-like properties at the end of each screen experience, but the set of screens are represented as a single User Task because at the point the collection exhibits ACID properties. The nature of these two patterns is described in Table 3 below.

ACID: Tasks should typically be more ACID-like with respect to usage within the BPMN process sequence of the system design	A: Atomic – task is a single operation task C: Consistent – task behavior is consistent I: Isolated – task results are isolated D: Durable – task results are durable
BASE: Tasks that are more BASE-like should typically be collected into abstraction Task types or into Subprocesses or Call Activities	B + A: Basically Available as an operation S: Soft state for the data object E: Eventual consistency of the intended result

Table 3 – ACID vs. BASE Design Characteristics

In general, modeled Tasks should individually exhibit ACID properties, at least at the main or top-level representation, but the system design as a whole does not (and often will not) collectively exhibit ACID properties. The specific characteristics to seek for a given system design depend on the specific needs of the system being designed in BPMN, so these patterns are provided to aid in that determination.

Execution design patterns that can also be followed consist of different ways of processing transactions regardless of whether they are ACID or BASE:

- Idempotent – Process reacts only once for a specific instance of the submitted trigger, no matter how many times the same trigger is sent
- State Machine – Process manages the state of the transaction through a series of stateless moments in the Tasks
- Data Access Management – Cache access service vs. database access service is invoked via a Service Task.

Implementing these patterns amounts to selecting specific implementing component types, which should be a consciously-directed choice of the Solution Architect as opposed to an unintended result.

8.8.2.2 Usage and Attribution of Task Types

In using BPMN to design a system, the concept of the Task needs to be more fully understood. In BPMN 2.0, the Task type has specific behavioral meaning and, in some cases, defines the type of abstraction involved and the nature of the component that executes it. This is defined in Figure 7 below.

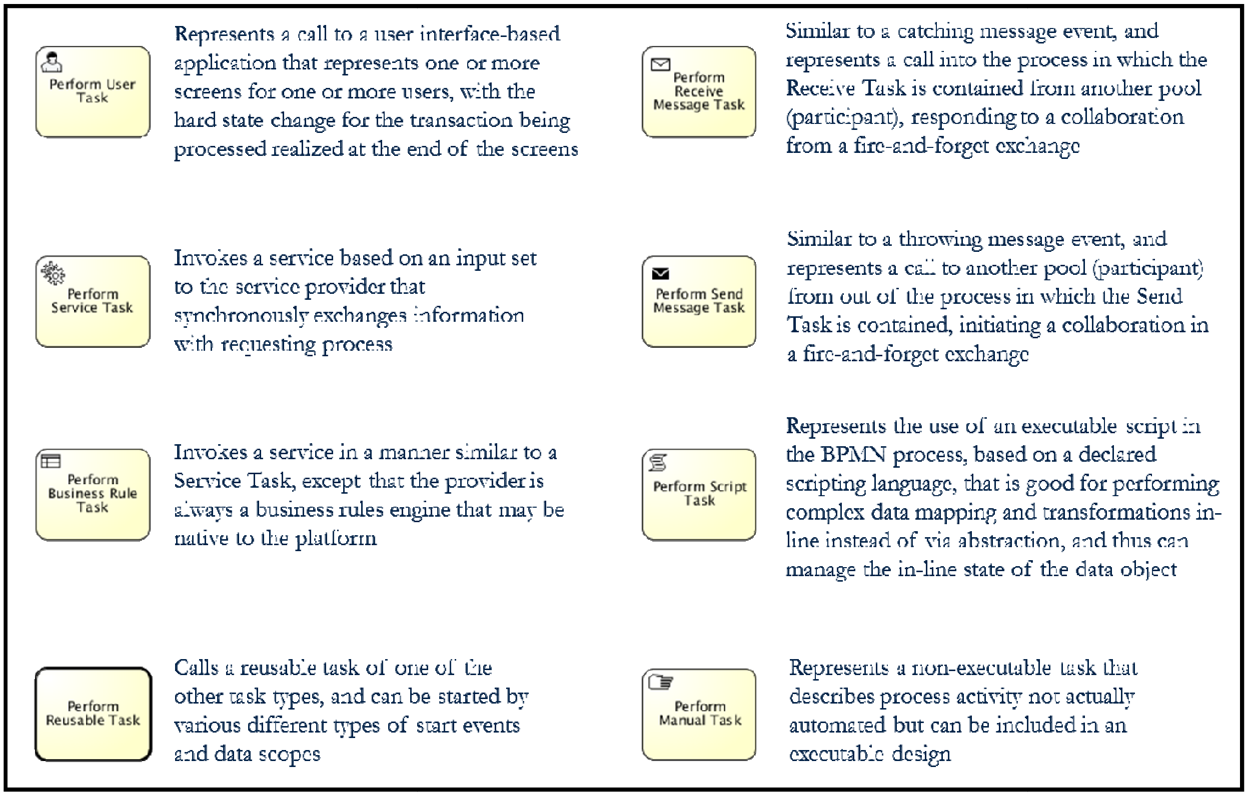


Figure 7 – Specific Nature of BPMN Task Types

To make the most of using specific Task types, it is necessary to capture or define the Operation Name for a Task that is an abstraction and is referencing an implementing component. It is also necessary to capture or define the Implementation Reference for the implementing component, which can be expressed as one of the following:

- Web Service – typically a Simple Object Access Protocol (SOAP)-style web service that leverages relevant application programming interface (API) defined standards for web services
- Unique Resource Identifier (URI) – typically a Representational State Transfer (REST)-style web service that leverages industry accepted API standards for web services
- Other – typically a custom API that is proprietary for the service provider or is required by the requester.

Note that the capability of the modeling tool may constrain how or if this level of attribution can be done, but these attributes – taken from the Common Execution Conformance Class – are available for use within the BPMN XML.

8.8.2.3 Usage and Attribution of Task Types

The concept of Messaging in BPMN 2.0 similarly needs to be more fully understood. In BPMN 2.0, the use of Messages, Message Flows, Message Tasks, and Message Events have specific behavioral meaning and identify the conversations that occur between participants as part of the process collaboration. In particular, the key message exchange patterns (MEPs) of Synchronous Request/Response and

Asynchronous Request/Response can be represented in BPMN, though it requires a modeling convention and attribution scheme to do so. Request/Response is where a request is made by a participant of another participant that may generate a response from the latter to the former.

If this exchange is Synchronous, then the moments of Request and Response are blocking with respect to further processing, and thus are best suited for transaction processing of very short duration. The Request has an input payload, and the response has a concurrent output payload. This yields two Messages and Message Flows: the Request that is initiating and invoked as a call to the service provider, and the Response that is not initiating and is not a separate call to the service provider. The detailed nature of a Synchronous Request/Response MEP is outlined in Figure 8 and Figure 9 below.

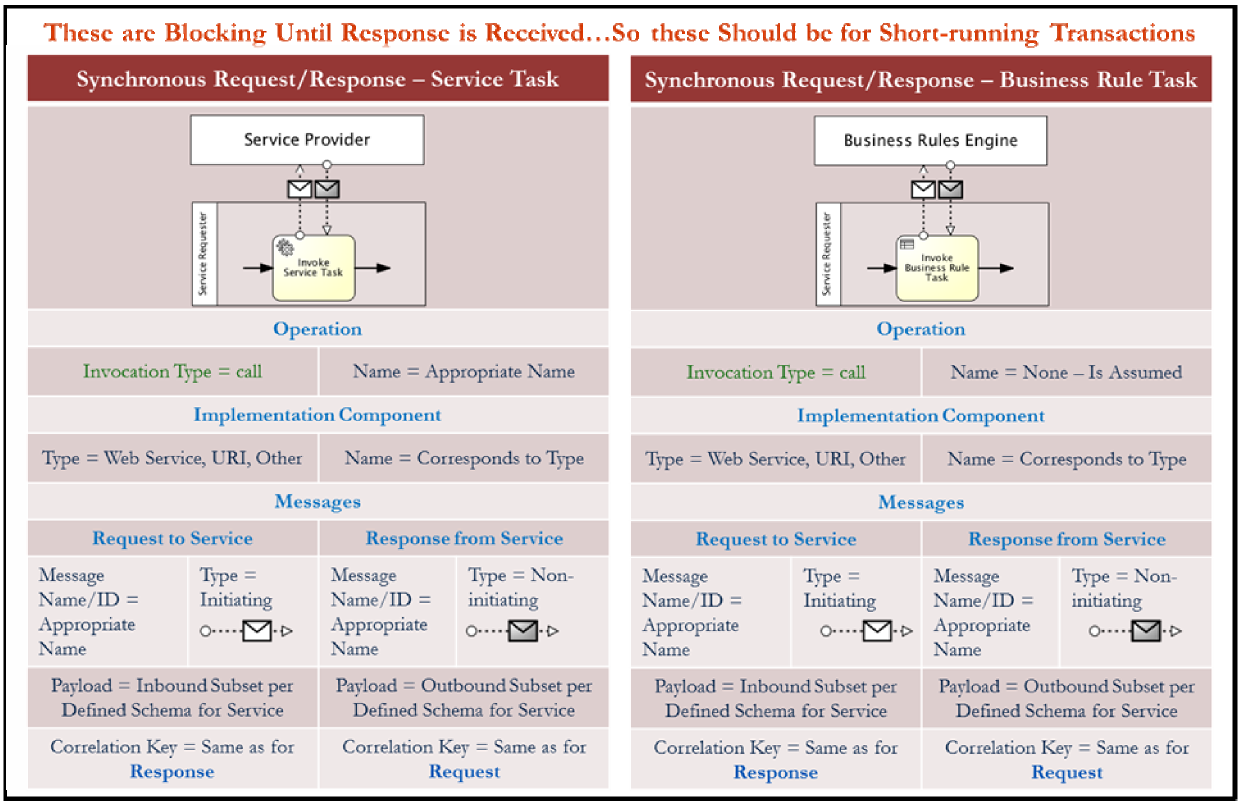


Figure 8 – BPMN Modeling Convention and Attribution Scheme, Synchronous Request/Response

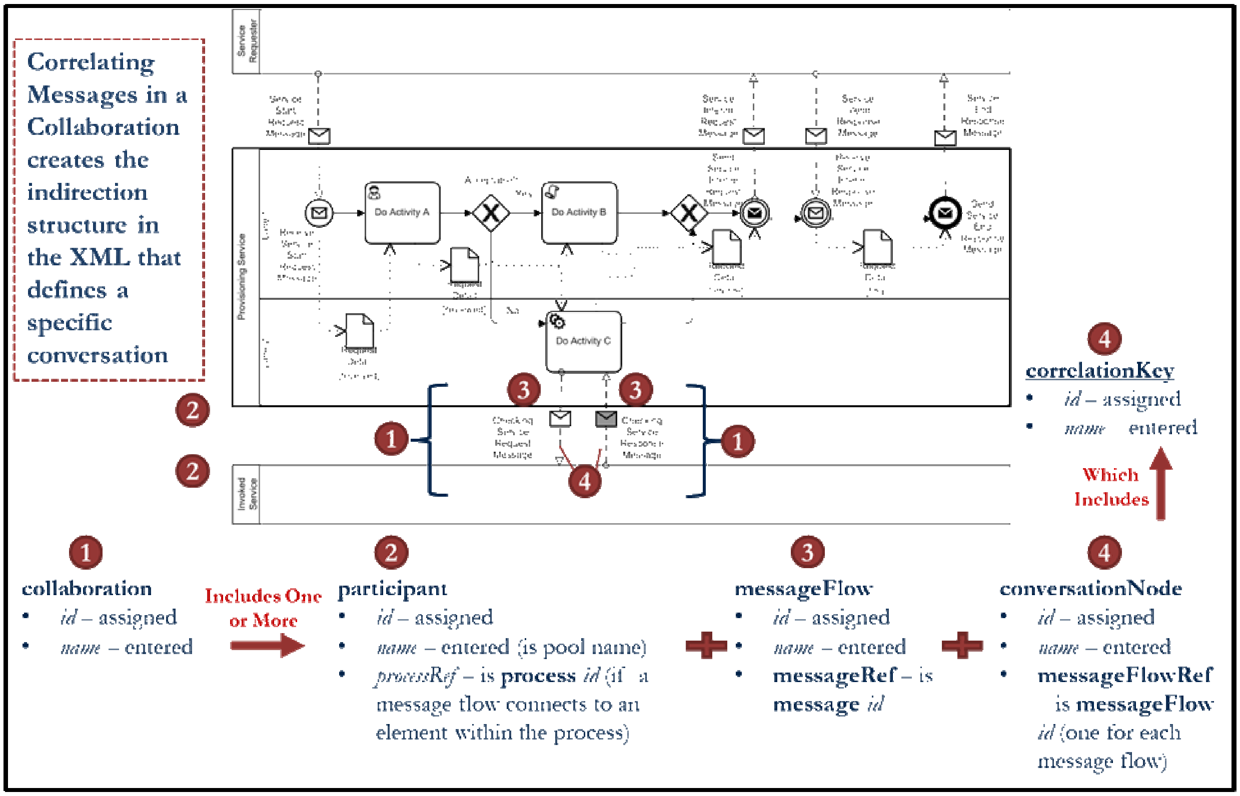


Figure 9 – Synchronous Request/Response Example

If the exchange is Asynchronous, then the moments of Request and Response are not blocking with respect to further processing, and thus are best suited for long-running transaction processing. This also yields multiple Messages and Message Flows: the Request that is initiating and invoked as a call to the service provider, and the Response that is also initiating because it is a call back to the service provider. If there is not Response expected, then the MEP involved here is also known as Fire-and-Forget (or One-Way) since the Request message is sent out and the call made, and that is all that is done. The detailed nature of an Asynchronous Request/Response MEP is outlined in Figure 10 and Figure 11 below.

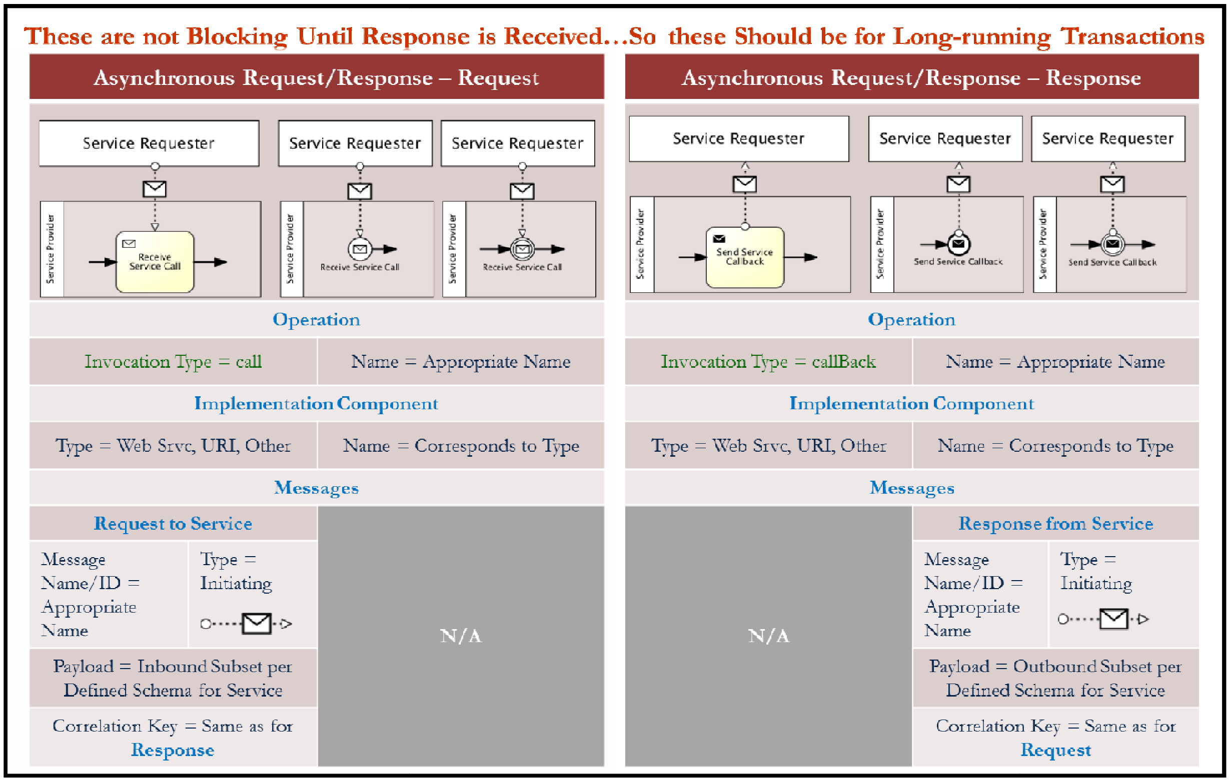


Figure 10 – BPMN Modeling Convention and Attribution Scheme, Asynchronous Request/Response

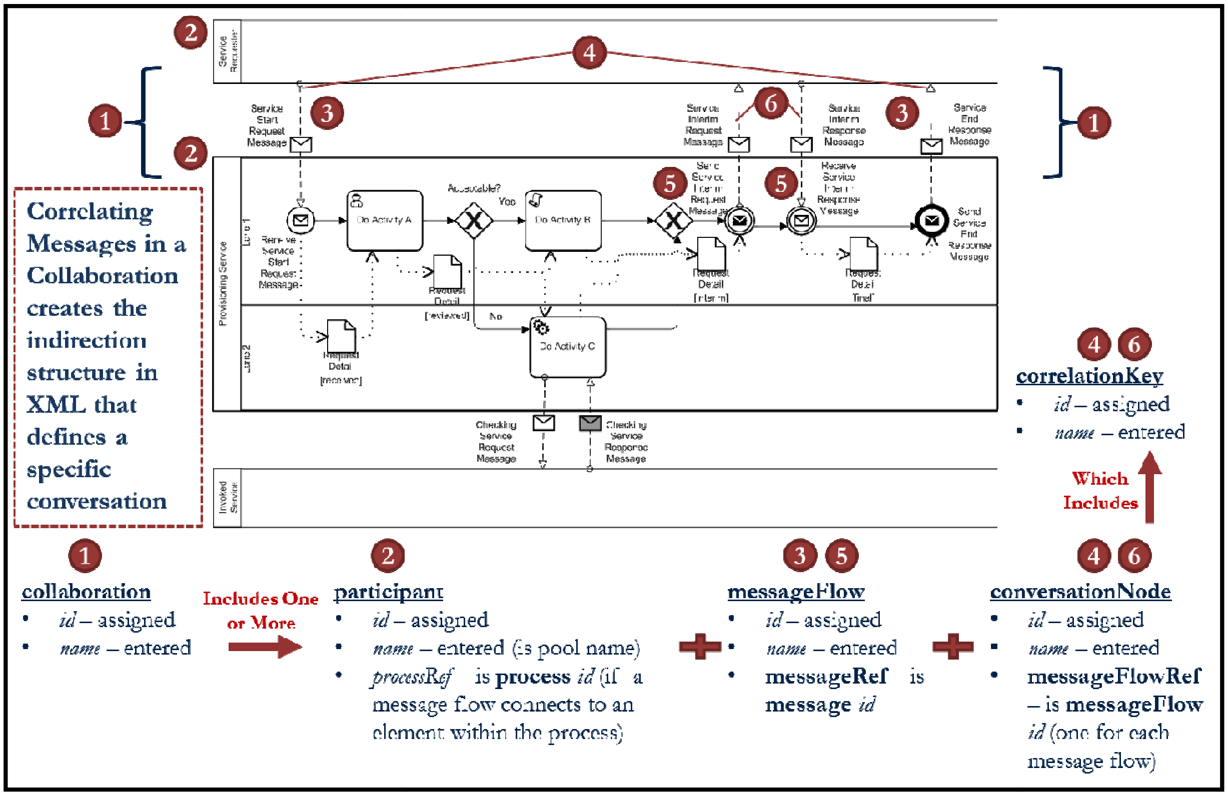


Figure 11 – Asynchronous Request/Response Example

To make the most of representing MEPs, it is necessary to capture or define the Operation Name for the Message Task that is an abstraction or the Message Event that is a service endpoint. It is also necessary to capture or define the Conversation and Correlation Key for the involved Messages. Note that the capability of the modeling tool may constrain how or if this level of attribution can be done, but the these attributes – taken from the Common Execution Conformance Class – are available for use within the BPMN XML. (The other elements represented are already part of the Analytic Conformance Class.)

In addition, the nature of the MEP can also be characterized by the API style that appropriately applies. These API styles match up with the Implementation Reference types mentioned earlier, and are defined more fully at http://www.servicedesignpatterns.com:

- Message-based APIs for SOAP-style web services for appropriately-named operations of some complexity
- Resource-based APIs for RESTful services for simply-named operations of Create, Read, Update, and Delete (CRUD) operations (e.g., see POST, GET, PUT, and DELETE)
- Custom remote procedures call (RPC) APIs for proprietary services for specifically-named operations to effect non-standard interactions.

Implementing these API styles amounts to selecting specific implementing component types, which should be a consciously-directed choice of the Solution Architect as opposed to an unintended result.

8.8.3 Summary and Conclusion

A System Pool that is created using the approach described in this paper can be isolated and exported as corresponding BPMN XML. This BPMN XML can be translated into OWL per the BPMN 2.0 ontology created by the DCMO. Combined and mapped to other ontologies, notably domain ontologies that were created pursuant to the naming of core DoDAF Meta Model (DM2) concepts of Capabilities, Activities, Resources, and Performers (aka CARP), this ontological representation of the system's functionality can be measured against the applicable governing architectures using specific SPARQL queries crafted to determine the presence or absence of conforming relationships.

Through such a means, the architectural conformance of proposed or existing (legacy) systems can be ascertained automatically and unequivocally. It is proposed here that DoD systems be designed in this manner and submitted to conformance checking as part of ongoing investment review of IT systems.

9 Business Intelligence derived from Architecture Descriptions

9.1 Architecture Planning

9.1.1 Defining the Enterprise

In a generic sense, an enterprise is any collection of organizations that has a common set of goals and/or a single bottom line. An enterprise, by that definition, can encompass a Military Department, DoD as a whole, a division within an organization, an organization in a single location, or a chain of geographically distant organizations linked by a common management or purpose. An enterprise today is often thought of as an *extended enterprise* where partners, suppliers, customers, along with their activities and supporting systems, are included in the Architectural Description.

Government agencies may comprise multiple enterprises, and there may be separate enterprise architecture, or Architectural Description projects. However, the projects often have much in common about the execution of process activities and their supporting information systems, and they are all linked an enterprise architecture. The DoD Enterprise Architecture is described in Section 3.1. Architectural description development in conjunction with the use of a common architecture framework, which describes the common elements of Architectural Descriptions, lends additional value to the effort, and provides a basis for the development of an architecture repository for the integration and reuse of models, designs, and baseline data.

9.1.2 The Enterprise-level Architecture

Enterprise-level Architectural Descriptions in DoD are generally created under the responsibility and authority of a senior-level official within the Department, Component, Organization, Agency, or the program office responsible for development of JCAs. As an enterprise-level effort, it is expected that all of the major processes are documented and described, even if a specific project involves only a more limited subset of processes or activities. That way, subsequent Architectural Description efforts can build on previous efforts to ensure the integration and extension of the enterprise is not compromised.

Enterprise-level Architectural Descriptions usually exhibit breadth rather than depth. Since this Architectural Description is the ‘capstone’, or highest level of an Architectural Description, on which others will build, it is especially important that processes, which relate to each other, either through interaction of activities, or the use of data by internal and external stakeholders, are identified or documented.

9.1.3 Solution Architectures

The solution-architecture is scoped to include all major activities that are associated with an identified solution for a capability gap in response to a specific requirement. This solution may contain links to one or programs which require the data and/or outputs produced by the specified the solution identified to fill a specified gap.

9.1.4 Architecture Management

Architectural Descriptions are designed to describe the data on an organization or program/capability that will support continuing managing decision-making over time. Creation of Architectural Descriptions and their management follow an established lifecycle that is similar to those other resources that have well-described lifecycles. OMB Circular A-130[11] describes the lifecycle as:

- Develop
- Use
- Maintain

For consistency, that structure is followed in this volume as well. These phases recognize discreet actions that occur at various times, all designed to ensure that architectural data can be collected and later reused for management decision-making and reporting.

9.1.4.1 Architecture Development

Architectural Descriptions are developed to represent either the state of an activity at a specified time (i.e., baseline architecture) or the results of change in an activity that will occur over some future time (i.e., "To-Be" or future architecture). Enterprise architectures (usually with Departmental, Capability, Segment, or Component content) are initially created to create a common context needed to understand the organization and operations of high-level processes under their control.

Solution Architectural Descriptions collect data that is specific to their program or capability, and data necessary to link to both the higher-level Architectural Descriptions with which they share common parentage, and any lower-level Architectural Descriptions, which describe in more detail particular aspects of the program or JCA.

Visualization of data provides a unique perspective of data from the viewpoint needed for decision-making. That may be a commander/director, action officer, system developer, data administrator, user, or anyone else executing some part of the architected process. More discussion of data collection and visualization is contained in DoDAF Volume II.

9.1.4.2 Architecture Utilization

The ultimate success of an Architectural Description effort lies in the ability to use architectural-related data to support decisions for change within the organization. While Architectural Description development is generally accomplished as a project, accomplished through a team trained for that purpose, the results of the Architectural Description development, to be effective over the longer term, need to be adopted as the common, normal mode of performing the organization's business.

The enterprise architecture, as a corporate asset, should be managed like any other asset, and reinforced by management as a key part of the formal program that results in decision-making. Achieving that level of acceptance occurs only when Architectural Descriptions are created that reflect reality (e.g., baseline), or planned change/growth (e.g., "To-Be", or target).

---

[11] *Office of Management and Budget (OMB) Circular-A-130, Management of Federal Information Resources*, February 8, 1996. Executive Office of the President, Office of Management and Budget. The current version can be found at: http://www.whitehouse.gov/omb/circulars/a130/a130trans4.html#2

Successful execution of the EA development process in an agency-wide endeavor requires management direction and support, allocation of resources, continuity, and coordination. Creating an EA program calls for sustained leadership and strong commitment, buy-in by the agency head, senior leadership, and early designation of a lead architect. These leaders and the supporting EA Team are the first level of support for institutionalizing the results of the effort.

When architectural data and views are constructed and organized in a way that they are understood, accepted, and utilized in daily activities, they facilitate decision-making. To achieve optimal success, architectural views and data must meet standards that facilitate reuse by others whose activities border on, or replicate activities, services and systems already documented by architectural data and products. To that end, data collection must adhere to the standards set by the COI, or other recognized authority so that the data can be registered for, and used by others.

9.1.4.3 Architecture Maintenance

Changes in an organization supported by Architectural Description development will achieve institutionalization only when the senior leadership agrees with, supports, encourages, reinforces, and adopts the results of the Architectural Description effort. Ideally, a member of the Senior Leadership Team should be designated as the 'champion' of the change effort, and should work with the process owner to ensure that institutionalization occurs Employees, who actually perform the daily activities described in the Architectural Description, must be represented in the Architecture Development Team and contribute to the overall data collection and view creation.

9.1.5 Architecture Compliance Reviews

Architectural description compliance reviews are a key part of the validation and verification (V&V) process ongoing throughout the Architectural Description development effort. A *compliance review* is a type of review that analyzes whether Architectural Description developers are progressing according to the specifications and requirements developed for the Architectural Description effort by the process owner. The goals of an architecture compliance review include:

- Identifying errors in the Architectural Description early to reduce the cost and risk of changes required later in the project. These error-catching actions will reduce cost and schedule slips, and will quickly realize business objectives.
- Ensuring the application of best practices to Architectural Descriptions work (Development, use, and maintenance).
- Providing an overview of the compliance of architecture to mandated enterprise standards.
- Identifying and communicating significant architectural gaps to supplier and service providers.
- Communicating to management the status of technical readiness of the project.

Utilization of architecture compliance reviews as an integral part of the development process ensures that utilization of architectural data and views later will be in conformance with applicable requirements. A more in-depth discussion of the compliance review process is contained in the DoDAF Journal.

9.1.5.1 OMB Architecture Assessment

The OMB requires departments and independent agencies to submit a self-assessment of their enterprise architecture programs in February of each year. For DoD, this applies at the Department level. The self-assessment is performed in three EA capability areas: completion of the EA, use of the EA and results, and utilization of the OMB Federal Enterprise Architecture program EA Assessment Framework.12 Specifics of the DoD/OMB architecture self-assessment are described in the DoDAF Journal.

9.1.5.2 GAO Architecture Assessment

The Government Accountability Office (GAO) periodically requires all departments and independent agencies to submit a self-assessment of the maturity of the management of their EA programs. In addition, GAO may perform their own review and assessment of architecture efforts associated with large-scale programs.13 In certain cases, GAO expects an agency to establish an independent quality assurance process for a large-scale architecture to determine whether it meets quality criteria such as those identified earlier in this section.14 Specifics of the DoD/GAO architecture self-assessment are described in the DoDAF Journal. The Enterprise Architecture Management Maturity Framework (EAMMF) (Table 3.2.1.1) can also be used for this purpose.

9.1.6 User Support

User support is the service that each enterprise unit provides its users, both internally and externally to the enterprise, as described in the architectural data and views.

9.1.7 Training

It is the responsibility of agency executive management to institutionalize the control structures for the EA process, as well as for the agency Capital Planning & Investment (CPIC) and Shelf Life Code (SLC) processes. For each decision-making body, all members should be trained, as appropriate, in the EA, the EA process, the relationship of the EA to the Agency's mission, DoDAF, and the FEA. Specific training, at various levels of detail, should be tailored to the architecture role of the personnel.

Architecture development training for team members is often provided by the team leader and Chief Architect during the course of team operations. Training for team members includes sessions on group interactions, toolset operations, data collection, and creation of models and views.

---

[12] *Federal Enterprise Architecture Program: Enterprise Architecture Assessment Framework, version 2.2*, October 2007. Executive Office of the President, Office of Management and Budget. The current version can be found at: http://www.whitehouse.gov/omb/egov/a-2-EAAssessment.html.

[13] *United States Government Accountability Office (GAO) Report: DoD Business Systems Modernization: Long-standing Weaknesses in Enterprise Architecture Development Need to Be Addressed*, July 2005, GAO-05-702. A copy of the report is available at: http://www.gao.gov/new.items/d05702.pdf

[14] *United States Government Accountability Office (GAO) Report: Framework for Assessing and Improving Enterprise Architecture Management, version 1.1*, April 2003, GAO-03-584G. A copy of the report is available at: www.gao.gov/new.items/d03584g.pdf

9.1.8 Communications Planning

Communication management is the formal and informal process of conducting or supervising the exchange of information to all stakeholders of enterprise architecture. Communication planning is the process of ensuring that the dissemination, management, and control of critical stakeholder information is planned and executed in an efficient and effective manner.

The purpose of communications planning is to (1) keep senior executives and business units continually informed, and (2) to disseminate EA information to management teams. The Chief Architect and support staff defines a marketing and communications plan consisting of:

- Constituencies.
- Level of detail.
- Means of communication.
- Participant feedback.
- Schedule for marketing efforts.
- Method of evaluating progress and buy-in.

The CIO's role is to interpret the Agency Head's vision, and recognize innovative ideas (e.g., the creation of a digital government) that can become key drivers in the EA strategy and plan. In turn, the Chief Architect is the primary technical communicator with the communities of interest involved in an Architectural Description effort.

At the Process Owner level, the communications plan is similar to that described above for the CIO. As with the CIO at the enterprise, the process owner is the manager of Architectural Description efforts, supported by an architect and development team. The process owner must clearly define the purpose and scope of an Architectural Description effort (i.e., "Fit-for-Purpose") and communicate those goals and objectives for the Architectural Description effort to the architect and team. In turn, as development of the Architectural Description progresses, the architect provides feedback to the process owner, participates in validation and verification activities, and provides revisions, as required to the original development plan.

9.1.9 Quality Planning

Quality management is the process of organizing activities involving the determination of quality requirements, establishing quality policies, objectives, performance measures (metrics), and responsibilities, and ensuring that these policies, objectives, and measures (metrics) will satisfy the needs within the enterprise. The quality management system executes policies, procedures, and quality planning processes, along with quality assurance, quality control processes, and continuous process improvement activities to improve the overall health and capability of the enterprise. The primary input into the quality management process is quality planning.

Quality planning for Architectural Description development identifies which quality standards are relevant to creation of the Architectural Description and determines how to satisfy them. Quality requirements are stated in the Project Scope Statement, further defined in the Program Management Plan and other guidance, such as that provided by the methodology being applied to the development effort. Guidance also includes other enterprise environmental factors, such as Governmental agency regulations, rules, standards, and guidelines specific to the application area. Information needed during quality planning is generally collected during Architectural Description development, and represented in architectural data and views as controls, resources, inputs, and outputs, as appropriate. A more comprehensive discussion of quality planning is provided online in the DoDAF Journal.

9.1.10 Risk Management

Risk management is the act or practice of dealing with risk. It includes planning for risk, assessing risk issues, developing risk handling strategies, and monitoring risk to determine how they have changed. Risk management planning is the process of deciding how to approach and conduct the risk management activities for the enterprise, program, and projects.

Architectural-based risk assessment is a risk management process that identifies flaws in Architectural Description and determines risks to business information assets that result from those flaws. Through the process of architectural risk assessment, risks are identified and prioritized based on their impact to the business; mitigations for those risks are developed and implemented; and the Architectural Description is reassessed to determine the efficacy of the mitigations.

Risk management planning should be initiated early during development of the scope for the Architectural Description effort. Mitigation of risk is crucial to success of the overall effort. Inputs to the risk management planning process include a review of existing enterprise environmental factors, organizational process assets, the proposed scope statement, and the program management plan. *Enterprise environmental factors* are the attitudes toward risk and the risk tolerance of the organizations and people involved in the organization that exert influence over change. Risk attitudes and tolerances may be expressed in policy statements or revealed in actions. *Organizational process assets* are tools and techniques, which normally predefine organizational approaches to risk management such as established risk categories, common definitions of concepts and terms, standard templates, roles and responsibilities, and authority levels for decision-making.

A comprehensive discussion of Risk management can be found online in the DoDAF Journal.

9.2 Architecture-Based Analytics

Architecture-based analytics includes all of the processes that transform architectural data into useful information in support of the decision making process. Various types of analysis are described below (static vs. dynamic), along with descriptions of desirable characteristics for the overall architectural data set needed for successful and accurate analysis capability. Architectural Descriptions are an ideal construct to use in decision-making since they represent the most current, and accurate information about a program or mission requirement.

9.2.1 Analytics Context

DoDAF V2.0 has been designed to facilitate collection of data usable through quantitative, repeatable, analytical processes to support decisions at all levels of enterprise and/or system engineering. Architectural views (formerly products) are no longer the end goal, but are described solely to facilitate useful access to information. All views are tailorable. The requirements for data completeness and self-consistency within the data schema are more critical than the view chosen at any particular time by a particular user. Analytics, properly conducted, represent a powerful tool for the decision-maker, ensuring that the most appropriate and current, as well as valid data is used for decision-making.

[Figure 10.1-1](#) below, an adaptation of Figure 2-2, from Section 2, illustrates the overall architecting process. More specifically, it illustrates that analytics, the process of doing analysis with and on architectural data, is central to successful decision-making. Analysis defines and describes potential courses of action (i.e., alternatives) that can be considered when considering a mission or program decision.

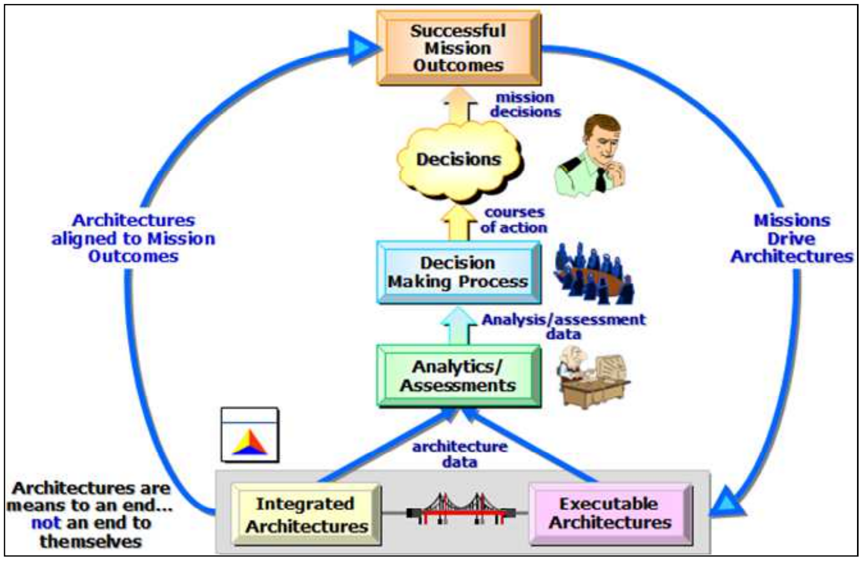
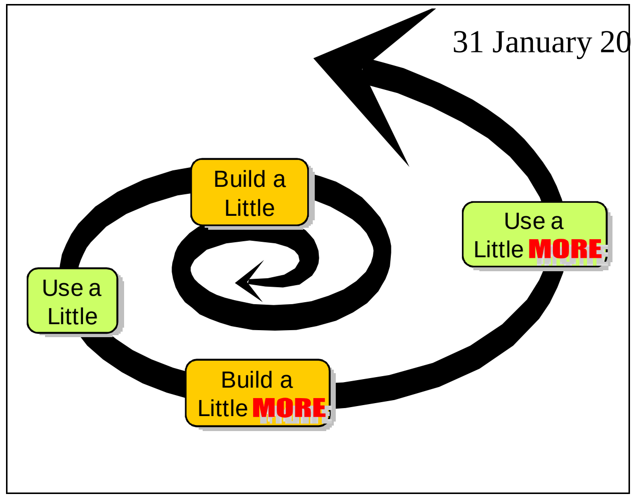


Figure 10.1-1: Analytics Process, Central to Transforming Architectural Data into Usable Forms to Support Decision-Makers

Architecture development is an iterative process, evolved over time. Analyses developed from architectural data remain valid only as long as the processes and information do not change, and management decision-making remains focused on the same problem for which the architectural data was collected. When any of these variables (i.e., architecture purpose, process steps, information, or management direction) change, previous analyses should be reviewed to determine if the previous analysis needs to be redone, based on the newly provided information. Constant feedback and examination needs to be understood as natural in an environment where program direction and priorities are constantly in flux.

Figure 10.1-2: Iterative Approach



The need for an iterative analytical capability points towards tool-assisted and tool-supported analyses whenever possible. Process steps, such as re-running analyses, that are difficult or time consuming to perform will not likely be performed unless automated. The iterative approach, shown in **Figure 10.1-2** of build a little, use a little, build a little, enables Architectural Descriptions to achieve incremental, reachable goals early and throughout the entire architecture lifecycle process.

9.2.2 Architecture Analytic

This is a process that uses architectural data to support decision-making through automated extraction of data from a structured dataset. Automated extraction may be nothing more than results from a query into a database. Architectural Descriptions that are well designed, and consistent with the purpose for which they were created, are also well suited to the analytics process.

9.2.3 Types of Architecture Analysis

There are two categories of analytical activity. These are:

- Static Analyses: Those analyses, which are based on making a value judgment, based on data extracted from the Architectural Description. For example, analysis of the weather patterns and measurements for the last 50 years to determine trends and correlations would be static analyses.
- Dynamic Analyses: Those analyses, which are based on running an executable version of the architectural data to observe the overall behavior of the model. For example, the construction and execution of a dynamic weather prediction model to determine the possible future weather trends is an example of dynamic analysis.

9.2.4 Examples of Analytics

Analytics can be used in conjunction with many aspects of the architecting process. Examples of analytical support can be found within DOTMLPF, as shown in **Table 10.4-1**, below. DOTMLPF is the analysis of who (people, organization, leadership) perform what operations (doctrine) at which locations (facilities) using (training) which system resources (material) to produce and consume information and data. DOTMLPF analysis leads to better definitions of warfighting capabilities by being able to anticipate effects and assess impact of change on domains and by examining usage (who/what affects something) and references (who/what is affected by something). DOTMLPF domains map to DoDAF CDM concepts with the following analytical support activities.

Table 10.4-1: DOTMLPF

DOTMLPF Domains	DoDAF Conceptual Data Model concepts	Analytical Support Activities
Doctrine	Functions, Performers, Assets, Locations	Examine Tactics, Techniques, and Procedures
Organization	Performers, Org Units	Examine organizational structure
Training	Functions, Performers, Assets	Train personnel on their activities and the systems they use
Materiel	Functions, Material, Data, Information, Location, Assets, Performers	Examine materiel solutions – a new system?
Leadership	Org Units, Performers, Assets	Examine leadership issues
Personnel	Performers	Examine personnel solutions – new personnel or personnel with better qualifications
Facilities	Locations	Examine fixing, building, or modifying facilities

It is not the intent for DoDAF to prescribe all possible analytical activities. The list above is only a partial listing of potential activities that relate to DoDAF CDM concepts useful to the DOTMLPF domains. As more demands are placed on architecture, and as industry spawns more automation, the flexibility described in DoDAF will encourage further innovation from architects and from tool vendors.

9.2.5 Principles of Architecture Analytics

The five key foundational principles of architecture analytics are described below. These principles help in maintaining quality Architectural Description and foster further innovation for spawning new analytical activities in the future.

9.2.5.1 Information Consistency

Information consistency means that data (and its derived information) within an Architectural Descriptions is consistent with an overarching metadata structure (called a 'schema'). In addition to adhering to the explicit syntax rules of the schema, data also needs to be consistent with any additional rules specified for the project. Information consistency is often checked to some degree by commercial architecture tools, and additional checking capabilities can be implemented to help assure a more reliable architectural view.

Information consistency also refers to whether the data in one section of the Architectural Description agrees with the data in another section. For instance, if a specific Activity is assigned to a role in one place, yet in another portion of the Architectural Description, that role is shown as not having responsibility for that activity, this would be an information inconsistency. This is normal because the underlying architectural data is found in two or more places. In this case, the tool itself or some configurable process should perform rule-based checks for redundancy to ensure the data in multiple places is consistent. Consistency also involves architecture integration where the underling architectural data is stated only once—one fact, one place—and the architectural views are projections of a single, inherently consistent model.

9.2.5.2 Data Completeness

Data completeness refers to the requirement that all required attributes of data elements are specified. For example, a set of system functions where only some of the functions have associated textual descriptions would not be data complete. Data completeness also refers to the property of having all necessary data to perform certain analyses, view (product/artifact) generation, and/or simulations or executable architectures.

Analytics demands that the architectural data be understandable. Not every analytical procedure will need to examine every part of the Architectural Description. However, no analytical procedure can analyze an Architectural Description that it cannot sufficiently understand, so the Architectural Description's structured dataset needs to be complete enough to support required analytics, thus making it essential that the structured dataset support and define all aspects of the Architectural Description. The architectural model, the projections of the model, and the transformations of the model should, to the extent possible, be based upon open standards. Open standards allow analytics choices

9.2.5.3 Transformation

Many decisions require the use of data contained in datasets created by different toolsets. Utilizing the data for analysis may require a transformation of the data into an alternative structure, which in turn may be accessed by another tool. Transformation allows the intellectual capital invested in the Architectural Description to reach beyond the set of tools used in creating it.

9.2.5.4 Iteration

Analysis needs to support an iterative architecture refinement and decision process (refer to Figure 10.1-1). Analysis that takes too long in any iteration will quickly become irrelevant to the overall process. Rather, small iterative steps or modules should be created that will produce reliable, trustable results.

9.2.5.5 Lack of Ambiguity

An architecturally structured dataset must make clear the meaning of each defined element. If there are semantically variable architectural constructs, they cannot be accurately analyzed by multiple analysis tools. This limits the scope and effectiveness of analytics and therefore limits the usefulness of the architecture itself. Semantic specificity is essential to gain the full benefit of analytics.

9.3 Customer Requirements

In a large organization such as DoD, there are myriad decisions made each day. These decisions require facts (i.e., valid information) for successful execution. Two things affect the ability to make decisions. First, information must be available; second, a decision support process must exist to frame how the decision, once made, can be executed. Decision support can be as simple as an established procedure or rule for execution, or a more complex, integrated set of actions to ensure that a decision is executed properly.

Within DoD are a number of very complex, overarching, decision support services that provide a framework for execution on DoD's most critical program activities. These key DoD change management decision support processes include JCIDS, DAS, SE, PPBE, and PfM. The following paragraphs discuss how these key decision support processes impact management decision making in DoD using architectural data.

9.3.1 Tailoring Architecture to Customers' Needs

Architectural Descriptions are collections of information about an organization that is relevant to a requirement. This information frequently includes processes, supporting systems, needed or desired services, interfaces, business rules, and other details that can be organized to facilitate a decision. From this perspective, Architecture applies a method for tailoring information collection to a specific local need with a clear understanding of the decisions the Architectural Description needs to support, how those decisions should be made, and what information they require. Responding to the organization's requirements generally requires the following information to apply the methodology described in Section 7, or another selected by the architect:

- Detail on specific implementations of the basic processes, including explicit identification of critical decisions mandated or implied.
- Identification of performance measures that can be used to judge the effectiveness of each process (including any mandated by the authoritative documents), taking special note of those that sample the effectiveness of Architectural Description support (the DoDAF Journal includes a tutorial on a relatively painless method for performance engineering).
- For each critical decision, identification of at least one method (and optionally several alternatives) for making that decision, identifying analyses to perform and questions to answer.
- For each analysis or question, identification of needed information.
- Creation of additional business objects/elements and attributes as needed to capture information in the architecture repository.
- Process and information definitions for utilization in Architectural Description development.

The architect simplifies the architectural design by eliminating unneeded objects and attributes through a 'best sense of opportunity' approach, whereby interaction with the customer provides normal and expected needs that generally satisfies the majority of information needs for Architectural Description development. Architectural views should be created to reflect, as closely as possible, the normal 'culture', and preferred presentation design of the agency.

9.3.2 Key Decision Support Processes

Organizations within the DoD may define local change management processes, supportable by Architectural Descriptions, while adhering to defined decision support processes mandated by the Department, including JCIDS, the DAS, SE, PPBE, Net-centric Integration, and PfM. These key support processes are designed to provide uniform, mandated, processes in critical decision-making areas, supplemented by individual agency operations, defined by Architectural Descriptions tailored to support those decisions-making requirements.

9.3.2.1 Joint Capability Integration and Development System

The primary objective of the JCIDS process is to ensure warfighters receive the capabilities required to execute their assigned missions successfully. JCIDS defines a collaborative process that utilizes joint concepts and integrated Architectural Descriptions to identify prioritized capability gaps and integrated joint Doctrine, Organization, Training, Materiel, Leadership and Education, Personnel, and Facilities (DOTMLPF) and policy approaches (materiel and non-materiel) to resolve those gaps.[15] JCIDS implements an integrated, collaborative process to guide development of new capabilities through changes in joint DOTMLPF and policy.

The JCIDS process owners recognized the need for architecture and wrote policy to support architecture requirements (i.e., specific product sets required in specific documents, such as the Information Support Plan, Capability Development Document, and Capability Production Document) that permits components and lower echelon commands to invoke the JCIDS process for requirements at all levels. A more comprehensive discussion of JCIDS is contained in the DoDAF Journal.

9.3.2.2 Defense Acquisition System

The DAS exists to manage the nation's investments in technologies, programs, and product support necessary to achieve the National Security Strategy and support employment and maintenance of the United States Armed Forces.[16] The DAS uses Joint Concepts, integrated architectures, and DOTMLPF analysis in an integrated, collaborative processes to ensure that desired capabilities are supported by affordable systems and other resources.[17]

DoD Directive 5000.1 provides the policies and principles that govern the DAS. In turn, DoD Instruction 5000.2, Operation of the DAS establishes the management framework for translating mission needs and technology opportunities, based on approved mission needs and requirements, into stable, affordable, and well-managed acquisition programs that include weapon systems and automated information systems (AISs).[18] The Defense Acquisition Management Framework[19] provides an event-based process where acquisition programs advance through a series of milestones associated with significant program phases.

The USD (AT&L) leads the development of integrated plans or roadmaps using integrated architectures as its base. DoD organizations use these roadmaps to conduct capability

[15] *Chairman of the Joint Chiefs of Staff (CJCS) Instruction 3170.01F, Joint Capabilities Integration and Development System (JCIDS*), 1 May 2007. A copy of the current version of the instruction can be found at: http://www.dtic.mil/cjcs_directives/cdata/unlimit/3170_01.pdf.

[16] *Department of Defense Directive (DoDD) 5000.1, The Defense Acquisition System*, 12 May 2003 (certified current as of November 20, 2007). A current copy of the directive can be found at: https://akss.dau.mil/dag/DoD5000.asp?view=document&doc=2)

[17] *Department of Defense Instruction (DoDI) 5000.2., Operation of the Defense Acquisition System*. (2003) Under-Secretary of Defense (Acquisition, technology & Logistics) (OUSD AT&L). A current copy of this document can be found at: https://akss.dau.mil/dag/DoD5000.asp?view=document&doc=2

[18] *Department of Defense Instruction (DoDI) 5000.2., Operation of the Defense Acquisition System*. (2003) Under-Secretary of Defense (Acquisition, technology & Logistics) (OUSD AT&L). A current copy of this document can be found at: https://akss.dau.mil/dag/DoD5000.asp?view=document&doc=2

[19] *Integrated Defense Acquisition, Technology, & Logistics Life Cycle Management Framework* (2005). Defense Acquisition University, Ft. Belvoir, VA. A current copy of the chart is found at: http://www.dau.mil/pubs/IDA/IDA_04.aspx

assessments, guide systems development, and define the associated investment plans as the basis for aligning resources and as an input to the Defense Planning Guidance (DPG), Program Objective Memorandum (POM) development, and Program and Budget Reviews.[20]

9.3.2.3 Systems Engineering

DoD Acquisition policy directs all programs responding to a capabilities or requirements document, regardless of acquisition category, to apply a robust SE approach that balances total system performance and total cost with the family-of-systems, and system-of-systems context. Programs develop a Systems Engineering Plan (SEP) for Milestone Decision Authority (MDA) that describes the program's overall technical approach, including activities, resources, measures (metrics), and applicable performance incentives.

SE processes are applied to allow an orderly progression from one level of development to the next detailed level using controlled baselines. These processes are used for the system, subsystems, and system components as well as for the supporting or enabling systems used for the production, operation, training, support, and disposal of that system. Execution of technical management processes and activities, such as trade studies or risk management activities may point to specific requirements, interfaces, or design solutions as non-optimal and suggest change to increase system-wide performance, achieve cost savings, or meet scheduling deadlines21.

Architecture supports SE by providing a structured approach to document design and development decisions based on established requirements.

9.3.2.4 Planning, Programming, Budgeting, and Execution

The PPBE process allocates resources within the DoD and establishes a framework and process for decision-making on future programs. PPBE is a systematic process that guides DoD's strategy development, identification of needs for military capabilities, program planning, resource estimation, and allocation, acquisition, and other decision processes. JCIDS is a key supporting process for PPBE, providing prioritization and affordability advice.

DoDAF V2.0 supports the PPBE process by identifying the touch points between architecture and the PPBE process, identifying the data to be captured within an Architectural Description, facilitating informed decision-making, and identifying ways of presenting data to various stakeholders/roles in the PPBE decision process.

9.3.2.5 Portfolio Management

DoD policy requires that IT investments be managed as portfolios to ensure IT investments support the Department's vision, mission, and goals; ensure efficient and effective delivery of capabilities to the Warfighter; and maximize return on investment within the enterprise. Each portfolio may be managed using the architectural plans, risk management techniques, capability goals and objectives, and performance measures. Capability architecting is done primarily to

[20] *Department of Defense Instruction (DoDI) 5000.2., Operation of the Defense Acquisition System*. (2003) Under-Secretary of Defense (Acquisition, technology & Logistics) (OUSD AT&L). A current copy of this document can be found at: https://akss.dau.mil/dag/DoD5000.asp?view=document&doc=2

[21] *DoD Acquisition Guidebook*. Office of the Under-Secretary for Acquisition, Technology & Logistics (AT&L). A current copy of the Guidebook can be found at: https://akss.dau.mil/dag/DoD5000.asp?view=document&doc=2

support the definition of capability requirements. PfM uses the Architectural Description to analyze decisions on fielding or analysis of a needed capability.22

Architectural support to PfM tends to focus on the investment decision itself (although not exclusively), and assists in justifying investments, evaluating the risk, and providing a capability gap analysis.

9.3.2.6 Operations

In most cases, an enterprise will capture its routine or repeatable business and mission operations as architectural content. However, when the basic structure of an activity is very stable and the activity repeated often, such as military operations planning or project definition and management, the enterprise may choose to include that structure as part of the Architectural Description itself. In this case, the architecture repository may be enhanced to include templates, checklists, and other artifacts commonly used to support the activity.

The JCIDS, PPBE, and DAS processes establish a knowledge-based approach, which requires program managers to attain the right knowledge at critical junctures to make informed program decisions throughout the acquisition process. The DoD IT PfM process continues to evolve that approach with emphasis on individual systems and/or services designed to improve overall mission capability. Consistent with OMB Capital Planning and Investment Control (CPIC) guidance, the DoD uses four continuous integrated activities to manage its portfolios – analysis, selection, control, and evaluation. The overall process is iterative, with results being fed back into the system to guide future decisions.23

9.3.2.7 Net-centric Integration.

Net-centric Integration and interoperability requirements, to include supporting architectural views, are required by CJCSI 6212.01E24. DoDAF V2.0 provides views that support interoperability requirements, both in DoDAF-described Models (including those from previous versions of DoDAF), and new viewpoints, described in Section 3. The DM2 provides data support to interoperability requirements and facilitates creation of user-defined views that meet specific, "Fit-for-Purpose" requirements.

9.3.3 Information Sharing

Information sharing across the Department has existed for many years in various forms. The sharing of information took on new urgency following the events of September 2001, especially in the area of terrorist-related information. Since that time, new Federal legislation[25] and

[22] *Department of Defense Directive (DoDD) 8115.01, Information Technology Portfolio Management*, October 10, 2005. Office of the Assistant Secretary of Defense (Networks & Information Integration) (NII)/DoD Chief Information Officer (DoD CIO). The latest copy of this directive can be found at:

http://www.dtic.mil/whs/directives/corres/rtf/811501x.rtf

[23] DoDD 8115.01, 10.

[24] Chairman of the Joint Chiefs of Staff (CJCS) Instruction 6212.01E, Interoperability and Supportability of Information Technology and National Security Systems, 15 Dec 2008. A copy of the current version of the instruction can be found at: http://www.dtic.mil/cjcs_directives/cdata/unlimit/6212_01.pdf.

[25] Intelligence Reform and Terrorism Prevention Act of 2004 (IRTPA), PL 108-458 (December 17, 2004).

presidential orders require that agencies develop a common framework for the sharing of information, and define common standards for how information is acquired, accessed, shared, and used within a newly created *Information Sharing Environment (ISE).* While initial efforts relate to terrorism-related data, the standards being set could apply, in the future, more broadly across the Department.

Importantly, an Information Sharing Environment Enterprise Architecture Framework (ISE-EAF) is under development[26], which will provide guidance for information collection and dissemination within the Information Sharing Environment (ISE). This Framework is consistent with the DoDAF, and is essential data structures will be mappable to the DM2 described in DoDAF Volumes 2 and 3. When published, that ISE document should be used in coordination with DoDAF to ensure that these specific types of data meet established Federal standards.

9.4 Interoperability via a Semantic BEA

9.4.1 Introduction

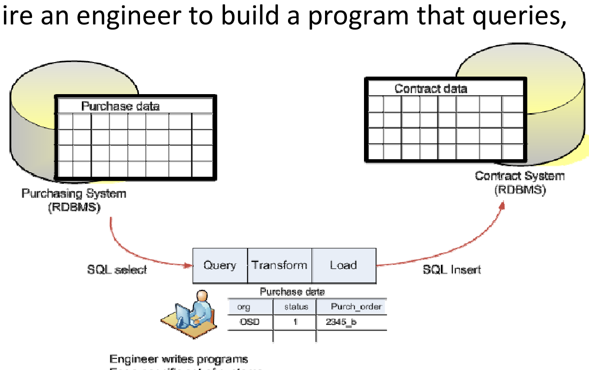
There are two main problems that affect interoperability between systems:

1. Systems usually have different data models so data must be transformed.
2. The system and system process that created data as well as the state of the data is known only to the engineers who built the source system. The data itself contains no information about who or what created it.

For very small systems and for very small numbers of systems, relational technology and XML schema technology can be used effectively and efficiently to achieve interoperability. But as the number of systems increases and as the complexity of systems and business processes increase, relational/XML technology becomes an increasingly difficult way to handle these problems. The cost and complexity of solving them at an enterprise scale is problematic. Semantic technology can be used to solve both of these interoperability problems. A semantic BEA, one that describes the architecture of the DoD enterprise with semantic data, provides the infrastructure to solve these problems.

9.4.2 Traditional Approach to Interoperability (lots of engineering)

As an example, let us consider how data can be brought together from multiple purchasing systems to create a dashboard that provides a strategic view of the state and dependencies of current contracts and efforts. In its simplest form, this would require an engineer to build a program that queries, transforms, and loads the data from one system into another one. For this to work the engineer must:



1. Identify the systems containing the authoritative data
2. Learn the schema of the source systems.

[26] *Information Sharing Environment Enterprise Architecture Framework ()* June, 2008. Office of the Program Manager, Information Sharing Environment.

3. Write the queries and program.
4. Maintain the queries and program if the authoritative data source changes.

Figure 24 Sample Purchase Transaction

One of the first difficulties to this approach is identifying authoritative data sources and then identifying what data within that source fits the need. Both problems rely heavily on the engineers and business analysts of all of the source systems. The BEA as it exists today cannot help solve this problem.

But beyond this initial difficulty, the solution relies heavily on the engineers and business analysts of the new system—the solution will very likely never be part of the BEA. So, even though creation of the new, aggregate view of the data has value to the enterprise, it is merely data that has been created for a new silo. It is not information that exists in a form that can be discovered and used by the enterprise. *Only the users, engineers, and analysts of the new dashboard know what the information is, where it came from, and how to access it in its silo*. This is the story of nearly every business intelligence effort and integration effort within DoD today.

9.4.3 Achieving Interoperability with a Semantic BEA

Now let's focus on how that same very simple fact (from above figure 1)—represented here by the following triple—can be consumed as semantic data and also on the role the Semantic BEA would have in making it real information.

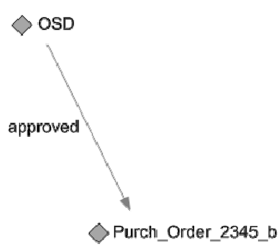


Figure 25 Purchase Transaction as Semantic Data

The triple tells us that something called OSD approved something called Purch_Order_2345_b. We don't know anything more about it than that because we're going to assume that all our knowledge must come from the data itself. We want to consume the data automatically. We want rules that can evaluate the data itself for its suitability to our purpose. And we want a minimum of integration engineering to occur. We are consuming data from many different systems to get a strategic view of the state and dependencies of current contracts and efforts.

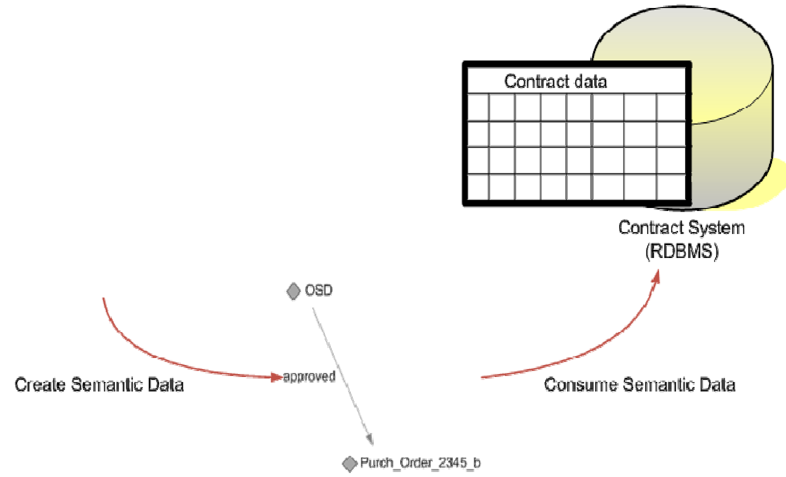


Figure 26 Consuming Semantic Data

So the fact that OSD approved this purchase order looks like something our system can use, but it would require a business analyst to look at where the data comes from (its lineage) to determine whether it is suitable.

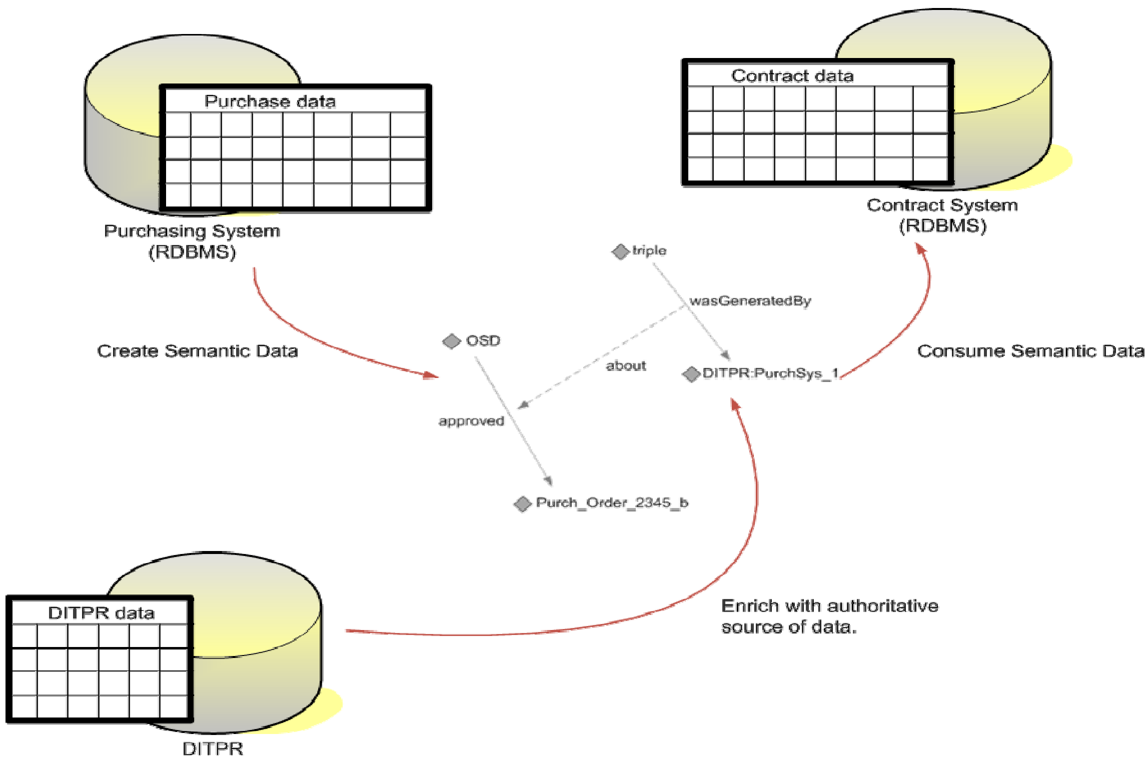


Figure 21 Enriching the Transaction with Provenance

If we are indexing such transactions in a provenance repository, this fact can be queried from the provenance repository to provide a runtime answer about its lineage. Doing this, the triple we began with is enriched with data about what system created it.

Now we know that the fact came from a purchasing system and that this system either is or is not an authoritative data source according to DITPR.

But we still do not have a complete picture of the information. We need to know how the data fits into the overall architecture of systems in the enterprise as well as with what business process it is associated. This too can be queried from our provenance system, and we can use data from the BEA to complete the picture.

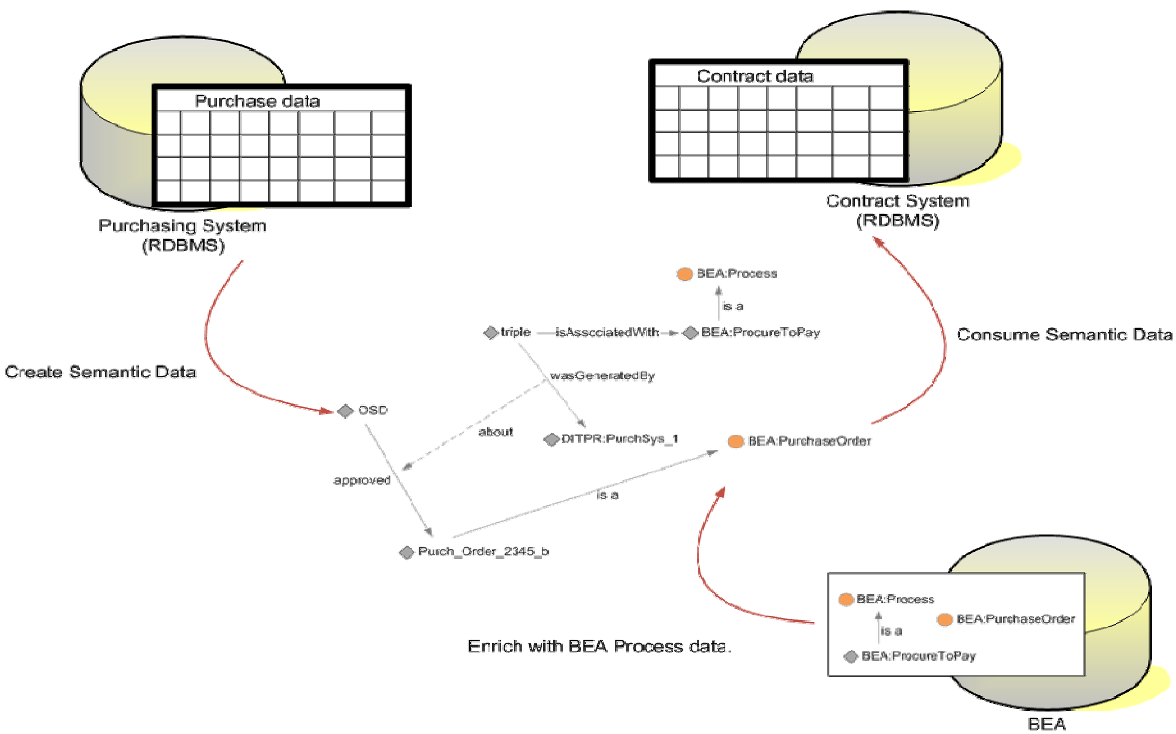


Figure 22 Enriching the Transaction with BEA Data

This is a very simple example that stops with the most general process information. The information provided by our provenance queries could go on to show very specific steps in a business process with which this data is associated.

Having a semantic BEA makes it possible to have information that is machine interoperable and discoverable. By rendering business processes in BPMN and by saving the details of the process information in them as semantic data, BEA data can be used to enrich provenance data about a transaction and facilitate interoperability. But perhaps the most remarkable and useful aspect to this solution is that the new aggregate view of the data can itself be indexed in our provenance repository *and made available for discovery*. One of the greatest sources of waste in DoD is the amount of duplicate work that is done accessing the same data and transforming it. Having a repository in which the new information can be described, along with detailed information about the full lineage of the information, makes it possible to effectively consume information without duplicate engineering efforts.

The figures above do not show the provenance repository. Instead they show a graph of the information that exists inside the provenance repository. But the repository is an actual database for storing and querying graphs.

9.4.4 Multiple Data Models

We find different data models throughout DoD because each system is built around a specific problem and the problem drives the data model. This has become so ingrained in the way we think about building systems that developing a new model seems the natural place to start in every new development effort. So for every new system we get a new model—which has nothing at all to do with whether or not the data going into the model is unique to that system. For all the various purposes for which a piece of information must be used, there is a model for each purpose and custom software to do the translation between each model. This engineering is harder and more fragile than it sounds at first blush: if we move data from system A to system B to system C, the translation software built to go from A to B and from B to C is unique "one-of" integration code. It cannot be used for any purpose than to translate data from one system to another.

This problem led to the development of standard data models (such as NIEM). The reasoning behind this is that we can perhaps re-use more code if each system uses a standard data model. This works within a very narrowly defined problem scope, but it begins to break down as the *information* is used in other problems. This happens because standard data models are, necessarily, aggregate models of all the problem domains they serve. This makes them larger and more complex than any model for a single system. But the greatest problem this introduces is that systems built around a standard data model become closely coupled to it. The more systems that are integrated using a shared model, the more difficult and expensive it becomes to change the model—because it is therefore necessary to change every system that uses it. But change is necessary. Systems must adapt as new information is needed to drive mission systems.

Here too is where the Semantic BEA provides a solution. Semantic technology is not based on data models. It provides a description of information that can be translated into and out of data models. The Semantic BEA would provide the infrastructure to describe information in a way that separates it from data models. It enables information to be described and translated as it exists in multiple data models. Engineers would not need to write code to translate data from system A to system B. Instead, the BEA gives them a description of the data in system A as well as a description of the model used to store it in system A. The engineer can then write or reuse code that translates the data from the model into data described by the semantics of the enterprise. Once data has been translated in this way, it can be consumed by any system. Engineers would write code to translate to and from a data model and semantic data—a semantic information model. This enables standard data models to be used safely and without adversely affecting the department's ability to adapt. It also creates a very efficient infrastructure for code re-use. The job of writing translation software from a standard data model to semantic data only needs to be done once. It is not specific to a system.

The Semantic BEA would provide the architecture and at least some of the infrastructure needed to make this possible. It would provide a description of which ontologies and data models are associated with the business processes. It could also provide a description of the processes (or services) that exist to manage translation to and from a semantic description of the information.

9.4.5 Conclusion

The DoD has attempted for years to achieve interoperability by brute force engineering—writing code that translates data directly from one system to the next. There is not currently an architecture that describes the state of information in the various systems, and there is not currently an architecture that provides a description of how that data is stored and used. Both of these are needed to achieve interoperability. For DoD to be able to easily consume information from around the enterprise, there must be an architectural description of what the data in each system means and how it is stored. And that architecture needs to be queryable and machine operable. The Semantic BEA can provide this.

9.5 The Business Value of Semantic Technology

9.5.1 Introduction

If you don't understand what your software engineers are talking about, perhaps it's because they are using a vocabulary they invented for the problem they are solving. Engineers invent a vocabulary and data structure for each system they build and each problem they solve, and only the engineers who built the system understand this structure and vocabulary. Even other engineers must learn it in order to make the data usable. At the Department of Defense (DoD), we have as many different ways to ask questions of our data as we have systems to store it. We have as many different vocabularies and data structures as we have systems.

The problem is actually worse than it sounds. If we want to bring data together from many different systems or take data out of one system and put it into another one, we need to understand the vocabularies and structures of each and every system involved. That can be very difficult and time-consuming. The meaning of the data is supplied by the program(s) an engineer writes for the data. So, a new engineer looking at the data must generally understand the program in order to fully understand the data. Effectively, each system we build becomes the fiefdom of the engineers who build it. And each system becomes a silo. Combining data from multiple systems requires the time and cooperation of the engineers who maintain each system involved. This isn't deliberate on the part of engineers. It is a consequence of the way we have designed systems over the past twenty to thirty years and the technologies available at the time.

Semantic technology solves this problem by embedding the meaning of data in the data itself and by making it possible for different systems to use the same meaning and the same vocabularies. In traditional systems, sharing vocabulary and meaning is not practical. We must ask questions in terms of the structure of the data and the structure of the data comes directly from the problem the system has been built to solve. Each problem is different, so each structure is different. In contrast, semantic technology is not based upon data structure. In fact, semantic data has no pre-defined structure. What structure there is comes from the data itself and the relationships between the facts and things in the data. Today, very mature and established tools and methodologies exist for building systems entirely with semantic technology.

If that sounds futuristic, consider that if you call the customer service to complain about your phone bill, chances are good that that representative is using a business system built upon semantics. Or, if you use Google to search for a new cell phone, chances are you will be given a list of web pages that have been encoded with semantics. Google, Best Buy, and other large companies have already adopted semantic technology and are pushing it into our daily lives. Google has become a lot better at providing search results? That is semantics in action. Semantics enable these businesses to combine information much more quickly and economically.

9.5.2 What is Semantic Technology

Semantic technology is based upon data stored in a graph, referred to as *graph data*, and a description of what that data means. As the name implies, semantic technology stores meaning with the data. It also removes the need to define structure, or data models. There is no pre-defined structure to which the data that is described semantically must be bound. This provides tremendous flexibility in what information can be stored, and it enables information to be combined and used both rapidly and in ways that are not possible with relational or traditional XML technology (structured data).

Purchase data

org	status	Purch_order
OSD	1	2345_b

Figure 1 Structured Data

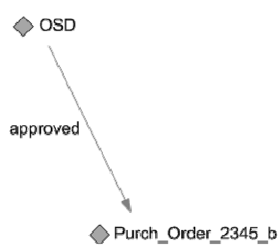
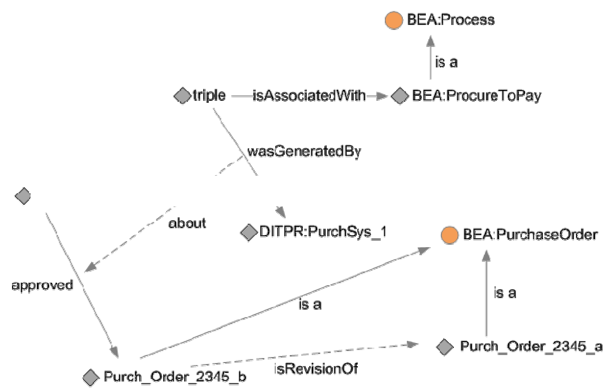


Figure 2 Semantic data

The two figures above illustrate the difference. Figure 1 displays data as stored in a relational database—structured data. In order to turn this structured data into information, an engineer must write an application to interpret it, such as what a status code of “1” means. Figure 2 depicts a piece of information that says OSD approved something called Purchase_Order_2345_b.

The meaning of the data in Figure 2 is stored with the data itself, thereby making it information. And if we want to improve upon our understanding of it, we can simply add more information. It isn't necessary to redesign a data model—there is no data model. To use this information, an application can simply consume it. An engineer does not need to write code to tell the application what it means or how to consume it. Because it contains meaning, it is information, not merely data.



Computer systems have traditionally been built upon data models. The models are built to satisfy the requirements for a specific system that have been developed in response to an identified need or problem. A data model serves to constrain data to a certain structure, and application logic is built upon this structure. Meaning comes from the application logic. There is no meaning stored with the data itself. There is, in fact, no information until the data is consumed by the application. Looking at the landscape of systems at DoD today, *most of the data is siloed because only the application(s) built specifically for it can turn the data into information*. Although no engineer sets out to create a silo, silos are the unavoidable consequence of data models—even standard data models.

Semantic technology avoids this consequence by putting meaning into the data. Rather than data models, the data is stored as a graph and the graph is self-describing. The only role an application has is to query and serve up the information to a user. The application is no longer a silo because the meaning of the data is not supplied by the application.

9.5.3 The Cost of Data

One consequence of traditional data modeling is that in order to use the data to meet a variety of needs, it must be stored in a variety of different structures and must be constantly translated, copied, and kept in sync. One user's need is almost always at least slightly different from another user's need, and need and intended use affect structure.

Data and data modeling incur very high engineering costs. Most of the cost of systems today and most of the IT budget at DoD comes not from modernization or new development but from maintaining data and the applications that make the data usable. It comes from maintaining all the different copies of data that are created for each intended use. It comes from maintaining all the code that copies data from one system to another. It comes from maintaining a staff whose purpose is simply to understand what the data means and how it is shuttled between systems. And it comes from the need to change a myriad of structures to accommodate any new information needs.

With semantic technology, how information is used does not affect how it is stored. The use does not affect its structure. There is no need to keep the information in a myriad of structures in order to satisfy a myriad of intended uses. It can be stored in one place to meet any number of intended uses. Semantic technology removes the need to move data in and out of authoritative sources. The data itself becomes authoritative information in the one place that it is stored. Because all applications can use the same piece of information, semantic technology removes the need to maintain a staff whose purpose is simply to "keep the silo operating."

9.5.4 Information

Information is one of the largely unsung inventions of science fiction. Flip phones, lasers, rockets—all accepted without a blink. But whenever we, as engineers, watch the hero effortlessly consume the information from a database, we wonder how that was possible. There was no design stage, no analysis, no development, no compiling, no testing, no debugging. It simply worked. This is not possible when an application is built to consume *data*. But it is possible when an application has been built to consume *information*. Once an application has been built around information, the mechanics of consuming it never change. It is the mechanics of consuming *data* that require so much design and development, so much analysis and testing and debugging.

Information has tremendous value. The business value of semantics is the value of information itself and the reduction in the cost of using that information. The value of semantics is in all of the new uses to which the information can be put to when we no longer need to invest in operating the silos. It is a reduction in complexity, a reduction in operating cost, a reduction in the sheer amount of storage and computing capacity, a better use of talent, and a leap forward in our ability to further automate what we do.

9.5.5 Conclusion

If you don't understand what software engineers are saying, perhaps it's because they are not creating information. They are engaged in the mechanics of designing, building, and integrating the data stored in their silos. The portfolio of systems at DoD contains an enormous amount of siloed data and relatively little usable information—a cause of many of the problems facing

DoD today. This makes interoperability and maintenance both expensive and complex, and it makes modernization and the realization of new capability elusive. Semantic technology offers a solution to these problems. It makes information available and less expensive, it enables more and better use of information, and it enables new capability and real modernization.

10 Transitioning from DoDAF 1.5 to DoDAF 2.0

The architectures for DoDAF V1.0 and DoDAF V1.5 may continue to be used. When appropriate (usually indicated by policy or by the decision-maker), DoDAF V1.X architectures will need to update their architecture. When pre-DoDAF V2.0 architecture is compared with DoDAF V2.0 architecture, concept differences (such as Node) must be defined or explained for the newer architecture.

In regard to DoDAF V1.5 products, they have been transformed into parts of the DoDAF V2.0 models. In most cases, the DoDAF V2.0 Meta-model supports the DoDAF V1.5 data concepts, with one notable exception: Node. As explained in Section 1.5 of V2.0, Node is a complex, logical concept that is represented with more concrete concepts. **Table 3.2-1** indicates the mapping of DoDAF V1.5 products to DoDAF V2.0 models.

Table 3.2-1: Mapping of DoDAF V1.5 Products to DoDAF V2.0 Models

DoDAF V2.0 / DoDAF V1.5	Operational Viewpoint	Systems Viewpoint	Services Viewpoint	All Viewpoint	Standards Viewpoint	Data & Information Viewpoint
AV-1				AV-1		
AV-2				AV-2		
OV-1	OV-1					
OV-2	OV-2					
OV-3	OV-3					
OV-4	OV-4					
OV-5	OV-5a, OV-5b					
OV-6a	OV-6a					
OV-6b	OV-6b					
OV-6c	OV-6c					
OV-7						DIV-2
SV-1		SV-1	SvcV-1			
SV-2		SV-2	SvcV-2			
SV-3		SV-3	SvcV-3a,			

DoDAF V2.0 / DoDAF V1.5	Operational Viewpoint	Systems Viewpoint	Services Viewpoint	All Viewpoint	Standards Viewpoint	Data & Information Viewpoint
			SvcV-3b			
SV-4a		SV-4				
SV-4b			SvcV-4			
SV-5a		SV-5a				
SV-5b		SV-5b				
SV-5c			SvcV-5			
SV-6		SV-6	SvcV-6			
SV-7		SV-7	SvcV-7			
SV-8		SV-8	SvcV-8			
SV-9		SV-9	SvcV-9			
SV-10a		SV-10a	SvcV-10a			
SV-10b		SV-10b	SvcV-10b			
SV-10c		SV-10c	SvcV-10c			
SV-11						DIV-3
TV-1					StdV-1	
TV-2					StdV-2	

Architectural data will need to be exchanged between Architecture tools. Architectures developed in accordance with DoDAF V1.0 or V1.5 may need to exchange data with Architectures developed in accordance with DoDAF V1.0, V1.5, and V2.0.

DoDAF V1.0 and V1.0 architectures that use the Node concept will need to update the architecture to express the concrete concepts in place of the abstract concept that Node represents. When pre-DoDAF V2.0 architecture is compared with DoDAF V2.0 architecture, the concrete concepts that Node represents must be defined for the newer architecture.

Table 2.2-1 clarifies actions to be performed when exchanging information between Architectures developed on same or different versions of DoDAF.

Table 9.5-1: Exchange Actions between Architectures

Architecture Source	Architecture Target	Actions
DoDAF V1.0 or V1.5	DoDAF V1.0 or V1.5	Use CADM as the exchange basis.
DoDAF V1.0 or V1.5	DoDAF V2.0	Determine the DoDAF V2.0 concepts of the Nodes in DoDAF V1.0 or V1.5 Architecture. Export the DoDAF V1.0 or V1.5 architectural data. As a step of the export, transform the DoDAF V1.0 or V1.0 Node concept into the appropriate DoDAF V2.0 concepts using DoDAF PES Import the architectural data in accordance to the PES into DoDAF V2.0 Architecture.
DoDAF V2.0	DoDAF V1.0 or V1.5	Determine the DoDAF V2.0 concepts of the Nodes in DoDAF V1.0 or V1.5 Architecture Export the DoDAF V2.0 architectural data. As a step of the export, transform the appropriate DoDAF V2.0 concepts into the appropriate DoDAF V1.0 or V1.0 Node concept. Import the architectural data in PES format into DoDAF V1.0 or V1.5 Architecture. Transformation into CADM format may be required.
DoDAF V2.0	DoDAF V2.0	Use PES as the exchange basis.

Appendix A: Acronyms

Acronym	Definition
ADM	Architecture Development Method
AMETL	Agency Mission Essential Task List
ASD	Assistant Secretary of Defense
AT&L	Acquisition Technology and Logistics
AV	All Viewpoint
AV-2	All Viewpoint 2: Integrated Dictionary
BEA	Business Enterprise Architecture
BMM	Business Motivation Model
BPMN	Business Process Modeling Notation
BPR	Business Process Reengineering
BRM	Business Reference Model
BT	Business Transformation
C4I	Command, Control, Communications, Computers and Intelligence
C4ISRAF	Command, Control, Communications, Computers, and Intelligence Surveillance Reconnaissance Architecture Framework
CADM	Core Architecture Data Model
C.A.R.P.	Capability, Activity, Resource, Performer
CCB	Configuration Control Board
CCP	Configuration Control Plan
CDD	Capability Development Document
CDM	Conceptual Data Model
CIO	Chief Information Officer
CJCSI	Chairman of the Joint Chiefs of Staff Instruction

Acronym	Definition
CJCSM	Chairman of the Joint Chiefs of Staff Manual
CM	Configuration Management
COI	Community Of Interest
COMSEC	Communications Security
CONOPS	Concepts of Operations
CPD	Capability Production Document
CPIC	Capital Planning and Investment Control
CPM	Capability Portfolio Management
CRM	Consolidated Reference Model
CV	Capability Viewpoint
CWID	Coalition Warrior Interoperability Demonstration
DAES	DoD Architecture Enterprise Services
DARS	DoD Architecture Registry System
DAS	Defense Acquisition System
DDMS	DoD Discovery Metadata Specification
DIEA	DoD Information Enterprise Architecture
DISR	DoD Information Technology Standards and Profile Registry
DITPR	DoD Information Technology Portfolio Repository
DIV	Data and Information Viewpoint
DM2	DoDAF Meta-model
DMR	DoD Metadata Registry
DoD	Department of Defense
DoDAF	DoD Architecture Framework

Acronym	Definition
DoDI	Department of Defense Instruction
DOTMLPF	Doctrine, Organization, Training, Materiel, Leadership and Education, Personnel, and Facilities
DPG	Defense Planning Guidance
DRM	Data Reference Model
EA	Enterprise Architecture
EAAF	Enterprise Architecture Assessment Framework
EAMMF	Enterprise Architecture Management Maturity Framework
EIA	Electronic Industries Alliance
E-ISP	Enhanced-Information Support Plan
FEA	Federal Enterprise Architecture
FEA-CRM	Federated Enterprise Architecture-Consolidated Reference Model
FEA-RM	Federal Enterprise Architecture Reference Model
FIPS	Federal Information Processing Standard
FISMA	Federal Information Security Management Act
GAO	Government Accountability Office
GIG	Global Information Grid
IC	Intelligence Community
ICD	Initial Capabilities Document
IDEAS	International Defence Enterprise Architecture Specification
IDEF0	Integration Definition for Activity Modeling
IE	Information Environment
IEA	Information Enterprise Architecture
INFOSEC	Information Security

Acronym	Definition
IP	Internet Protocol
IRTPA	Intelligence Reform and Terrorism Prevention Act of 2004
ISE	Information Sharing Environment
ISE-EAF	Information Sharing Environment Enterprise Architecture Framework
ISM	Information Security Marking
ISO	International Standards Organization
IT	Information Technology
ITS/NSS	Information Technology/National Security Systems
JCA	Joint Capability Area
JCIDS	Joint Capability Integration and Development System
JCPAT	Joint C4I Program Assessment Tool
JCS	Joint Chiefs of Staff
JCSFL	Joint Common System Function List
JFCOM	Joint Forces Command
JMETL	Joint Mission Essential Task List
KM/DS	Knowledge Management/Decision Support
LDM	Logical Data Model
M3	MODAF Meta Model
MDA	Milestone Decision Authority
MDR	Metadata Registry
MOD	Ministry of Defence
MODAF	Ministry of Defence Architecture Framework
NAERG	Naval Architecture Elements Reference Guide

Acronym	Definition
NATO	North Atlantic Treaty Organization
NCDS	Net Centric Data Strategy
NCE	Net-Centric Environment
NCSS	Net-Centric Services Strategy
NII	Networks and Information Integration
NIST	National Institutes for Standards & Technology
NSS	National Security Systems
OASD	Office of the Assistant Secretary of Defense
OASIS	Organization for the Advancement of Structured Information Standards
OMB	Office of Management and Budget
OMG	Object Management Group
OO	Object-Oriented
OOAD	Object-Oriented Analysis & Design
OSD	Office of the Secretary of Defense
OUSD	Office of the Undersecretary of Defense
OV	Operational Viewpoint
PDA	Personal Digital Assistant
PDCA	Plan, Do, Check, and Act
PDM	Physical Data Model
PES	Physical Exchange Specification
PFD	Process Flow Diagram
PL	Public Law
POM	Program Objective Memorandum

Acronym	Definition
PPBE	Planning, Programming, Budgeting, and Execution
PRM	Performance Reference Model
PTD	Process Task Dependency
PV	Project Viewpoint
RA	References Architecture
RM	Reference Model
SADT	Structured Analysis and Design Technique
SE	Systems Engineering
SEP	Systems Engineering Plan
SIPRNET	Secret IP Router Network
SLC	Shelf Life Code
SOA	Service-Oriented Architecture
SRM	Service Component Reference Model
SV	Systems Viewpoint
SvcV	Services Viewpoint
TA	Tiered Accountability
TAFIM	Technical Architecture for Information Management
TEMPEST	Transient Electromagnetic Pulse Emanation Standard
TOGAF	The Open Group Architecture Framework
TRM	Technical Reference Model
TV	Technical Standards View
TWG	Technical Working Groups
U.S.	United States

Acronym	Definition
UJTL	Universal Joint Task List
UK	United Kingdom
UML	Unified Modeling Language
UPDM	Unified Profile for DoDAF and MODAF
URL	Uniform Resource Locator
USD	Under Secretary of Defense
V&V	Validation & Verification
WBS	Work Breakdown Structure
XML	eXtensible Markup Language
XSD	XML Schema Definition

Appendix B: DoDAF Journal Glossary

BEA: Business Enterprise Architecture. The BEA is the enterprise architecture for the DoD BMA and reflects the DoD business transformation priorities; the business capabilities required to support those priorities; and the combinations of enterprise systems and initiatives that enable those capabilities.

Behavioral Rule: governs the behavior/actions associated with a particular concept.

BMM: Business Motivation Model. The BMM is an OMG business modeling specification that provides a scheme or structure for developing, communicating, and managing business plans in an organized manner.

BPM: Business Process Management. BPM is a holistic management approach focused on aligning all aspects of an organization with the wants and needs of clients. It promotes business effectiveness and efficiency while striving for innovation, flexibility, and integration with technology.

BPMN: Business Process Modeling Notation. BPMN provides businesses with the capability of defining and understanding their internal and external business procedures through a Business Process Diagram, which will give organizations the ability to communicate these procedures in a standard manner.

BPMS: Business Process Management System. A BPM system may comprise a variety of independent packages or a comprehensive business process management suite (BPMS), which includes tools for modeling and analysis, application integration, business rules, business intelligence (BI), activity monitoring and optimization. Advanced BPMSs provide a development tool for creating forms-based applications, which are often the start of many business processes.

BMA: Business Mission Area. The BMA ensures that the right capabilities, resources, and materiel are reliably delivered to our warfighters: what they need, where they need it, when they need it, anywhere in the world. In order to cost-effectively meet these requirements, the DoD current business and financial management infrastructure - processes, systems, and data standards - are being transformed to ensure better support to the warfighter and improve accountability to the taxpayer. Integration of business transformation for the DoD business enterprise is led by the Deputy Secretary of Defense in his role as the Chief Operating Officer of the Department.

Business Rule: refers to an operational rule corresponding to a high-level law, regulation or policy in terms of the business analyst; also referred to informally as 'What' rules.

Domain: defines a realm of administrative autonomy, authority, or control.

DoD: Department of Defense. The federal department responsible for safeguarding national security of the United States; created in 1947.

DoDAF: Department of Defense Architecture Framework. DoDAF Version 2.0 is the overarching, comprehensive framework and conceptual model enabling the development of architectures to facilitate the ability of Department of Defense (DoD) managers at all levels to make key decisions more effectively through organized information sharing across the Department, Joint Capability Areas (JCAs), Mission, Component, and Program boundaries.

DM2: DoDAF Meta Model. The DM2 defines architectural data elements and enables the integration and federation of Architectural Descriptions. It establishes a basis for semantic (i.e., understanding) consistency within and across Architectural Descriptions.

Federation: multiple domains that are linked (federated) through relationships identified between concepts defined in the different domains.

MDI: Model Driven Implementation. A methodological approach designed to achieve architectural round-tripping through the use of Semantic Technology as part of its modeling, solution generation and runtime phases. Architectural round-tripping is translation of information from models to executable code, and the propagation of changes at the executable level back to the conceptual models.

OMG: Object Management Group. A consortium, originally aimed at setting standards for distributed object-oriented systems, and is now focused on modeling (programs, systems and business processes) and model-based standards.

Operational Viewpoint: models that describe the tasks and activities, operational elements, and resource flow exchanges required to conduct operations.

OV-6a: DoDAF-described Operational Rules Model. One of three models used to describe activity (operational activity). It identifies business rules that constrain operations.

OV-6c: DoDAF -described Operational Event Trace Model. One of three models used to describe activity (operational activity). It traces actions in a scenario or sequence of events.

OWL: Web Ontology Language. The OWL 2 Web Ontology Language, informally OWL 2, is an ontology language for the Semantic Web with formally defined meaning. OWL 2 ontologies provide classes, properties, individuals, and data values and are stored as Semantic Web documents. OWL 2 ontologies can be used along with information written in RDF, and OWL 2 ontologies themselves are primarily exchanged as RDF documents.

Production Rule: refers to a representation of a run-time system/service level rule in terms executable by a rules engine; also referred to informally as ‘How’ rules. See also ‘Technical Rule’.

RIF: Rule Interchange Format. A W3C standard for exchanging rules among rule systems, in particular among Web rule engines.

Round Trip Engineering: closely related to forward engineering (creating software from specifications), reverse engineering (creating specifications from existing software), and reengineering (understanding existing software and modifying it). The key characteristic of round-trip engineering is the ability to synchronize existing artifacts that evolved concurrently by incrementally updating each artifact to reflect changes made to the other artifacts.

Rule: a principle or condition that governs behavior; a prescribed guide for conduct or action; a regulation or bylaw governing procedure or controlling conduct; a constraint that governs.

SBVR: Semantics of Business Vocabulary and Business Rules. SBVR is an adopted standard of the Object Management Group (OMG) intended to be the basis for formal and detailed natural language declarative description of a complex entity, such as a business. SBVR is intended to formalize complex compliance rules, such as operational rules for an enterprise, security policy, standard compliance, or regulatory compliance rules. Such formal vocabularies and rules can be interpreted and used by computer systems. SBVR is an integral part of the OMG's Model Driven Architecture (MDA).

System/Service Viewpoint: models that describe systems, services, and their interconnections that associate systems/services resources to the operational requirements.

Structural Rule: governs the relationship between concepts.

SV-10a: DoDAF-described System Rules Model. One of three models used to describe system functionality. It identifies constraints that are imposed on systems functionality due to some aspect of system design or implementation.

SV-10c: DoDAF -described System Event Trace Model. It identifies system-specific refinements of critical sequences of events described in the Operational Viewpoint.

Technical Rule: refers to a representation of a run-time system/service level rule in terms executable by a rules engine; also referred to informally as 'How' rules. See also 'Production Rule'.

UML: Unified Modeling Language. Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

Vocabulary: the set of defined words (concepts) used by or known to a particular group of persons. When defined using the Web Ontology Language (OWL) this is represented as an ontology.

W3C: World Wide Web Consortium. The World Wide Web Consortium (W3C) is an international community that develops standards to ensure the long-term growth of the Web.

Appendix C: References/Bibliography

ANSI/GEIA Standard EIA 649-A National Consensus Standard for Configuration Management American National Standards Institute.

Chairman of the Joint Chiefs of Staff (CJCS) Instruction 3170.01F, Joint Capabilities Integration and Development System (JCIDS), 1 May 2007.

Chairman of the Joint Chiefs of Staff (CJCS) Instruction 6212.01E, Interoperability and Supportability of Information Technology and National Security Systems*,* 15 Dec 2008.

Department of Defense Directive (DoDD) 5000.1, The Defense Acquisition System, 12 May 2003 (certified current as of November 20, 2007).

Department of Defense Directive (DoDD) 8115.01, Information Technology Portfolio Management, October 10, 2005. Office of the Assistant Secretary of Defense (Networks & Information Integration) (NII)/DoD Chief Information Officer (DoD CIO).

Department of Defense Instruction 4630.8, Procedures for Interoperability and Supportability of Information Technology (IT) and National Security Systems (NSS) 30 June 2004. Office of the Assistant Secretary of Defense (Networks & Information Integration) (NII)/ DoD Chief Information Officer (DoD CIO).

Department of Defense Instruction (DoDI) 5000.2., Operation of the Defense Acquisition System (2003). Under-Secretary of Defense (OUSD) AT&L (Acquisition, Technology & Logistics).

Department of Defense Net-Centric Data Strategy, 9 May, 2003. Office of the Assistant Secretary of Defense (Networks & Information Integration) (NII)/DoD Chief Information Officer (DoD CIO).

Dinoli, D. (2008) *Enterprise Architecture A to Z: Frameworks, Business Process Modeling, SOA, and Infrastructure Technology.* Auerbach Publishing. 504pp.

DoD Acquisition Guidebook. Office of the Under-Secretary for Acquisition, Technology & Logistics (AT&L).

Federal Enterprise Architecture, Executive Office of the President, Office of Management and Budget E-Gov Initiative.

Federal Enterprise Architecture Consolidated Reference Model Version 2.3. Executive office of the President, Office of Management and Budget (OMB).

Federal Enterprise Architecture Program: Enterprise Architecture Assessment Framework, version 2.2, October 2007. Executive Office of the President. Office of Management and Budget.

Federal Enterprise Architecture Records Management Profile, Version 1.0, December 15, 2005. Executive Office of the President, Office of Management and Budget.

Global Information Grid (GIG) Architecture Federation Strategy, 1 August 2007. Office of the Assistant Secretary of Defense (Networks & Information Integration) (NII)/DoD Chief Information Officer (DoD CIO).

Grigoriu, A. (2007) An *Enterprise Architecture Framework*. Privately Published 226.pp.
Information Sharing Environment Enterprise Architecture Framework () June, 2008. Office of the Program Manager, Information Sharing Environment.

Integrated Defense Acquisition, Technology, & Logistics Life Cycle Management Framework (2005). Defense Acquisition University, Ft. Belvoir, VA.

McGovern, J, Ambler, S, Stevens, M. E., Linn, J, Sharan, V & Jo, E. K. (2004) A *Practical Guide to Enterprise Architecture*. New Jersey: Prentice-Hall. 306pp.

Intelligence Reform and Terrorism Prevention Act of 2004 (IRTPA), Public Law (PL) 108-458 (December 17, 2004)

Ministry of Defence Architecture Framework. Ministry of Defence of the United Kingdom. The latest electronic version of the MODAF can be found at the MODAF Website: *http://www.modaf.org.uk/*.

NATO Architecture Framework for C3 Systems (2006). The North Atlantic Treaty Organization.

Office of Management and Budget (OMB) Circular-A-130, Management of Federal Information Resources, February 8, 1996. Executive Office of the President. Office of Management and Budget.

Ross, J. W., Weill, P., & Robertson, D. C. (2006). *Enterprise Architecture as Strategy*. Boston, MA: Harvard Business School press. 235pp.

Ross, R. (2005). *Business Rule Concepts* (2nd. Ed.) Business Rules Forum. 135pp.

Ross, R. (2003). *Principles of the Business Rule Approach*. Boston, MA: Addison-Wesley Publishing. 372pp.

Schekkerman, Jaap (2004). *How to Survive in the Jungle of Enterprise Architecture Frameworks*. Victoria, BC: Trafford 222. pp.

Spewak, Steven H (1992). *Enterprise Architecture Planning*. New York: John Wiley & Sons. 368 pp.
The Open Group Architecture Framework, Version 8.11

United States Government Accountability Office (GAO) Report: DoD Business Systems Modernization: Long-standing Weaknesses in Enterprise Architecture Development Need to Be Addressed, July 2005, GAO-05-702.

United States Government Accountability Office (GAO) Report: Framework for Assessing and Improving Enterprise Architecture Management, version 1.1, April 2003, GAO-03-584G.

Zachman, John. Zachman Framework. © Zachman International.