Date: July 2022

UAF®

OMG UNIFIED
ARCHITECTURE
FRAMEWORK ®

Enterprise Architecture Guide for UAF (Informative)

Appendix C

Version 1.2

OMG Document Number: formal/22-07-10

Standard document URL: https://www.omg.org/spec/UAF/1.2

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 9C Medway Road, PMB 274, Milford, MA 01757, U.S.A.

TRADEMARKS

IMM®, MDA®, Model Driven Architecture®, UML®, UML Cube logo®, OMG Logo®, CORBA® and XMI® are registered trademarks of the Object Management Group, Inc., and Object Management Group™, OMG™ , Unified Modeling Language™, Model Driven Architecture Logo™, Model Driven Architecture Diagram™, CORBA logos™, XMI Logo™, CWM™, CWM Logo™, IIOP™ , MOF™ , OMG Interface Definition Language (IDL)™ , and OMG SysML™ are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer

Enterprise Architecture Guide for UAF

Abstract. This document describes a workflow for creating Enterprise Architecture (EA) views in accordance with the Unified Architecture Framework (UAF) Modeling Language (UAFML). This EA Guide for UAF is published as a non-normative component of the UAF specification. It is intended to be used in conjunction with the UAF Sample Problem that defines architecture views for a Search and Rescue Mission. The nine steps of the workflow are laid out in alignment with the stakeholder viewpoints in UAF for producing the requisite architecture views in each of those viewpoints. This underlying architecture description method is an implementation of the Architecture Elaboration process in ISO 42020 and can be used in conjunction with processes for the Conceptualization and Evaluation of an architecture specified in ISO 42020. It can also be used as the basis for an EA modeling methodology, architecture development planning, and modeling project organization and planning. The Guide covers architecting of the enterprise as well as architecting (at a high level) of major entities within the enterprise.

Table of Contents

Table of Figures

1 Introduction

Because of increasing complexity and rising costs, it is important to ensure that systems under development can properly interoperate with each other, be resilient against various threats, and meet the overarching capabilities that they were intended to achieve. UAF architecture models provide a means to develop an understanding of the complex relationships that exist between organizations, operations, systems, and services and enable the analysis of these things to ensure that they meet the expectations of the user community. UAF defines ways of representing an enterprise architecture that enables stakeholders to focus on specific areas of interest in the enterprise while retaining sight of the big picture.

This Enterprise Architecture (EA) Guide defines a workflow for creating EA views in accordance with the Unified Architecture Framework Modeling Language (UAFML). The Guide is published as a non-normative component of the UAF specification. It is intended to be used in conjunction with the UAF Sample Problem that defines architecture views for a Search and Rescue Mission. UAF Specification documents can be downloaded from the OMG webpage: www.omg.org/spec/UAF/About-UAF/.

The example approach defined in this Guide is just one way to approach architectures when using UAF and is intended to be informative only, not an official OMG-mandated way of doing this.

The nine steps of the workflow are laid out in alignment with the stakeholder viewpoints in the UAF for producing the requisite architecture views with respect to each of those viewpoints. These steps can be performed in any order and in practice are often done concurrently. The workflow connections between architecture views are intended to show how views can or should influence each other, not the order in which the views are created. It is not uncommon that many of the predecessor views already exist somewhere in one form or another and these can be either used as-is or enhanced as needed.

This architecture description method can be used in conjunction with processes for the conceptualization and evaluation of an architecture [ISO 42020 2019], and also used as the basis for an EA modeling methodology, architecture development planning, MBSE capability assessment, and modeling project organization. The Guide covers architecting of the enterprise as well as architecting (at a high level) of major entities within the enterprise. The Guide can also be used by Systems Engineers to model complex systems, organizations and services.

In the context of UAF, an enterprise is a "human undertaking or venture that has a mission, goals and objectives to offer products or services, or to achieve a desired project outcome or business outcome" [ISO 42010 2021]. An enterprise architecture is a set of "fundamental concepts and properties … and governing principles for the realization and evolution" of the enterprise [ibid.].

UAF views tie the architecture to the requirements in a more holistic manner. Ensuring the context of the requirements is properly defined will help improve acquisition speed and effectiveness. Requirements apply to resources of various kinds (e.g., capability configurations, systems, software, artifacts, technologies) but also apply to operational performers and activities, services and service functions, organizational positions and responsibilities, security processes and controls, etc.

The EA Guide is intended for those people who already have good knowledge of the following areas:

a) Model-based systems engineering (MBSE) principles and concepts
b) Systems Modeling Language (SysML) concepts and practices
c) Object-oriented analysis and design principles and concepts
d) Systems engineering and systems modeling practices
e) Architecture description concepts and practices
f) Enterprise architecting and modeling practices
g) UAF concepts and applications

1.1 Overview of the Unified Architecture Framework

The UAF specification consists of four main components as illustrated in Figure 1:1. View specifications are organized in a two-dimensional grid and these provide direction to tool vendors and to those who are creating the architecture views regarding what types of model elements are pertinent to those views. The Domain Metamodel

(DMM) establishes the underlying foundational modeling constructs to be used in modeling an enterprise and major entities within the enterprise. The UAF Modeling Language (UAFML) specifies how SysML modeling constructs can be used to create the views defined by the view specification. Finally, the Enterprise Guide provides a structured way to create the views defined in UAFML and is intended to be used in conjunction with the Sample Problem for a Search and Rescue Mission enterprise architecture.

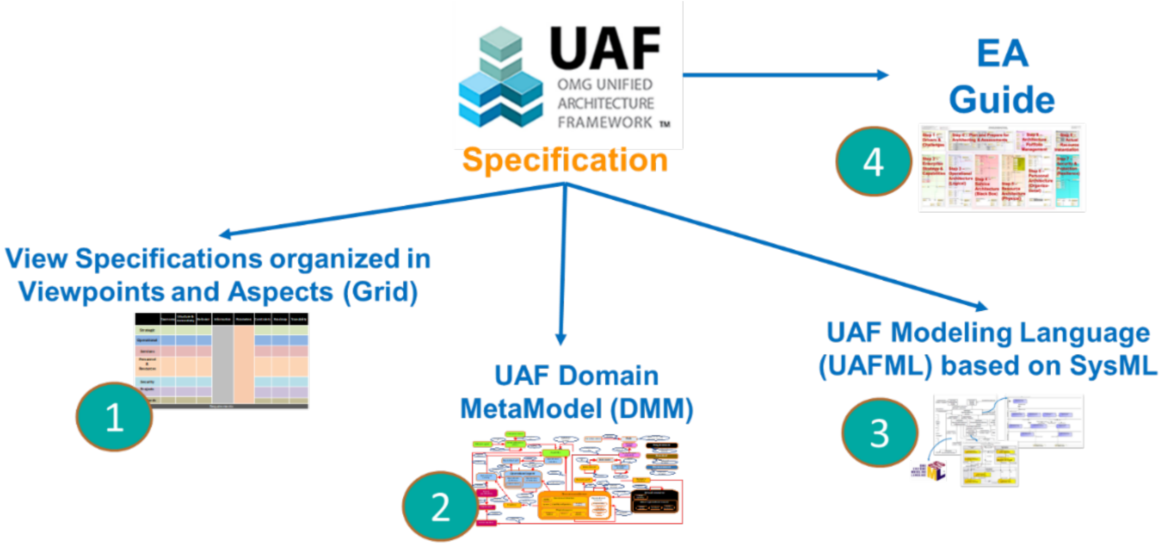
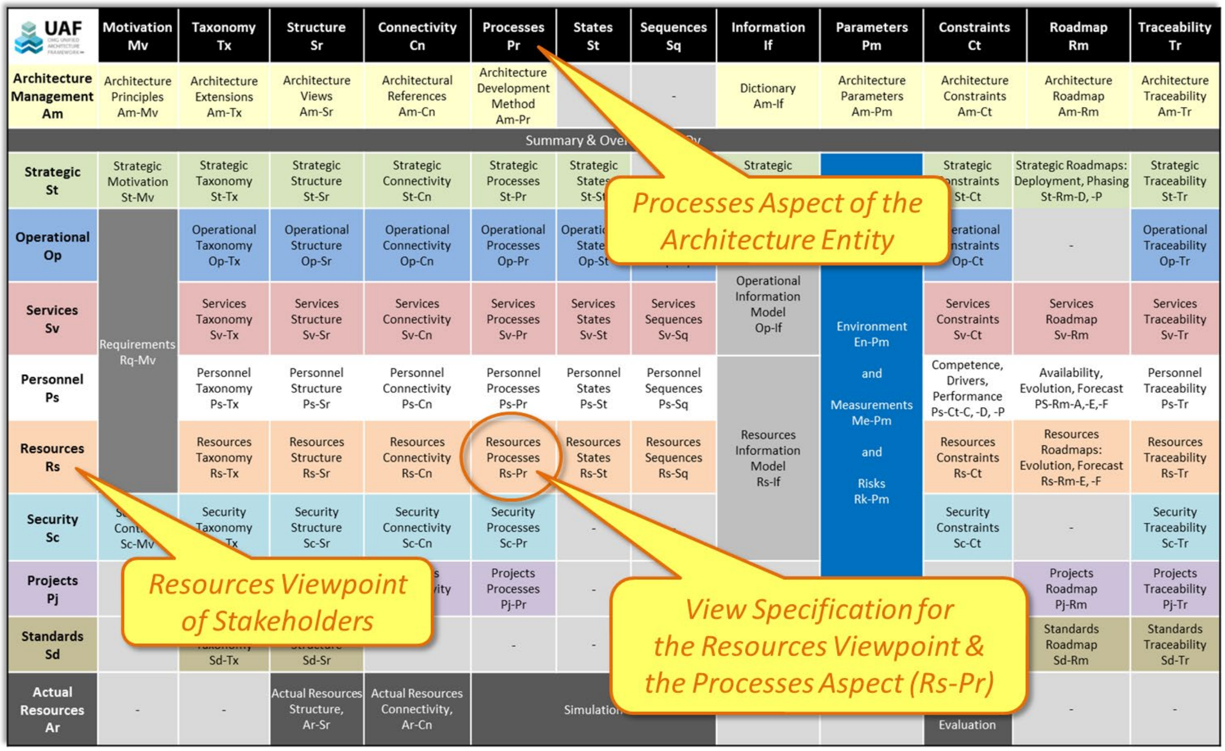

UAF
OMG UNIFIED ARCHITECTURE FRAMEWORK ™
Specification
EA Guide
4
View Specifications organized in Viewpoints and Aspects (Grid)
1
UAF Domain MetaModel (DMM)
2
UAF Modeling Language (UAFML) based on SysML
3

Figure 1:1 - UAF Specification Components

The planning of the EA effort should establish which architecture views and products will be developed and how they will be represented for this Enterprise to ensure coherence and completeness of all the EA views and products. The Planning and Preparation activity outlined in section 3.2 and detailed in Appendix A will help determine the best views to develop or use in a particular EA or other architecting effort.

An example of a non-EA effort is a "solution architecture" that specifies a real-world implementation which, if realized, could solve a specific problem. To illustrate this, suppose you have a problem space characterized as "the lack of an operational performer capable of visually locating floating victims regardless of ambient light." The problem space may include an airborne team (which may be part of a Search and Rescue mission) that lacks equipment capability for discerning the heat signature of a victim who is floating in the water. The solution space could include a sensor system incorporating infrared sensor technology. The UAF could be used to specify the architecture and design of the hardware, firmware and software comprising a "floating victim sensor" product, which in turn could be linked to the enterprise's operational performer that is enabled by the sensor.

The UAF Grid as a View Organizing Construct. The UAF Grid (Figure 1:2) has rows that represent typical stakeholder domains (or *viewpoints* as they are called in UAF) that can be used when modeling an enterprise architecture. The Grid has columns that represent the architecture *aspects* (in UAF these were formerly called 'model kinds') that correspond to "part of an entity's character or nature" [42010 2021]. This Grid is provided in the UAF standard as a structuring formalism for organizing the 82 view specifications defined within UAF.



UAF	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Architecture Management Am	Architecture Principles Am-Mv	Architecture Extensions Am-Tx	Architecture Views Am-Sr	Architectural References Am-Cn	Architecture Development Method Am-Pr		-	Dictionary Am-If	Architecture Parameters Am-Pm	Architecture Constraints Am-Ct	Architecture Roadmap Am-Rm	Architecture Traceability Am-Tr
					Summary & Ove		Ov					
Strategic St	Strategic Motivation St-Mv	Strategic Taxonomy St-Tx	Strategic Structure St-Sr	Strategic Connectivity St-Cn	Strategic Processes St-Pr	Strategic States St-St		Strategic	Environment En-Pm and Measurements Me-Pm and Risks Rk-Pm	Strategic Constraints St-Ct	Strategic Roadmaps: Deployment, Phasing St-Rm-D, -P	Strategic Traceability St-Tr
Operational Op	Requirements Rq-Mv	Operational Taxonomy Op-Tx	Operational Structure Op-Sr	Operational Connectivity Op-Cn	Operational Processes Op-Pr	Operational States Op-St		Operational Information Model Op-If		Operational Constraints Op-Ct	-	Operational Traceability Op-Tr
Services Sv		Services Taxonomy Sv-Tx	Services Structure Sv-Sr	Services Connectivity Sv-Cn	Services Processes Sv-Pr	Services States Sv-St	Services Sequences Sv-Sq			Services Constraints Sv-Ct	Services Roadmap Sv-Rm	Services Traceability Sv-Tr
Personnel Ps		Personnel Taxonomy Ps-Tx	Personnel Structure Ps-Sr	Personnel Connectivity Ps-Cn	Personnel Processes Ps-Pr	Personnel States Ps-St	Personnel Sequences Ps-Sq	Resources Information Model Rs-If		Competence, Drivers, Performance Ps-Ct-C, -D, -P	Availability, Evolution, Forecast PS-Rm-A,-E,-F	Personnel Traceability Ps-Tr
Resources Rs		Resources Taxonomy Rs-Tx	Resources Structure Rs-Sr	Resources Connectivity Rs-Cn	Resources Processes Rs-Pr	Resources States Rs-St	Resources Sequences Rs-Sq			Resources Constraints Rs-Ct	Resources Roadmaps: Evolution, Forecast Rs-Rm-E, -F	Resources Traceability Rs-Tr
Security Sc	Se Cont Sc-Mv	Security Taxonomy Tx	Security Structure Sc-Sr	Security Connectivity Sc-Cn	Security Processes Sc-Pr	-				Security Constraints Sc-Ct	-	Security Traceability Sc-Tr
Projects Pj				s ity	Projects Processes Pj-Pr	-					Projects Roadmap Pj-Rm	Projects Traceability Pj-Tr
Standards Sd		Sd-Tx	Sd-Sr		-	-					Standards Roadmap Sd-Rm	Standards Traceability Sd-Tr
Actual Resources Ar	-	-	Actual Resources Structure, Ar-Sr	Actual Resources Connectivity, Ar-Cn	Simulatio					Evaluation	-	-

Processes Aspect of the Architecture Entity

Resources Viewpoint of Stakeholders

View Specification for the Resources Viewpoint & the Processes Aspect (Rs-Pr)

Figure 1:2 - UAF Grid as a View Organizing Construct

Modeling Using UAF. The UAF Modeling Language[1] (UAFML) is an implementation of the DMM that specifies how the UAF views can be modeled using the SysML notation and semantics. Even though the UAFML is based on SysML, there are some significant differences that should be noted. SysML is great for doing the following activities: (a) modeling systems and for doing systems engineering, (b) defining and tracing between levels of abstraction within a system, (c) defining the logical and physical attributes for a system and the mapping of requirements and functions to these attributes. The UAF Modeling Language provides all this, plus more:

a) **Capability and Enterprise Concepts:** defines the "why" and "what" and "when" before the "how"
b) **Services Concepts:** definition of enterprise services (producing and consuming) and traceability to capabilities, operations and implementing resources
c) **Human Factors:** How people and systems interact, and their expected knowledge & skills
d) **Security:** Identifying risk, its mitigation, and integrating security into the architecture
e) **Standards:** definition of and compliance with standards in the architecture
f) **Project Deliveries:** phased milestone approach to capability deployment
g) **System Configuration Over Time:** deployment and changes in roadmaps and timelines
h) **Tie-in to Non-System Elements in the Architecture:** Easy way to link the entire Architecture to Requirements
i) **Built-in Traceability Between Multiple Views:** Between Layers and Across Layers

[1] In version 1.2 of the UAF specification, the UAF Profile (UAFP) was renamed as the UAF Modeling Language (UAFML) to better reflect its intended purpose. Where SysML is a general-purpose language for doing systems engineering, the UAFML is a general-purpose language for modeling an enterprise in support of Enterprise Systems Engineering (ESE) and Enterprise Architecture activities. Of course, UAFML can also be used to model systems, subsystems, major assemblies, products, software applications, but there is usually a transition point where SysML is used primarily at some lower level.

1.2 Relationship to Architecture Standard Processes

The architecture description approach defined in the EA Guide can be used in conjunction with the processes for conceptualization and evaluation of an architecture. The defined architecture description approach is consistent with the Architecture Elaboration process in the Architecture Processes standard [ISO 42020 2019] in the sense that the Elaboration process is where the architecture models and views are created that become part of the architecture description. The defined method in this Guide has the following potential applications:

- ✓ Workflow reference model in this Enterprise Architecture (EA) Guide,
- ✓ Reference model as the basis for an EA Modeling Methodology that defines associated methods, patterns, templates, tools, and techniques for each workflow step,
- ✓ Framework for project planning and architecture definition activities, and
- ✓ Training and certification on architecture frameworks and modeling approaches.

The focus of the EA Guide is on providing a method for implementing the Elaboration Process shown in Figure 1:3. The "elaboration" of the architecture is in the form of architecture models and views, such as those provided by an architecture description built using UAF.

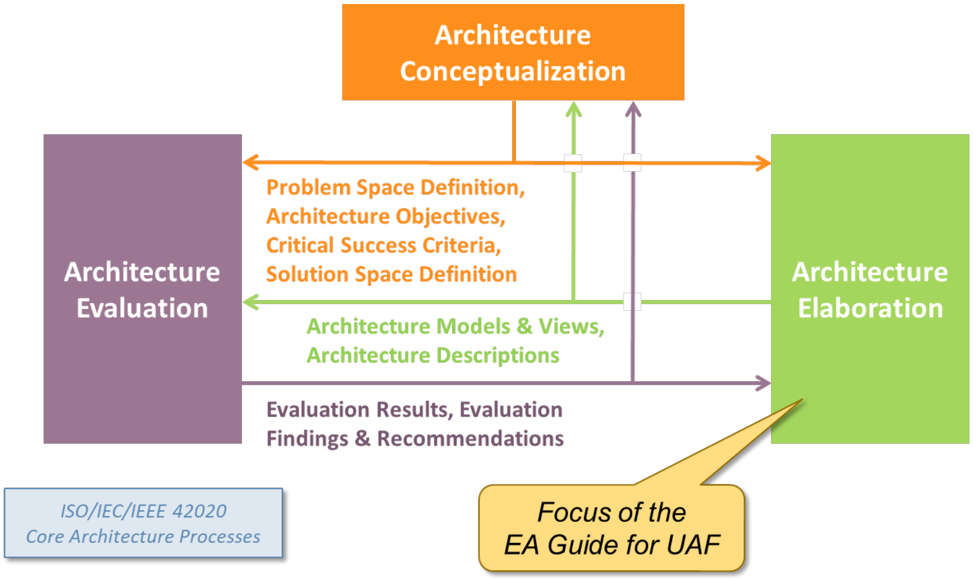

Architecture Conceptualization
Architecture Evaluation
Architecture Elaboration
Problem Space Definition, Architecture Objectives, Critical Success Criteria, Solution Space Definition
Architecture Models & Views, Architecture Descriptions
Evaluation Results, Evaluation Findings & Recommendations
ISO/IEC/IEEE 42020 Core Architecture Processes
Focus of the EA Guide for UAF

Figure 1:3 - Guide Focus on Architecture Elaboration Process in ISO 42020

The architecture models and views also provide useful information for the other architecture processes in the ISO 42020 standard as shown in Figure 1:4. An EA model is especially useful in support of architecture governance and management conducted at the enterprise level. The processes in ISO 42020 are defined as follows:

Architecture Governance	Establish and maintain alignment of architectures with enterprise goals and strategies and with related architectures
Architecture Management	Ensure the proper implementation of architecture governance directives and the timely and efficient achievement of architecture collection objectives
Architecture Conceptualization	Identify architectural solutions that address stakeholder concerns, achieve architecture objectives, and meet relevant requirements
Architecture Evaluation	Determine the extent to which architectures meet their objectives and address stakeholder concerns
Architecture Elaboration	Describe or document an architecture in a sufficiently complete and correct manner for the intended uses of the architecture
Architecture Enablement	Develop, maintain and improve the enabling capabilities, services and resources needed in performing the other architecture processes

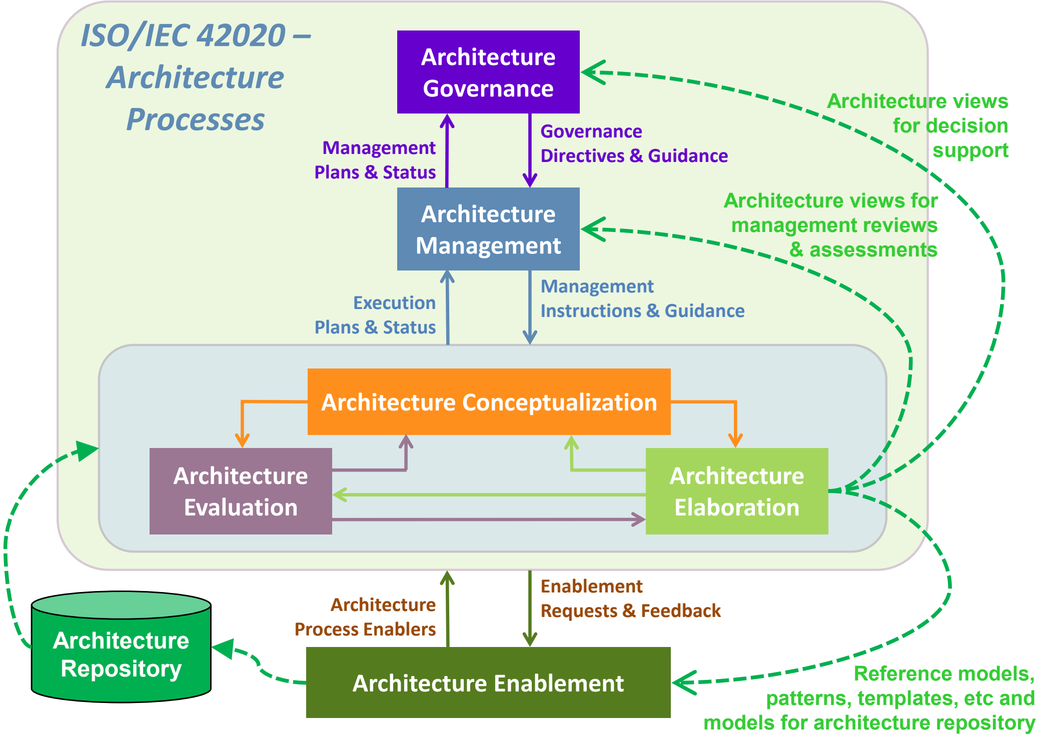

ISO/IEC 42020 – Architecture Processes
Architecture Governance
Management Plans & Status
Governance Directives & Guidance
Architecture views for decision support
Architecture Management
Architecture views for management reviews & assessments
Execution Plans & Status
Management Instructions & Guidance
Architecture Conceptualization
Architecture Evaluation
Architecture Elaboration
Architecture Process Enablers
Enablement Requests & Feedback
Architecture Repository
Architecture Enablement
Reference models, patterns, templates, etc and models for architecture repository

Figure 1:4 - Architecture Views and Models are Used in Other Architecture Processes

1.3 Purpose and Intended Applications for the EA Guide

The purpose of the EA Guide is to define what steps to take when creating UAF views based on the underlying models of an enterprise architecture. The Guide covers architecting of the enterprise as well as architecting (at a high level) of key entities within the enterprise, such as major items like missions, systems, services, organizations, programs, facilities, etc. There are a number of potential ways the guide can be used:

1) **Basis for creating a Unified Modeling Methodology**
(where Methodology = Process + Methods + Tools + Techniques + Templates…)

2) **Basis for building Architecture Views and Models**
   a) Agreement on division of responsibilities, apportionment, and allocation of work
      i) Between an upper enterprise and lower enterprise(s)
      ii) Between an acquisition office and its prime contractor

iii) Between a prime contractor and its suppliers

b) Development of training for architecture modeling classes and workshops

c) Problem framing workshop to identify appropriate models and views for an activity or effort

3) Basis for a Process Guide template instantiated in UAF plug-ins for MBSE tools

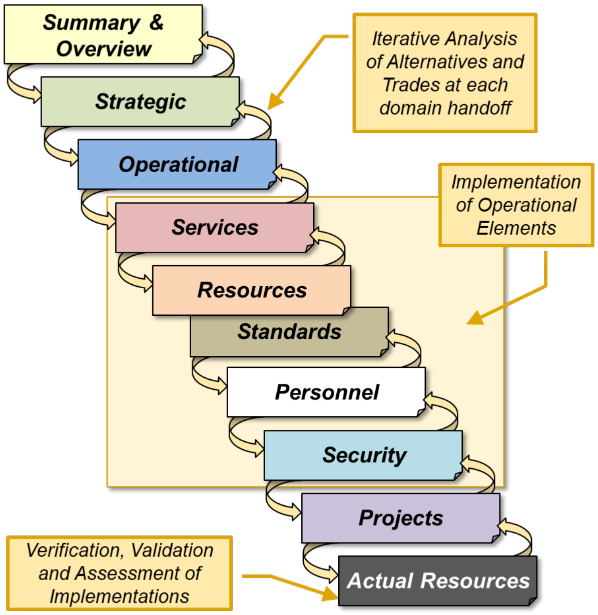
a) Navigation Panel, Dashboard, Landing Page (or similar item) within the model

b) Model Management Work Breakdown Structure used in resource planning

1.4 Layered Progression of Architecture Definition

UAF provides a complete set of stakeholder viewpoints as the basis for defining the variety of necessary architecture views of an Enterprise and these views are specified in the UAFML. The viewpoints allow for a logical and systematic flow of architecting activities:

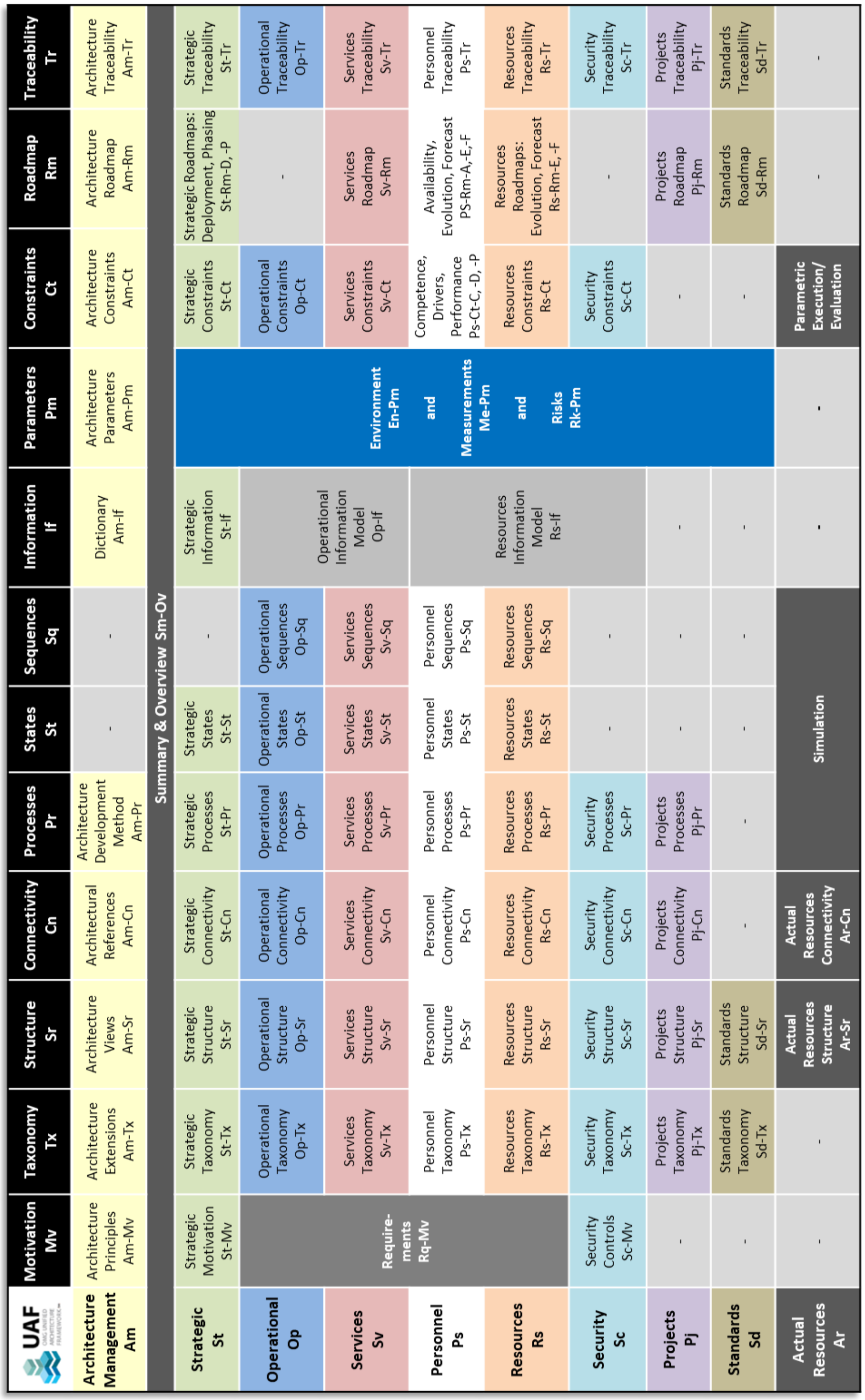
- ✓ Concerns and objectives drive a strategic plan that increases value to enterprise stakeholders
- ✓ The strategic plan deploys capabilities in phases to help address gaps and shortfalls
- ✓ Capabilities are actualized by operational roles, activities, and performers
- ✓ Operational concepts are implemented through services, resources, and personnel
- ✓ Resources, services, personnel, and operations are linked to standards
- ✓ Risk and threats are mitigated through necessary security and protection controls
- ✓ Requirements, constraints and concerns are understood and communicated to projects
- ✓ Plans deliver the resources according to project activities and milestones
- ✓ Resources are characterized and verified


Summary & Overview
Strategic
Operational
Services
Resources
Standards
Personnel
Security
Projects
Actual Resources
Iterative Analysis of Alternatives and Trades at each domain handoff
Implementation of Operational Elements
Verification, Validation and Assessment of Implementations

Even though these are presented here as a "waterfall," this is merely a logical flow rather than a strictly temporal flow. This workflow can be performed top-down, bottom-up, or middle-out (whatever is needed to meet the objectives), should be adapted and tailored to fit the situation, and will likely be performed iteratively. Tailoring of the workflow should be done in conjunction with tailoring of the modeling methods and tools in accordance with the Problem Framing step conducted in the Architecture Enablers Development workflow described in section A.4.2 and the tailoring of the modeling profile and environment described in section A.4.5.

An important consequence of performing these steps in parallel with different groups and people is the challenge to keep things in concordance and synchronized across the full set of architecture views. This will usually require establishment of architecture governance procedures and forums, perhaps with something like an architecture governance board to help orchestrate changes and serve as a decision body with authority for making architectural changes. The UAF architecture views can be used to inform the governance process as illustrated in Figure 1:4.

1.5 Two-Dimensional Grid of View Specifications in UAF



UAF	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Architecture Management Am	Architecture Principles Am-Mv	Architecture Extensions Am-Tx	Architecture Views Am-Sr	Architectural References Am-Cn	Architecture Development Method Am-Pr	-	-	Dictionary Am-If	Architecture Parameters Am-Pm	Architecture Constraints Am-Ct	Architecture Roadmap Am-Rm	Architecture Traceability Am-Tr
Summary & Overview Sm-Ov												
Strategic St	Strategic Motivation St-Mv	Strategic Taxonomy St-Tx	Strategic Structure St-Sr	Strategic Connectivity St-Cn	Strategic Processes St-Pr	Strategic States St-St	-	Strategic Information St-If	**Environment En-Pm** **and** **Measurements Me-Pm** **and** **Risks Rk-Pm**	Strategic Constraints St-Ct	Strategic Roadmaps: Deployment, Phasing St-Rm-D, -P	Strategic Traceability St-Tr
Operational Op	**Require-ments Rq-Mv**	Operational Taxonomy Op-Tx	Operational Structure Op-Sr	Operational Connectivity Op-Cn	Operational Processes Op-Pr	Operational States Op-St	Operational Sequences Op-Sq	Operational Information Model Op-If		Operational Constraints Op-Ct	-	Operational Traceability Op-Tr
Services Sv		Services Taxonomy Sv-Tx	Services Structure Sv-Sr	Services Connectivity Sv-Cn	Services Processes Sv-Pr	Services States Sv-St	Services Sequences Sv-Sq			Services Constraints Sv-Ct	Services Roadmap Sv-Rm	Services Traceability Sv-Tr
Personnel Ps		Personnel Taxonomy Ps-Tx	Personnel Structure Ps-Sr	Personnel Connectivity Ps-Cn	Personnel Processes Ps-Pr	Personnel States Ps-St	Personnel Sequences Ps-Sq	Resources Information Model Rs-If		Competence, Drivers, Performance Ps-Ct-C, -D, -P	Availability, Evolution, Forecast PS-Rm-A,-E,-F	Personnel Traceability Ps-Tr
Resources Rs		Resources Taxonomy Rs-Tx	Resources Structure Rs-Sr	Resources Connectivity Rs-Cn	Resources Processes Rs-Pr	Resources States Rs-St	Resources Sequences Rs-Sq			Resources Constraints Rs-Ct	Resources Roadmaps: Evolution, Forecast Rs-Rm-E, -F	Resources Traceability Rs-Tr
Security Sc	Security Controls Sc-Mv	Security Taxonomy Sc-Tx	Security Structure Sc-Sr	Security Connectivity Sc-Cn	Security Processes Sc-Pr	-	-			Security Constraints Sc-Ct	-	Security Traceability Sc-Tr
Projects Pj	-	Projects Taxonomy Pj-Tx	Projects Structure Pj-Sr	Projects Connectivity Pj-Cn	Projects Processes Pj-Pr	-	-	-		-	Projects Roadmap Pj-Rm	Projects Traceability Pj-Tr
Standards Sd	-	Standards Taxonomy Sd-Tx	Standards Structure Sd-Sr	-	-	-	-	-		-	Standards Roadmap Sd-Rm	Standards Traceability Sd-Tr
Actual Resources Ar	-	-	**Actual Resources Structure Ar-Sr**	**Actual Resources Connectivity Ar-Cn**	**Simulation**			-	-	**Parametric Execution/ Evaluation**	-	-

1.6 Cross-Cutting Viewpoints

Even though the UAF grid shows the viewpoints as being horizontal layers, the nature of the relationships between viewpoints is somewhat more complicated in reality. There are some cross-cutting viewpoints, such as the Services and Security viewpoints, that are associated with architecture elements in the Operational, Personnel and Resources viewpoints. The cross-cutting nature of these viewpoints is illustrated in Figure 1:5. Several of the viewpoints are cutting across all the other viewpoints and are illustrated here as vertical bars (even though these are shown horizontally in the grid), namely the Architecture Management, Summary and Overview, and Standards viewpoints.

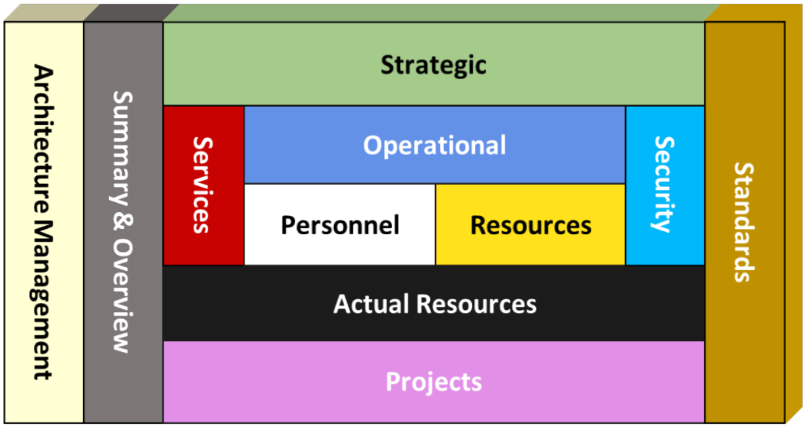


Figure 1:5 - Cross-Cutting Viewpoints in UAF

1.7 Modeling Tools

This Guide will not address how the various modeling tools can be used for UAF modeling. However, the view specifications defined in UAF can be implemented by tool vendors so that modelers can create architectures that are organized by these views where each viewpoint and aspect is clearly delineated in the model. Their implementation is usually in the form of a plug-in for their tool so the modeler can readily use the views provided by the UAFML.

2 Overview of the Guide

2.1 General Nature of the Workflow

The general workflow to implement these architecting activities is illustrated in Figure 2:1. Each step in the workflow conveys the architecture information to iteratively produce a definition of the problem space along with a definition of the solution space (i.e., implementation and instantiation). Tradeoffs are identified along the way and the results of architectural decisions are captured or articulated in the architecture views as they are fleshed out.

There will be some repetition back and forth between the steps to ensure a complete and coherent depiction of the architecture as it unfolds. It is not necessary that it must be implemented in a top-down fashion and can at times be counter-productive. The downward arrows represent going from higher levels of abstraction to more concrete things below and is not intended to show a sequential ordering of the work to be performed. The upward arrows represent iteration back to higher levels of abstraction to make adjustments and modifications, as necessary.

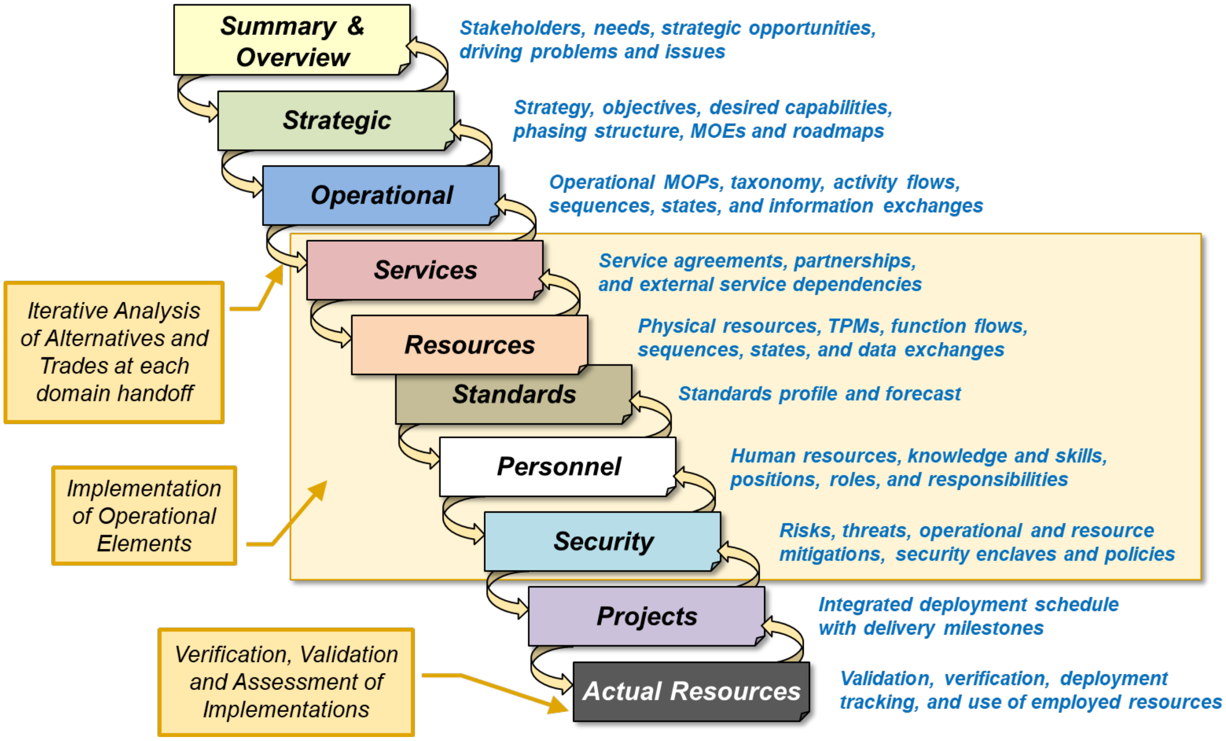

Summary & Overview
Stakeholders, needs, strategic opportunities, driving problems and issues
Strategic
Strategy, objectives, desired capabilities, phasing structure, MOEs and roadmaps
Operational
Operational MOPs, taxonomy, activity flows, sequences, states, and information exchanges
Services
Service agreements, partnerships, and external service dependencies
Resources
Physical resources, TPMs, function flows, sequences, states, and data exchanges
Standards
Standards profile and forecast
Personnel
Human resources, knowledge and skills, positions, roles, and responsibilities
Security
Risks, threats, operational and resource mitigations, security enclaves and policies
Projects
Integrated deployment schedule with delivery milestones
Actual Resources
Validation, verification, deployment tracking, and use of employed resources
Iterative Analysis of Alternatives and Trades at each domain handoff
Implementation of Operational Elements
Verification, Validation and Assessment of Implementations

Figure 2:1 - General Nature of the Workflow

The various steps, as mentioned before, are not intended to be the order in which the work is to be done. In practice, many of these are being done in parallel. However, they do have a logical relationship to each other in terms of how the information generated in the form of architecture views can "influence" the information contained in views generated in other steps.

The relationships between steps are illustrated in the interactions matrix shown in Table 1. In this N2 (n-squared) diagram, the outbound influencing relationships are shown in the upper right portion of the matrix (above the diagonal) while the inbound ones are shown in the lower left portion (below the diagonal).

Table 1: Information Relationships Between Major Steps

1. Architecture Drivers & Challenges	1	>							
2. Enterprise Strategy & Capabilities		2	>					>	
3. Operational Architectures	^	^	3	>	>	>	>		
4. Service Architectures		^	^	4	>				>
5. Resource Architectures			^		5	>			>
6. Personnel Architectures			^		^	6	>	>	>
7. Security Architectures			^		^		7	>	>
8. Projects Portfolio Management		^			^			8	>
9. Resource Realization									9

2.2 Architecture Description Workflow Steps

The workflow defines "what" to do when creating the UAF views but does not identify or define methods (the "how") or tools needed for each step (since this is methodology dependent). The nine steps illustrated in Figure 2:2 follow the basic flow illustrated in Figure 2:1 and these steps are decomposed to the third level to get to the point where individual UAF views are generated for each of the sub-steps.

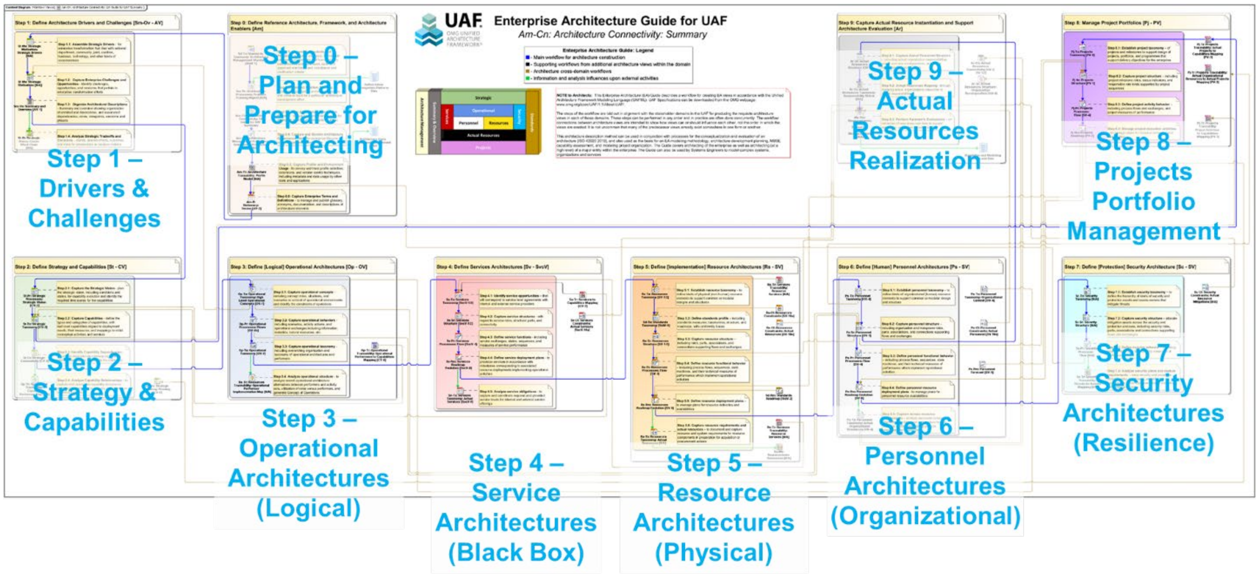

UAF
Enterprise Architecture Guide for UAF
Am-Cn: Architecture Connectivity: Summary
Step 0 – Plan and Prepare for Architecting
Step 1 – Drivers & Challenges
Step 2 – Strategy & Capabilities
Step 3 – Operational Architectures (Logical)
Step 4 – Service Architectures (Black Box)
Step 5 – Resource Architectures (Physical)
Step 6 – Personnel Architectures (Organizational)
Step 7 – Security Architectures (Resilience)
Step 8 – Projects Portfolio Management
Step 9 – Actual Resources Realization

Figure 2:2 - Workflow Major Steps

The UAFML provides view specifications for the UAF views in this workflow. This Guide is intended to be used in conjunction with the UAF Sample Problem document which provides example UAF views for an enterprise responsible for conducting Search and Rescue Mission activities.

These steps are not necessarily done in this order and are often done simultaneously with much iteration both internally within the viewpoint level and collaboratively between the stakeholders in the viewpoint levels/steps above and below. In any architecting effort, only some portions of the steps are carried out. The Problem Framing step outlined in section 2.4 and detailed in section A.4.2 can help identify the most pertinent and useful architecture views to be collected or created. Problem Framing helps you focus on the users and uses of each view along with the questions to be answered by each view. This approach helps avoid unnecessary work and can minimize the amount of churn and rework that is often encountered.

2.3 Multi-tier Application of the Workflow

It is common that the workflow, in part or in whole, is repeated at multiple tiers of an enterprise. For example, the top level of a company or government organization may develop an enterprise architecture for the entire organization but at a high level focused mainly on areas of strategic concern. So, in this case the enterprise architecture (at this level or "tier") might only deal with Steps 1 and 2, along with some of Step 8 for Portfolio Management of their enterprise assets. Step 0 for planning and preparing for architecture efforts would be performed to the extent necessary for their scope of architecting work.

A business unit or lower-level government organization would "fit within" the larger enterprise and might take the results of Steps 1 and 2 from the parent organization as an input to their enterprise architecture, hence only dealing with Steps 3 through 5, albeit perhaps only defining the Resource Architecture in Step 5 at a high level. Then a program within that mid-level organization could have its own enterprise architecture where they add some details to the Operational and Service Architecture (steps 3 and 4) but spend most of their efforts on fleshing out the Resource Architecture and adding the Personnel Architecture within their 3rd tier enterprise.

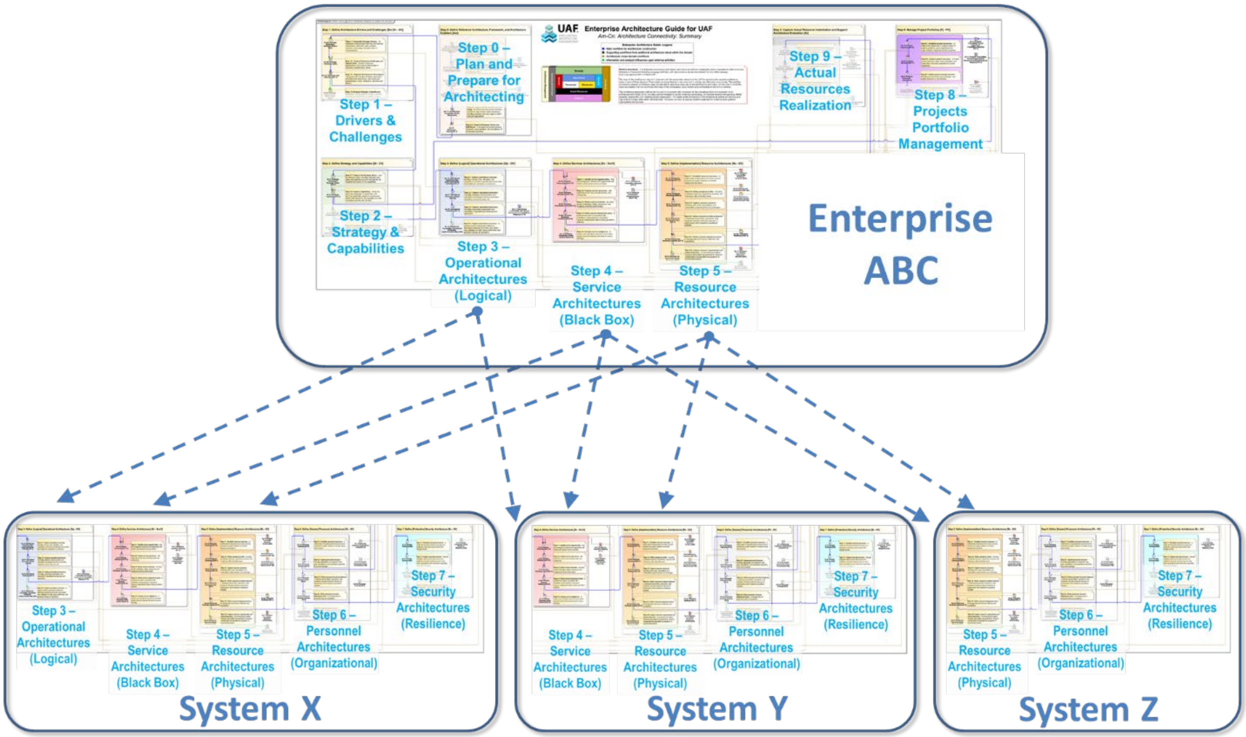

Step 0 – Plan and Prepare for Architecting
Step 1 – Drivers & Challenges
Step 9 – Actual Resources Realization
Step 8 – Projects Portfolio Management
Step 2 – Strategy & Capabilities
Step 3 – Operational Architectures (Logical)
Step 4 – Service Architectures (Black Box)
Step 5 – Resource Architectures (Physical)
Enterprise ABC
Step 3 – Operational Architectures (Logical)
Step 4 – Service Architectures (Black Box)
Step 5 – Resource Architectures (Physical)
Step 6 – Personnel Architectures (Organizational)
Step 7 – Security Architectures (Resilience)
System X
Step 4 – Service Architectures (Black Box)
Step 5 – Resource Architectures (Physical)
Step 6 – Personnel Architectures (Organizational)
Step 7 – Security Architectures (Resilience)
System Y
Step 5 – Resource Architectures (Physical)
Step 6 – Personnel Architectures (Organizational)
Step 7 – Security Architectures (Resilience)
System Z

Figure 2:3 - Multi-tier Application of Workflow Steps

A two-tier structure is illustrated in Figure 2:3 where the Enterprise defines the high-level aspect of the overall enterprise while the lower-level Systems projects will only add detail as appropriate to the purpose and scope of their project. The lower-level Systems projects may either simply view its higher-level representation in the enterprise project and then duplicate its elements and add details, or it may directly use its elements from the enterprise project and add more details.

These tiers could be internal to a single organization, or they could be separated into different organizations. Sometimes the enterprise tier is for a government agency and the systems tier is for a contractor hired by the government, with similar situations for prime contractor and subcontractor, corporate headquarters and a business unit, national and regional levels, state, and city levels, etc. Furthermore, there may be occasions where lower-level tiers will need to repeat step 1 and 2 to address their own drivers, challenges, strategies, and capabilities at their level.

The simple example above does not address the various interdependencies that will occur between enterprises and the interdependencies between the systems with an enterprise and between systems in related enterprises.

This tiering structure has implications for Architecture Governance as discussed in ISO 42020 and illustrated in Figure 1:4. This can be a complex subject and we cannot discuss the details in this Guide. Suffice it to say that this needs to be addressed in an architecture modeling methodology document and in organizational governance processes and procedures.

There is some point at which EA stops and regular program level Systems Engineering (SE) takes over. The Enterprise Architecture will necessarily overlap with the various System Architectures involved. This also should be documented in an organizational architecture modeling methodology and perhaps also in a SE management plan. This should outline when SE models take over and how information is exchanged between tiers and possibly between UAF and SysML models, as well as with other SE tools.

2.4 Problem Framing

The ordering shown in the workflow diagram is conceptual with respect to levels of abstraction and is not intended to show a sequential order of execution. These steps can be performed in any order and often many of them are done in parallel. Steps will be skipped or modified. It is not uncommon that architectural information from prior steps in the workflow will be provided to your project by another organization, typically a higher-level or external organization. Likewise, your project will often deliver architectural information to lower-level or internal organizations to use as the basis for their work. Each architecture effort will need to tailor the workflow according to the needs of the architecture development project.

Problem framing can be conducted to help identify the appropriate architecture models and views to build. Intended uses and users of the architecture are captured to determine issues to be explored, questions to be answered, the types of analysis that need to be performed using architecture models and views, and what are the interests and perspectives of the intended audience and expected users of the architecture description.

Tailoring of the workflow should be done in conjunction with tailoring of the modeling methods and tools in accordance with the Problem Framing step in section A.4.2 and the tailoring of the modeling profile and environment described in section A.4.5. More details on problem framing are provided in the paper "Problem Framing: Identifying the Right Models for the Job" from the 2019 INCOSE Symposium.

Once an Enterprise has invested in building UAF views and products at the strategic or operational level, these should be maintained and reused. Problem Framing for any time after the first iteration should use existing authoritative source products or should update them as appropriate. This reduces systems engineering staff burden and improves effectiveness and efficiency.

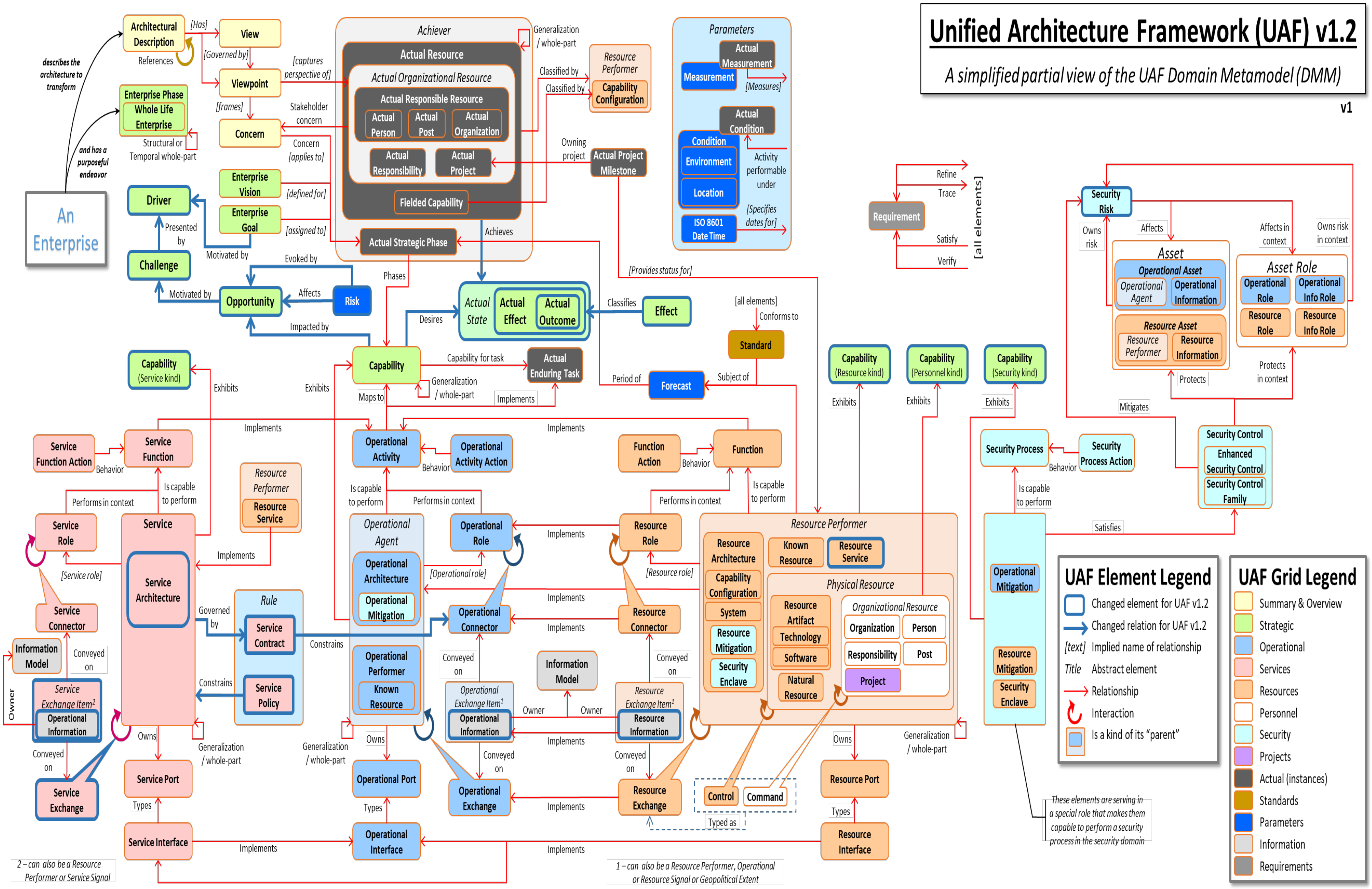
2.5 Conceptual Schema

The concepts described in this Guide are based on the UAF Domain Metamodel. A top-level conceptual schema is illustrated below which is used as the basis for conceptual schema diagrams portrayed for each of the workflow steps. Terms and definitions are provided throughout the document in the sections where the terms are used and collected together in a Glossary in Appendix B while Acronyms are in Appendix C.

Unified Architecture Framework (UAF) v1.2

A simplified partial view of the UAF Domain Metamodel (DMM)

v1



2.6 Symbology and Color Scheme

Conceptual Schema Symbology. The conceptual schema uses the symbology and color scheme as shown in Figure 2:4. The element colors align with the colors in the rows of the UAF Grid. New or changed elements and relationships in UAF version 1.2 are shown with heavy blue lines. Abstract elements are named using *italics* and using a lighter shade of the corresponding color for that viewpoint.

The notation used is a modified entity-relationship form of diagram where the boxes embedded within another box represent parent-child relationships of generalization-specialization. For example, an *Operational Agent* (abstract) has subtypes (children) Operational Architecture and Operational Performer, while the Operational Performer has a subtype Operational Mitigation (which is a special kind of element from the Security Viewpoint that performs a Security Process and satisfies a Security Control that mitigates a Security Risk).

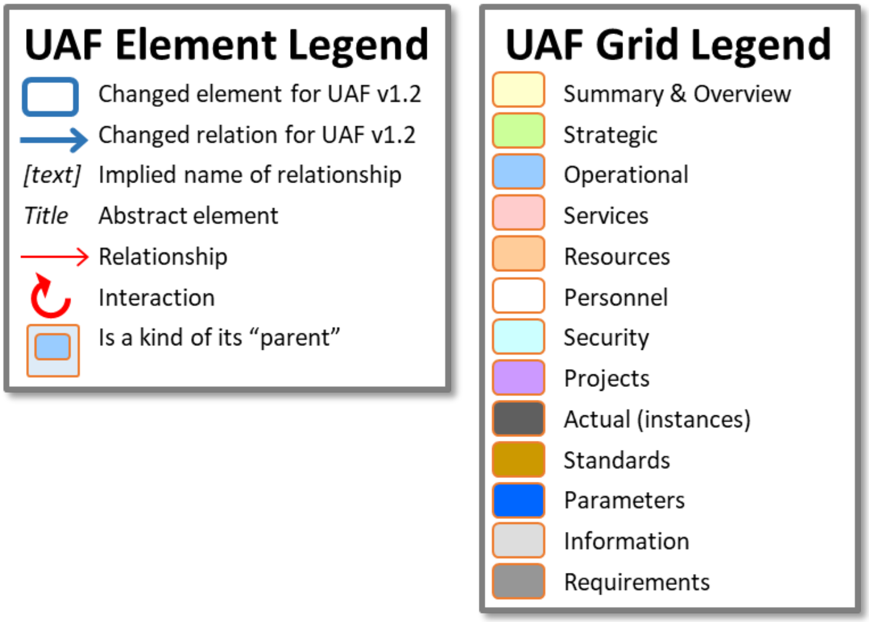


Figure 2:4 - Conceptual Schema Legend with Color Scheme and Symbology Used

Workflow Symbology. The flowlines in the diagrams have different meaning as shown in the legend in Figure 2:5. The main flow of work will follow the blue lines while support workflows within a particular UAF viewpoint are shown with black lines. Flowlines that cross between viewpoints are shown in brown while external flows are shown in green. An example of a workflow diagram showing typical view symbols and flowlines is shown in Figure 2:6

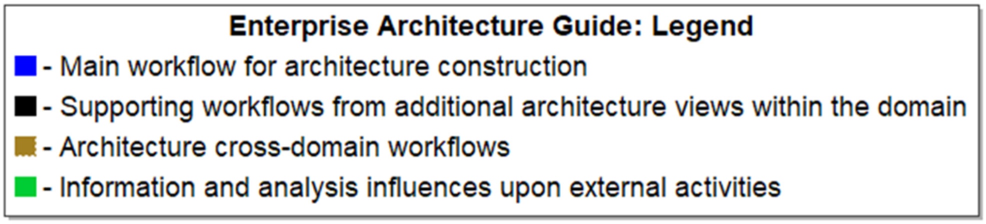
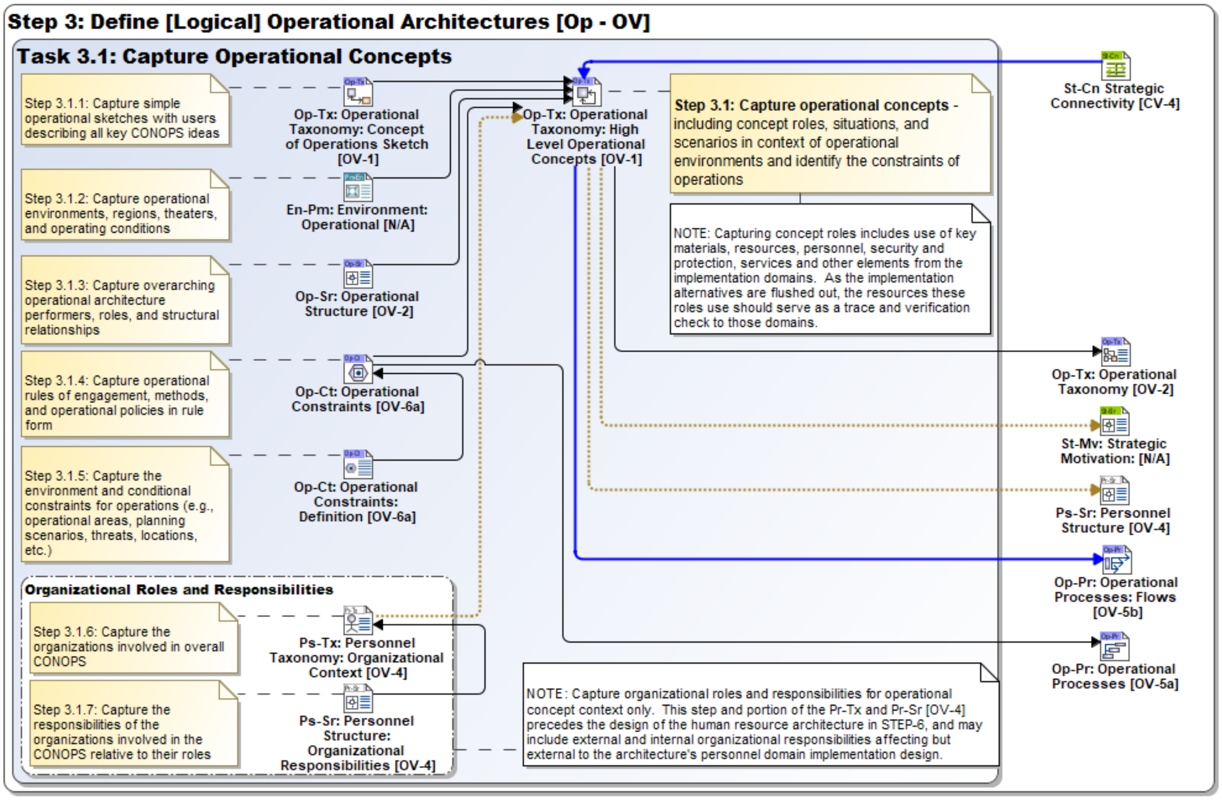

Enterprise Architecture Guide: Legend
- Main workflow for architecture construction
- Supporting workflows from additional architecture views within the domain
- Architecture cross-domain workflows
- Information and analysis influences upon external activities

Figure 2:5 - Workflow Lines Color Scheme Legend


Step 3: Define [Logical] Operational Architectures [Op - OV]
Task 3.1: Capture Operational Concepts
Op-Tx: Operational Taxonomy: Concept of Operations Sketch [OV-1]
En-Pm: Environment: Operational [N/A]
Op-Sr: Operational Structure [OV-2]
Op-Ct: Operational Constraints [OV-6a]
Op-Ct: Operational Constraints: Definition [OV-6a]
Op-Tx: Operational Taxonomy: High Level Operational Concepts [OV-1]
St-Cn Strategic Connectivity [CV-4]
Op-Tx: Operational Taxonomy [OV-2]
St-Mv: Strategic Motivation: [N/A]
Ps-Sr: Personnel Structure [OV-4]
Op-Pr: Operational Processes: Flows [OV-5b]
Op-Pr: Operational Processes [OV-5a]
Organizational Roles and Responsibilities
Ps-Tx: Personnel Taxonomy: Organizational Context [OV-4]
Ps-Sr: Personnel Structure: Organizational Responsibilities [OV-4]

Step 3.1.1: Capture simple operational sketches with users describing all key CONOPS ideas

Step 3.1.2: Capture operational environments, regions, theaters, and operating conditions

Step 3.1.3: Capture overarching operational architecture performers, roles, and structural relationships

Step 3.1.4: Capture operational rules of engagement, methods, and operational policies in rule form

Step 3.1.5: Capture the environment and conditional constraints for operations (e.g., operational areas, planning scenarios, threats, locations, etc.)

Step 3.1.6: Capture the organizations involved in overall CONOPS

Step 3.1.7: Capture the responsibilities of the organizations involved in the CONOPS relative to their roles

Step 3.1: Capture operational concepts - including concept roles, situations, and scenarios in context of operational environments and identify the constraints of operations

NOTE: Capturing concept roles includes use of key materials, resources, personnel, security and protection, services and other elements from the implementation domains. As the implementation alternatives are flushed out, the resources these roles use should serve as a trace and verification check to those domains.

NOTE: Capture organizational roles and responsibilities for operational concept context only. This step and portion of the Pr-Tx and Pr-Sr [OV-4] precedes the design of the human resource architecture in STEP-6, and may include external and internal organizational responsibilities affecting but external to the architecture's personnel domain implementation design.

Figure 2:6 - Example of Workflow Diagram Showing Typical View Symbols and Flowlines

Architecture View Designators. The architecture views produced by each step are identified by the UAF grid designator Aa-Bb where Aa represents the Viewpoint row in the grid and Bb represents the Aspect column in the grid. DoDAF view designations are also shown in [brackets]. The name of the architecture view matches the name of the view specification in the UAFML. In some cases, a subtitle is provided to identify what kind of information is provided in an instance of that UAF view. There will be some architecture views that are not part of the UAF specification since you will sometimes need fit-for-purpose (i.e., custom) views that capture the necessary architectural information.

This page intentionally left blank.

3 Architecture Planning and Preparation

3.1 Architecture Management Concepts

Conceptual Schema. The key concepts used in the Architecture Management Viewpoint that can be used as model elements in the architecture views and the relationships between these concepts are illustrated in the conceptual schema shown in Figure 3:1. These *key concepts* are highlighted in italics within the Narrative and some of the less obvious concepts are listed with the associated ISO-42010[2] *meaning* or the UAF ***meaning*** of that concept. Detailed definition of the entities and relationships shown in the conceptual schema are provided in the UAFML specification document.

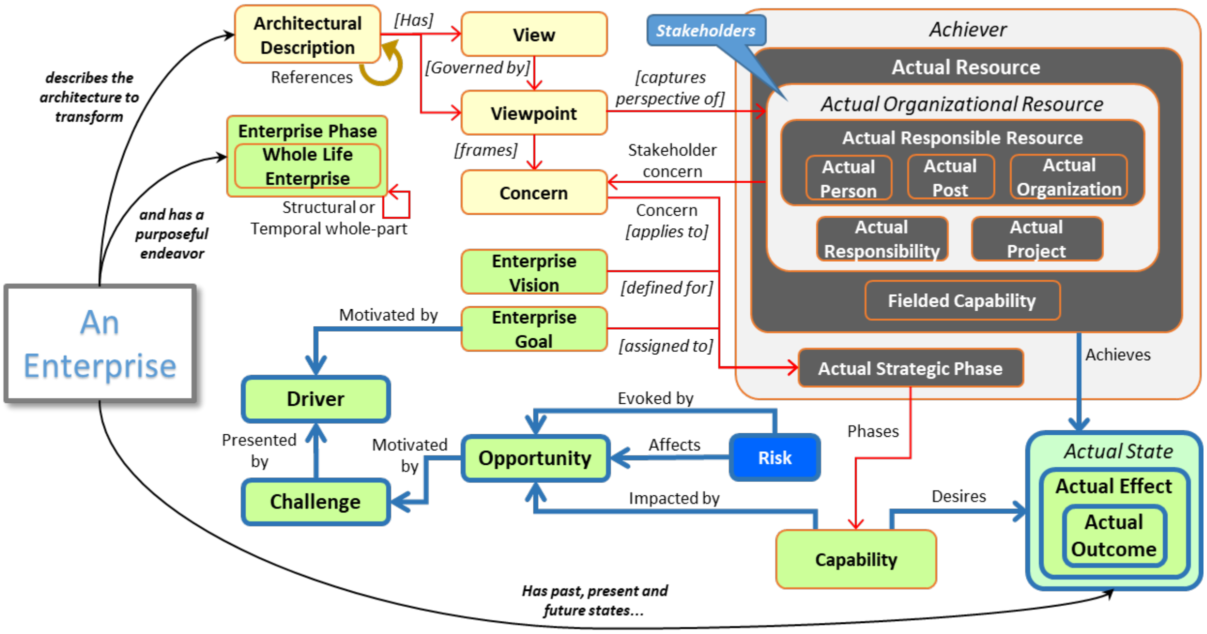


Figure 3:1 - Conceptual Schema for Architecture Management

An enterprise is a purposeful endeavor with an established ***enterprise vision*** to achieve its stated ***enterprise goals***. The enterprise will encounter strategic ***drivers*** that present ***challenges*** to the organizations that participate in the enterprise, which in turn will motivate the enterprise to pursue ***opportunities*** that address these challenges. The ***capabilities*** of the enterprise (or capabilities deployed to others for their own use) will be impacted by the opportunities to be pursued. The current or future capabilities will help achieve a series of ***effects*** that in the end will achieve some desired ***outcomes***. The enterprise will typically structure its transformation efforts into ***strategic phases*** that will endeavor to exhibit the desired ***capabilities***. Also, notice that the enterprise has had previous states and has current states that will affect how it currently responds to changes and to internal and external perturbations. This "history" of states must be considered when planning to transform the enterprise.

The enterprise can use an architecture framework (such as UAF) as the basis for developing a set of ***architectural descriptions*** to help transform the enterprise by setting new or modified ***enterprise goals***.

[2] ***ISO/IEC/IEEE 42010:2021 Software, systems and enterprise – Architecture description*** establishes a standard approach to describing an architecture using views and viewpoints, architecture description languages and architecture descriptions frameworks.

3.2 Introduction to Step 0

Purpose. The purpose of this step is to provide information pertinent to the entire architecture and to acquire or develop key enablers to facilitate development and maintenance of the architecture models and views. It presents supporting information rather than the architecture models themselves. Key stakeholders for this step are enterprise architects, stakeholders who want to discover the architecture, and technical managers. Their concerns are mainly about metadata relevant to the entire architecture or the architecting effort.

Workflow. The detailed workflow steps for Architecture Management are provided in Appendix A. Architecture Management activities are captured in Step 0 in this Guide since they usually precede the large number of activities involved in describing an Enterprise Architecture. Step 0 could be accomplished once to plan and prepare for multiple EA workstreams and projects. Some elements of Step 0 can be revisited as the need for changes in how the work will be performed or if the purpose and scope of an EA effort changes. Some parts of Step 0 are performed continuously, if necessary, to maintain the overarching items that enable the overall collection of EA and other architecture projects, such as modeling templates and patterns, architecture glossary and dictionary, architecture plans and guidebooks.

Architecture Description Standard Practices. Architecture Description as specified in the ISO 42010 standard provides essential principles and concepts to be used when capturing the architecture in models and views. The key principles and concepts of Architecture Description are outlined in Appendix A.2.

3.3 Establishing the Purpose and Scope of the Architecting Effort

Step 0 – Define Reference Architecture, Framework and Architecture Enablers – The main entry criterion for this Step is reaching a decision regarding the purpose of the architecture description, partly based on the concerns of the primary *stakeholders* who have an interest in the *architecture*. These *stakeholders* are often business or government leaders who have an interest in transformation of the enterprise to become more effective and efficient in achieving its goals and objectives.

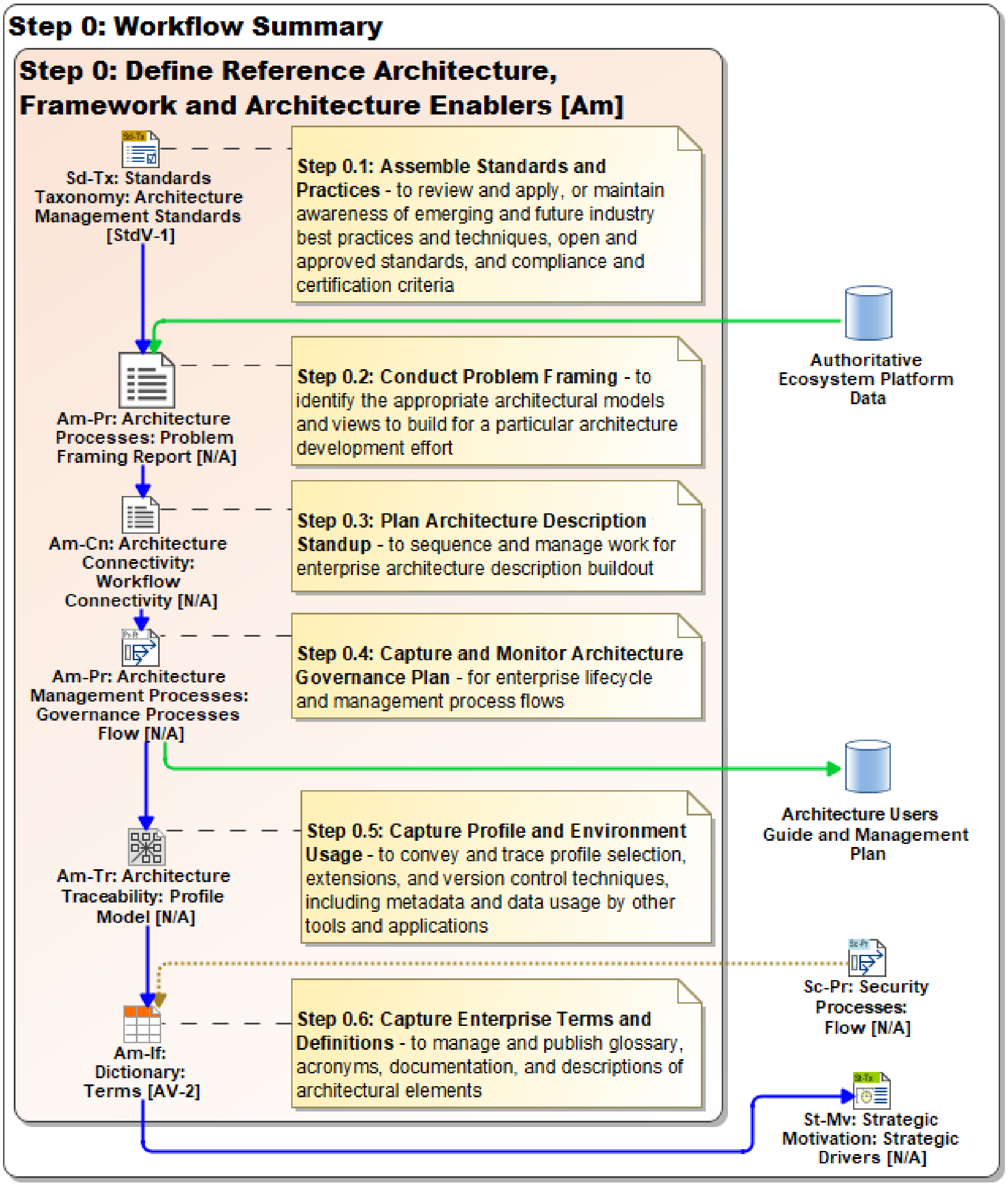
- *Architecture* – fundamental concepts or properties related to an entity in its environment and governing principles for the realization and evolution of this entity and its related life cycle processes
- *Architecture Description* – work product used to express an architecture
- *Stakeholder* – role, position, individual, organization or classes thereof, having an interest, right, share, or claim, in an entity or its architecture
- *Enterprise* – human undertaking or venture that has a mission, goals, and objectives to offer products or services or to achieve a desired project or business outcome

The enterprise architecture description is used by *stakeholders* to improve communication and cooperation among affected parties and enable them to work together in a more integrated, coherent fashion. This will, in turn, help the enterprise more effectively achieve its goals. This can be facilitated by creating a “reference architecture” that guides development of the rest of the enterprise architecture in Steps 3-7, as well as using an architecture framework that defines the views to be used.

This organizational framework can be tailored from the UAF by choosing the relevant views, modifying them where appropriate, and defining new views that are needed to express the key concepts and properties of the enterprise architecture. In addition, modeling templates, patterns and methods will be needed to help conduct and manage the architecture development efforts; these items are called “architecture enablers” in this Guide and are associated with the Architecture Enablement process in ISO 42020.

3.4 Workflow Summary

A summary level view of the six steps involved in Step 0 workflow is shown below. The flowlines represent how one architecture view “influences” another architecture view. These lines do not represent a particular sequence of process activity execution or imply information exchanges. These should be thought of as influence diagrams rather than process diagrams. These architecture views (e.g., views, diagrams, tables) can be mapped to the architecture processes used in your organization and can be incorporated into an architecture modeling methodology. Details for this workflow can be found in Appendix A.4.


Step 0: Workflow Summary
Step 0: Define Reference Architecture, Framework and Architecture Enablers [Am]
Sd-Tx: Standards Taxonomy: Architecture Management Standards [StdV-1]
Am-Pr: Architecture Processes: Problem Framing Report [N/A]
Am-Cn: Architecture Connectivity: Workflow Connectivity [N/A]
Am-Pr: Architecture Management Processes: Governance Processes Flow [N/A]
Am-Tr: Architecture Traceability: Profile Model [N/A]
Am-If: Dictionary: Terms [AV-2]
Authoritative Ecosystem Platform Data
Architecture Users Guide and Management Plan
Sc-Pr: Security Processes: Flow [N/A]
St-Mv: Strategic Motivation: Strategic Drivers [N/A]

Step 0.1: Assemble Standards and Practices - to review and apply, or maintain awareness of emerging and future industry best practices and techniques, open and approved standards, and compliance and certification criteria

Step 0.2: Conduct Problem Framing - to identify the appropriate architectural models and views to build for a particular architecture development effort

Step 0.3: Plan Architecture Description Standup - to sequence and manage work for enterprise architecture description buildout

Step 0.4: Capture and Monitor Architecture Governance Plan - for enterprise lifecycle and management process flows

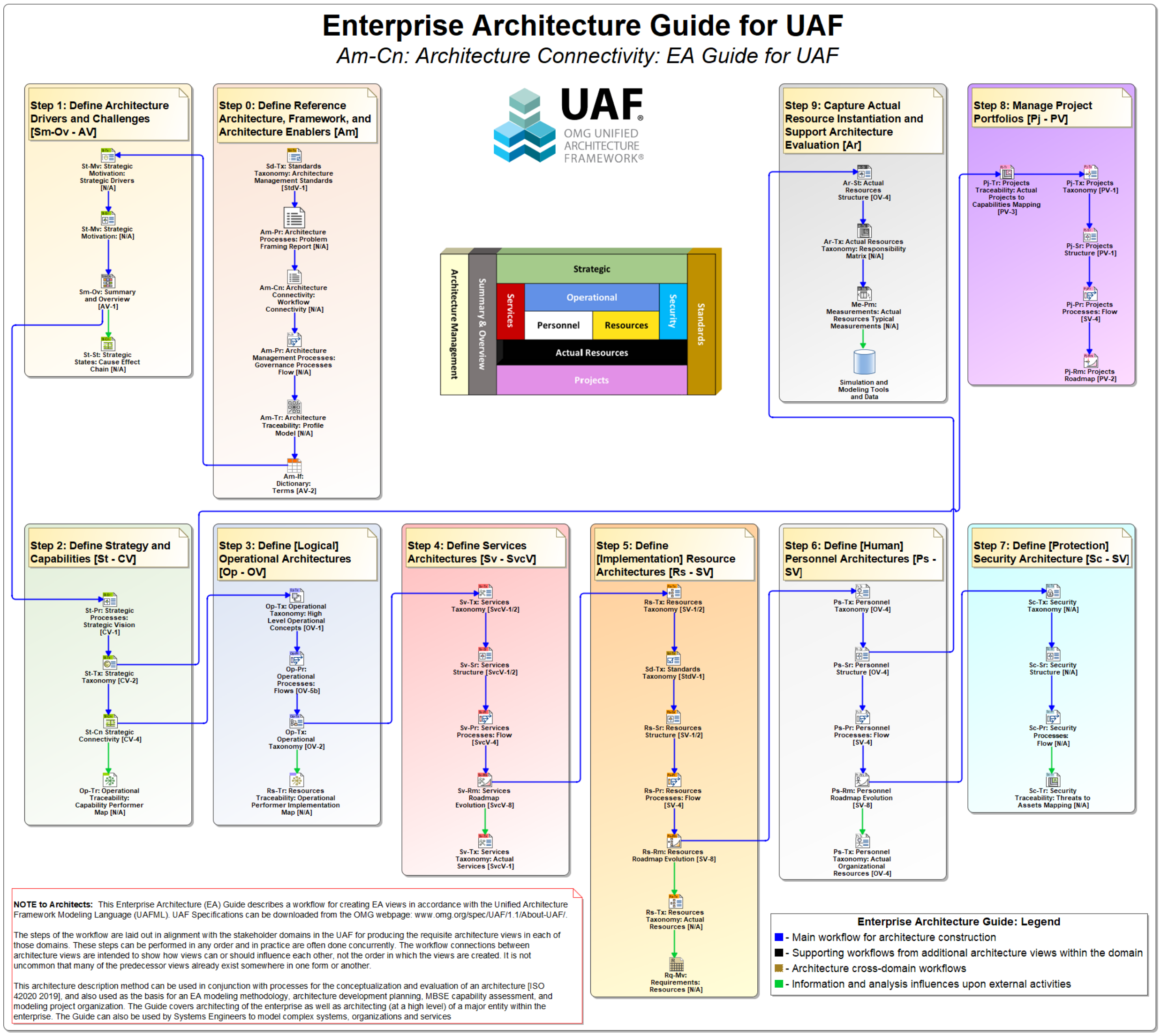
Step 0.5: Capture Profile and Environment Usage - to convey and trace profile selection, extensions, and version control techniques, including metadata and data usage by other tools and applications

Step 0.6: Capture Enterprise Terms and Definitions - to manage and publish glossary, acronyms, documentation, and descriptions of architectural elements

Figure 3:2 - Step 0: Define Reference Architecture, Framework and Architecture Enablers

4 Workflow Steps

The key architecture views in each major step are shown below. The blue line between these views represents the main workflow for architecture view construction. However, this does not necessarily represent the order in which these views are created. This is a logical ordering in the sense that one view logically follows the other one. In practice, many of these will be developed simultaneously.


Enterprise Architecture Guide for UAF
Am-Cn: Architecture Connectivity: EA Guide for UAF
UAF
OMG UNIFIED ARCHITECTURE FRAMEWORK®
Step 1: Define Architecture Drivers and Challenges [Sm-Ov - AV]
St-Mv: Strategic Motivation: Strategic Drivers [N/A]
St-Mv: Strategic Motivation: [N/A]
Sm-Ov: Summary and Overview [AV-1]
St-St: Strategic States: Cause Effect Chain [N/A]
Step 0: Define Reference Architecture, Framework, and Architecture Enablers [Am]
Sd-Tx: Standards Taxonomy: Architecture Management Standards [StdV-1]
Am-Pr: Architecture Processes: Problem Framing Report [N/A]
Am-Cn: Architecture Connectivity: Workflow Connectivity [N/A]
Am-Pr: Architecture Management Processes: Governance Processes Flow [N/A]
Am-Tr: Architecture Traceability: Profile Model [N/A]
Am-If: Dictionary: Terms [AV-2]
Architecture Management
Summary & Overview
Strategic
Services
Operational
Security
Standards
Personnel
Resources
Actual Resources
Projects
Step 9: Capture Actual Resource Instantiation and Support Architecture Evaluation [Ar]
Ar-St: Actual Resources Structure [OV-4]
Ar-Tx: Actual Resources Taxonomy: Responsibility Matrix [N/A]
Me-Pm: Measurements: Actual Resources Typical Measurements [N/A]
Simulation and Modeling Tools and Data
Step 8: Manage Project Portfolios [Pj - PV]
Pj-Tr: Projects Traceability: Actual Projects to Capabilities Mapping [PV-3]
Pj-Tx: Projects Taxonomy [PV-1]
Pj-Sr: Projects Structure [PV-1]
Pj-Pr: Projects Processes: Flow [SV-4]
Pj-Rm: Projects Roadmap [PV-2]
Step 2: Define Strategy and Capabilities [St - CV]
St-Pr: Strategic Processes: Strategic Vision [CV-1]
St-Tx: Strategic Taxonomy [CV-2]
St-Cn Strategic Connectivity [CV-4]
Op-Tr: Operational Traceability: Capability Performer Map [N/A]
Step 3: Define [Logical] Operational Architectures [Op - OV]
Op-Tx: Operational Taxonomy: High Level Operational Concepts [OV-1]
Op-Pr: Operational Processes: Flows [OV-5b]
Op-Tx: Operational Taxonomy [OV-2]
Rs-Tr: Resources Traceability: Operational Performer Implementation Map [N/A]
Step 4: Define Services Architectures [Sv - SvcV]
Sv-Tx: Services Taxonomy [SvcV-1/2]
Sv-Sr: Services Structure [SvcV-1/2]
Sv-Pr: Services Processes: Flow [SvcV-4]
Sv-Rm: Services Roadmap Evolution [SvcV-8]
Sv-Tx: Services Taxonomy: Actual Services [SvcV-1]
Step 5: Define [Implementation] Resource Architectures [Rs - SV]
Rs-Tx: Resources Taxonomy [SV-1/2]
Sd-Tx: Standards Taxonomy [StdV-1]
Rs-Sr: Resources Structure [SV-1/2]
Rs-Pr: Resources Processes: Flow [SV-4]
Rs-Rm: Resources Roadmap Evolution [SV-8]
Rs-Tx: Resources Taxonomy: Actual Resources [N/A]
Rq-Mv: Requirements: Resources [N/A]
Step 6: Define [Human] Personnel Architectures [Ps - SV]
Ps-Tx: Personnel Taxonomy [OV-4]
Ps-Sr: Personnel Structure [OV-4]
Ps-Pr: Personnel Processes: Flow [SV-4]
Ps-Rm: Personnel Roadmap Evolution [SV-8]
Ps-Tx: Personnel Taxonomy: Actual Organizational Resources [OV-4]
Step 7: Define [Protection] Security Architecture [Sc - SV]
Sc-Tx: Security Taxonomy [N/A]
Sc-Sr: Security Structure [N/A]
Sc-Pr: Security Processes: Flow [N/A]
Sc-Tr: Security Traceability: Threats to Assets Mapping [N/A]

NOTE to Architects: This Enterprise Architecture (EA) Guide describes a workflow for creating EA views in accordance with the Unified Architecture Framework Modeling Language (UAFML). UAF Specifications can be downloaded from the OMG webpage: www.omg.org/spec/UAF/1.1/About-UAF/.

The steps of the workflow are laid out in alignment with the stakeholder domains in the UAF for producing the requisite architecture views in each of those domains. These steps can be performed in any order and in practice are often done concurrently. The workflow connections between architecture views are intended to show how views can or should influence each other, not the order in which the views are created. It is not uncommon that many of the predecessor views already exist somewhere in one fom or another.

This architecture description method can be used in conjunction with processes for the conceptualization and evaluation of an architecture [ISO 42020 2019], and also used as the basis for an EA modeling methodology, architecture development planning, MBSE capability assessment, and modeling project organization. The Guide covers architecting of the enterprise as well as architecting (at a high level) of a major entity within the enterprise. The Guide can also be used by Systems Engineers to model complex systems, organizations and services

Enterprise Architecture Guide: Legend

🟦 - Main workflow for architecture construction
■ - Supporting workflows from additional architecture views within the domain
🟫 - Architecture cross-domain workflows
🟩 - Information and analysis influences upon external activities

Figure 4:1 - Architecture View Connectivity Between Major Steps

The depiction above shows workflow steps represented by the key view for that step down to the second level. The complete workflow is decomposed down to the third level to cover all views defined in the UAF specification. All steps in the workflow to the third level are shown on the next page. The “external” views that relate to each major step are shown on the right side of the workflow depiction for that step, some coming “in” and others coming “out.”

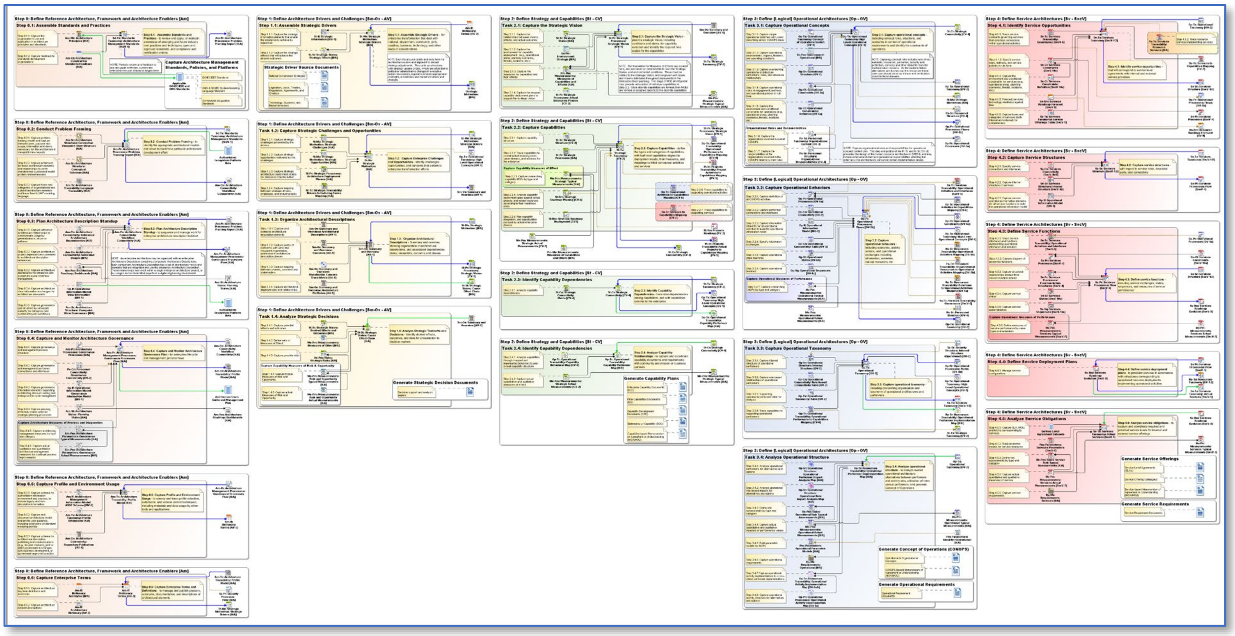


Figure 4:2 - Workflow Steps 0 - 4 at the 3rd Level of Decomposition

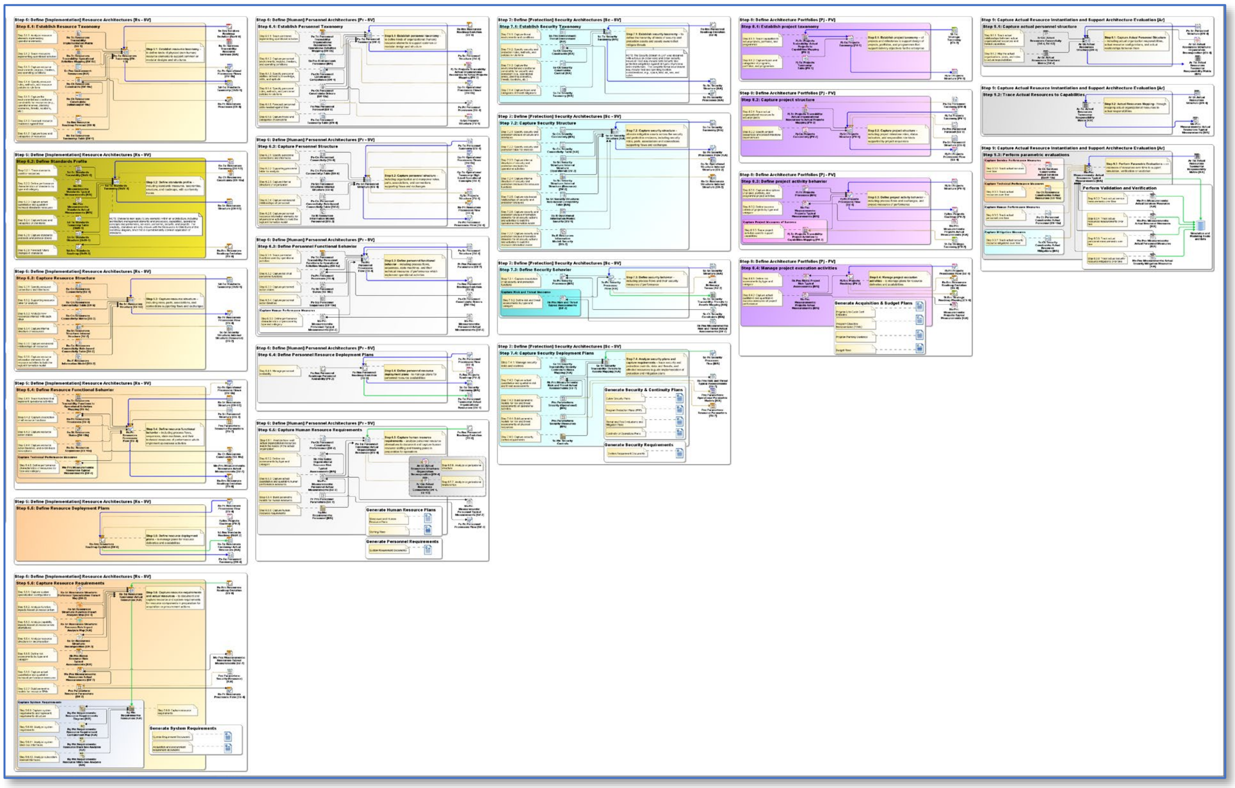


Figure 4:3 - Workflow Steps 5 - 9 at the 3rd Level of Decomposition

STEP 1 – Architecture Drivers & Challenges

4.1 Step 1 – Architecture Drivers and Challenges

Purpose. The purpose of this step is to identify those things that drive the enterprise to do what it does and the associated challenges that present difficulties in addressing these drivers. Key stakeholders for this step are Executive Managers, Strategic Planners, Program Managers and Enterprise Architects. Their concerns are mainly about what does the enterprise need to do to address the drivers and how do these drivers provide justification for what changes need to be made to the enterprise. They also have concerns about how the challenges can or will be addressed.

Conceptual Schema. The key concepts in the Strategic Viewpoint and the Summary and Overview Viewpoint that can be used as model elements in the architecture views and the relationships between these concepts are illustrated in the conceptual schema shown in Figure 4:4. These ***key concepts*** are highlighted in italics within the Narrative and some of the less obvious concepts are listed with the associated UAF meaning of that concept. Detailed definition of the entities and relationships shown in the conceptual schema are provided in the UAF Modeling Language (UAFML) specification document.

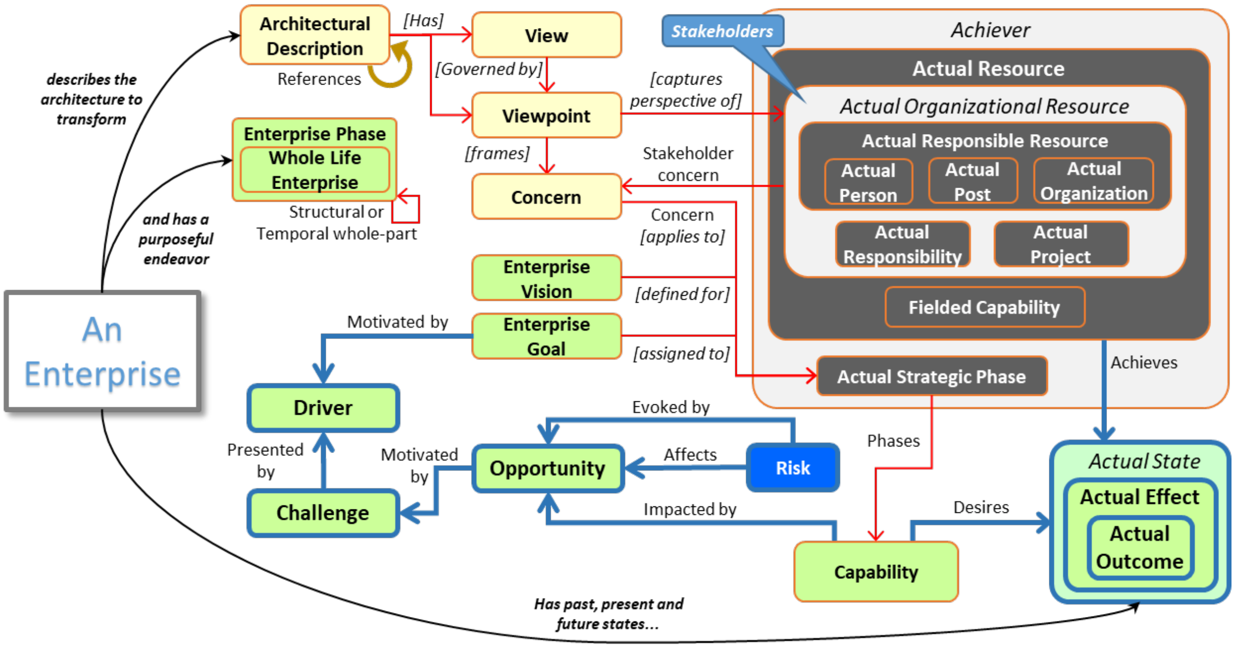


Figure 4:4 - Conceptual Schema for Architecture Drivers and Challenges

This Guide is intended to be used in conjunction with the UAF Sample Problem that defines architecture views for a Search and Rescue Mission. UAF Specification documents can be downloaded from the OMG webpage: www.omg.org/spec/UAF/About-UAF/.

Step 1.0 – Define Architecture Drivers and Challenges – The main entry criterion for this Step is bringing forward architecture management reference materials, architecture frameworks and utilities from the Architecture Management activities in Step 0 (and from relevant architecture repositories) with associated legacy architecture information, including governance and analysis process flows and architecture development workflows.

These are used as the foundation to begin development of the summary and overview of the architecture description effort and identification of its key *stakeholders*, which provides a planning guide that includes examination and monitoring of motivations, influencers, and contexts to align business and enterprise planning efforts, usually tied to business or program life cycle milestones and key decision points. The Summary and Overview architecture view defines the overall ***architecture*** in the context of the variety of endeavors, purposes, contexts, environments, and constraints that affect the ***challenges*** and ***opportunities*** for transformation of an enterprise.

- *Architecture* – fundamental concepts and properties related to an entity in its environment and governing principles for the realization and evolution of this entity and its related life cycle processes [ISO/IEC/IEEE 42020:2019] (Note: This architecture entity can be an enterprise or system or some other kind of thing. These fundamental concepts and properties can be about key entities and relationships, along with associated behaviors, which are characterized in an ***architectural description***.)

Drivers and ***effects*** are identified in order to frame the ***challenges*** and ***opportunities*** that serve as a basis for ***capabilities*** in the architecture, which in turn help to identify associated risks. It is common to focus on the desired "end effects" that would result from changes to the architecture where these end effects are called ***actual outcomes*** which are considered (and defined) as a special kind of ***effect***. Subsequently, the ensemble of ***actual effects*** that lead to these ***actual outcomes*** are identified. These planning efforts will guide assessments, decisions, and courses of action, steered by the influencing ***drivers***, for changing or transforming the enterprise.

When you identify the Drivers that apply to the enterprise and Outcomes that must be achieved, this facilitates more complete identification of relevant Challenges and Opportunities and in turn helps to identify the relevant capability gaps and shortfalls to be addressed in the future architecture. These gaps and shortfalls are measured in terms of the desired effects that lead to the expected outcomes. This flow from Drivers to Outcomes is illustrated in Figure 4:5.

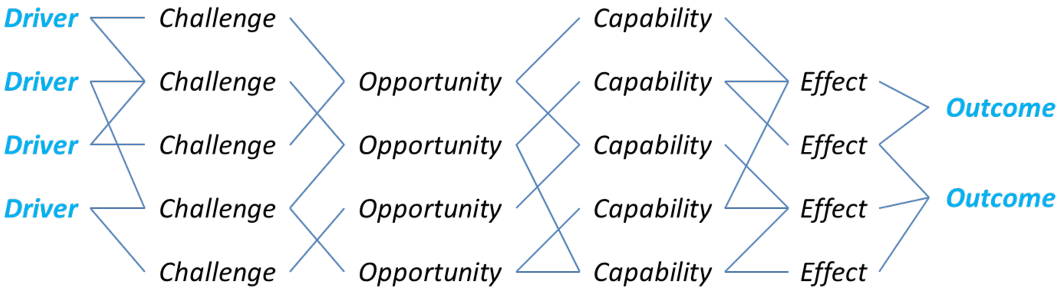


Figure 4:5 - Challenges and Opportunities Framed by Drivers and Outcomes

Workflow Summary. A summary level view of the four steps involved in the Step 1 workflow is shown below. The flowlines represent how one architecture view "influences" another architecture view. These lines do not represent a particular sequence of process activity execution or imply information exchanges. These should be thought of as influence diagrams rather than process diagrams. These architecture views (e.g., views, diagrams, tables) can be mapped to the architecture processes used in your organization and can be incorporated into an architecture modeling methodology.

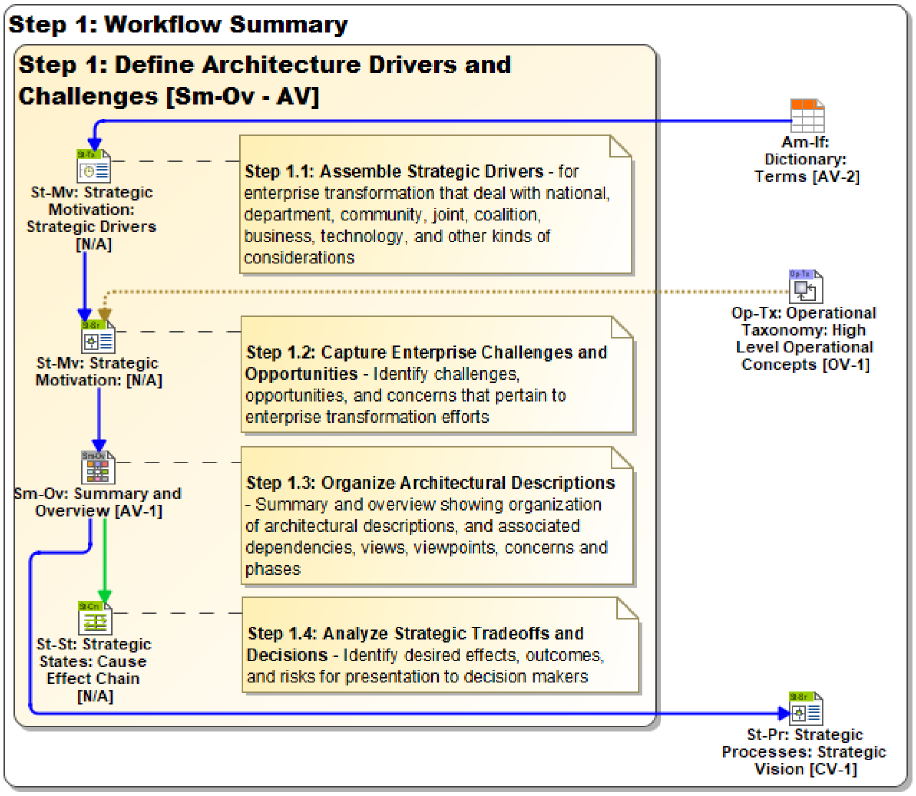

Step 1: Workflow Summary
Step 1: Define Architecture Drivers and Challenges [Sm-Ov - AV]
St-Mv: Strategic Motivation: Strategic Drivers [N/A]
Step 1.1: Assemble Strategic Drivers - for enterprise transformation that deal with national, department, community, joint, coalition, business, technology, and other kinds of considerations
Am-If: Dictionary: Terms [AV-2]
St-Mv: Strategic Motivation: [N/A]
Step 1.2: Capture Enterprise Challenges and Opportunities - Identify challenges, opportunities, and concerns that pertain to enterprise transformation efforts
Op-Tx: Operational Taxonomy: High Level Operational Concepts [OV-1]
Sm-Ov: Summary and Overview [AV-1]
Step 1.3: Organize Architectural Descriptions - Summary and overview showing organization of architectural descriptions, and associated dependencies, views, viewpoints, concerns and phases
St-St: Strategic States: Cause Effect Chain [N/A]
Step 1.4: Analyze Strategic Tradeoffs and Decisions - Identify desired effects, outcomes, and risks for presentation to decision makers
St-Pr: Strategic Processes: Strategic Vision [CV-1]

Figure 4:6 - Workflow Summary for Step 1: Define Architecture Drivers and Challenges

The second-level steps in the Step 1 workflow are listed below along with the UAF views (and corresponding [DoDAF views] where applicable) associated with those steps. A complete list of the detailed steps at the third level with their corresponding views is provided at the end of this section.

29	Step 1: Define Architecture Drivers and Challenges [SmOv - AV]	Views
30	**Step 1.1: Assemble Strategic Drivers** - for enterprise transformation that deal with national, department, community, joint, coalition, business, technology, or other kinds of considerations	St-Mv: Strategic Motivation: *Strategic Drivers* [N/A]
33	**Step 1.2: Capture Enterprise Challenges and Opportunities -** Identify challenges, opportunities, and concerns that pertain to enterprise transformation efforts	St-Mv: Strategic Motivation: [N/A]
38	**Step 1.3: Organize Architectural Descriptions** - Summary and overview showing organization of architectural descriptions, and associated dependencies, views, viewpoints, concerns and phases	Sm-Ov: Summary and Overview [AV-1]
43	**Step 1.4: Analyze Strategic Tradeoffs and Decisions** - Identify desired effects, outcomes, and risks for presentation to decision makers	St-St: Strategic States: *Cause Effect Chain* [N/A]

4.1.1 Strategic Drivers and Stakeholders

Step 1.1 – Assemble Strategic Drivers – An overarching set of ***drivers*** is assembled from various sources that will guide or can influence the direction of the enterprise. These ***drivers*** may come from strategic plans, competitive assessments, laws and regulations, treaties and other agreements, technology, business and market forecasts, organizational commissions and charters, operational demands, and other kinds of influencing source materials or elicited directly from stakeholders. These drivers are organized by structuring, relating, or containing them by source, topical similarities, or other factors. They can be presented in fit-for-purpose views used to filter, simplify, or summarize ***drivers*** for relevance to particular aspects of the enterprise and associated systems, products, and services.

Descriptions of these drivers retain the language of *stakeholders* in their own terms to aid in understanding or validating enterprise outcomes relative to the ***drivers***. An initial set of *stakeholders* (each one being represented in UAF by the element called ***organizational resource***) is identified, including but not limited to those related to these drivers.

- **Driver** – a factor which will have a significant impact on the activities and goals of an enterprise (Note: drivers may also relate to the purpose or charter of an enterprise)
- *Stakeholder* – an individual, team, organization, or classes thereof, having an interest in an **Strategic Phase** [ISO/IEC/IEEE 42010:2011] (Note: a stakeholder may be an individual or group (both internal and external to the enterprise) who has an interest in, or is affected by, outcomes or intermediate effects generated or influenced by the enterprise, or a role, position, individual, organization or classes thereof, having an interest, right, share, or claim, in an entity or its architecture)

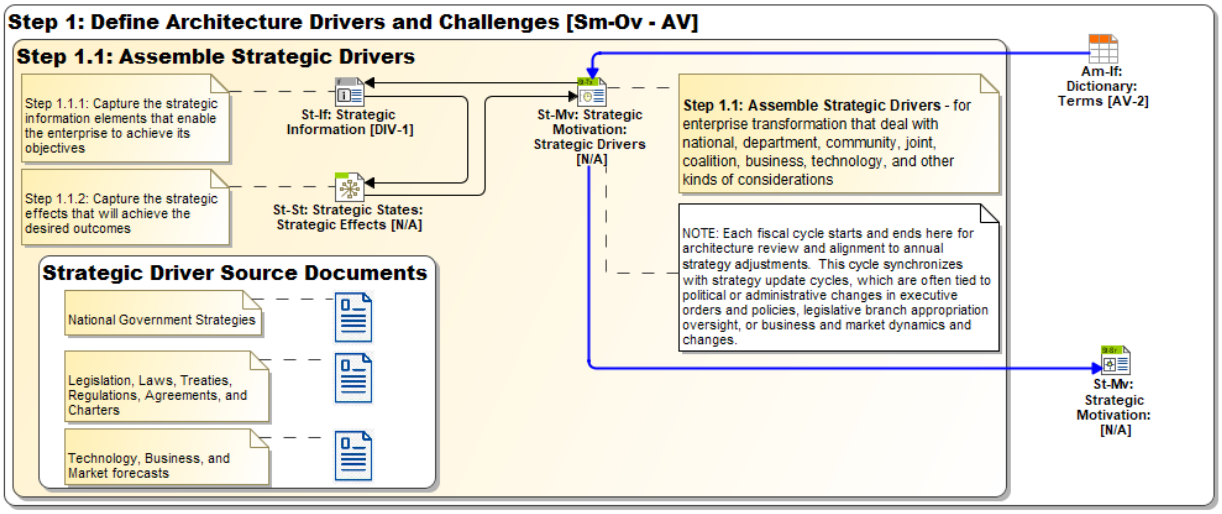

Step 1: Define Architecture Drivers and Challenges [Sm-Ov - AV]
Step 1.1: Assemble Strategic Drivers
Step 1.1.1: Capture the strategic information elements that enable the enterprise to achieve its objectives
St-If: Strategic Information [DIV-1]
St-Mv: Strategic Motivation: Strategic Drivers [N/A]
Step 1.1: Assemble Strategic Drivers - for enterprise transformation that deal with national, department, community, joint, coalition, business, technology, and other kinds of considerations
Am-If: Dictionary: Terms [AV-2]
Step 1.1.2: Capture the strategic effects that will achieve the desired outcomes
St-St: Strategic States: Strategic Effects [N/A]
NOTE: Each fiscal cycle starts and ends here for architecture review and alignment to annual strategy adjustments. This cycle synchronizes with strategy update cycles, which are often tied to political or administrative changes in executive orders and policies, legislative branch appropriation oversight, or business and market dynamics and changes.
Strategic Driver Source Documents
National Government Strategies
Legislation, Laws, Treaties, Regulations, Agreements, and Charters
Technology, Business, and Market forecasts
St-Mv: Strategic Motivation: [N/A]

Figure 4:7 - Step 1.1: Assemble Strategic Drivers

4.1.2 Enterprise Challenges and Opportunities

Step 1.2 – Capture Enterprise Challenges and Opportunities – A set of ***challenges*** are identified and organized with attributes geared toward understanding their feasibility of being addressed, and relative to impact on transformation of the enterprise. Additionally, a set of ***opportunities*** are identified and organized with attributes regarding the extent to which they can address the ***challenges***. For example, opportunities may be characterized by groupings such as doctrine, organization, training, materiel, leadership and education, personnel, and facilities (DOTMLPF). Alternatively, can use categories associated with the “defence lines of development” used in the UK: training, equipment, personnel, information, concepts and doctrine, organization, infrastructure, and logistics (TEPIDOIL).

Drivers are traced to their presented ***challenges***, and ***opportunities*** are traced to the ***challenges*** that motivate them. Grouping or bundling of opportunities may be organized by courses of action, business management areas, portfolios, or other conventions useful to the enterprise.

- **Challenge** – an existing or potential difficulty, circumstance, or obstacle which will require effort and determination from an enterprise to overcome in achieving its goals
- **Opportunity** – an existing or potential favorable circumstance or combination of circumstances which can be advantageous for addressing enterprise challenges

These ***opportunities*** and their associated courses of action will guide understanding of which ***operational*** and ***resource assets*** will be impacted, often in terms or measures of cost, benefit, ***risk***, and changes to mapped ***capabilities***, which frame the overall basis for understanding top-down overarching architectural tradeoffs.

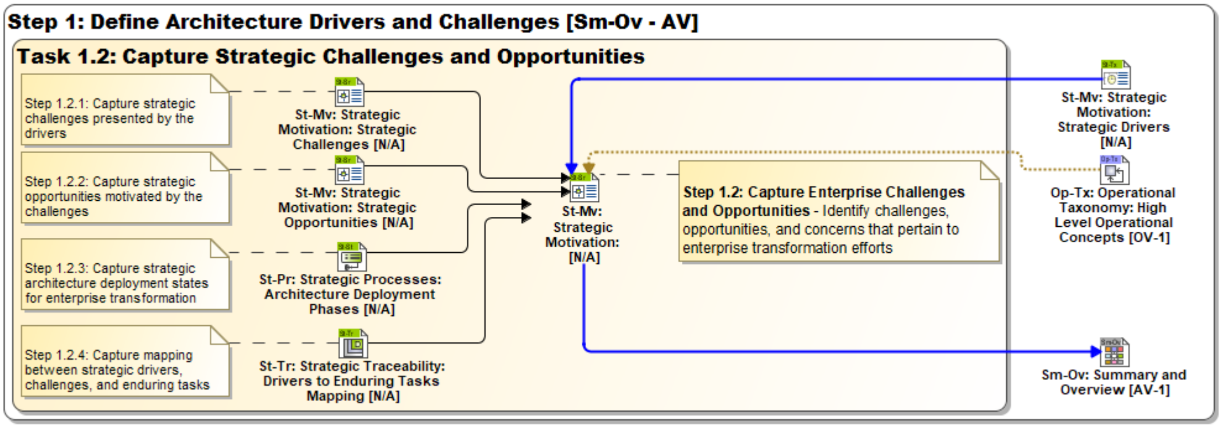

Step 1: Define Architecture Drivers and Challenges [Sm-Ov - AV]
Task 1.2: Capture Strategic Challenges and Opportunities
Step 1.2.1: Capture strategic challenges presented by the drivers
St-Mv: Strategic Motivation: Strategic Challenges [N/A]
Step 1.2.2: Capture strategic opportunities motivated by the challenges
St-Mv: Strategic Motivation: Strategic Opportunities [N/A]
Step 1.2.3: Capture strategic architecture deployment states for enterprise transformation
St-Pr: Strategic Processes: Architecture Deployment Phases [N/A]
Step 1.2.4: Capture mapping between strategic drivers, challenges, and enduring tasks
St-Tr: Strategic Traceability: Drivers to Enduring Tasks Mapping [N/A]
St-Mv: Strategic Motivation: [N/A]
Step 1.2: Capture Enterprise Challenges and Opportunities - Identify challenges, opportunities, and concerns that pertain to enterprise transformation efforts
St-Mv: Strategic Motivation: Strategic Drivers [N/A]
Op-Tx: Operational Taxonomy: High Level Operational Concepts [OV-1]
Sm-Ov: Summary and Overview [AV-1]

Figure 4:8 - Step 1.2: Capture Strategic Challenges and Opportunities

4.1.3 Architectural Description Structure

Step 1.3 – Organize Architectural Descriptions – *Architectural description* elements are captured in such a manner to describe an overall enterprise architecture, which can also include multiple sub-architectures or architecture versions. Typically, an enterprise architectural description contains planning and reference architecture elements, governed by enterprise life cycle activities, while a system architectural description contains architecture elements which are managed by system engineering life cycle activities.

An enterprise ***architectural description*** is dependent upon, and ***references***, individual system architectural descriptions, as well as relevant external architectural descriptions. ***Architectural description*** dependencies are then analyzed through ***architectural reference*** relationships, to aid in full comprehension of enterprise contexts, especially when other pre-existing or external architectures relate to the architecture of interest being described. An enterprise ***architectural description*** may also be composed of a collection of component ***architectural descriptions***, which could have dependencies that must be analyzed and assessed. Components in the enterprise are major items like ***missions, systems, services, organizations***, programs, facilities, etc. Each of these major items could have its own “component architecture” developed and described in a separate architecture description.

- **Architectural Description** – a work product to express the architecture of some system of interest. It provides an executive-level summary of information about the architecture description in a consistent form to allow quick reference and comparison between architecture descriptions. It includes assumptions, constraints and limitations that affect high-level decisions relating to an architecture-based work program.
- **Architectural Reference** – a tuple that specifies that one **Architectural Description** refers to another (Note: this reference may include dependencies)

Each ***architectural description*** is either decomposed into, or synthesizes, ***views*** and ***viewpoints***, which may be laid out in dashboards and other fit-for-purpose perspectives to aid in understanding an architectural plan. Each planned ***viewpoint*** should address specific ***concerns*** held by relevant ***stakeholders***. These ***concerns*** are assigned to ***actual enterprise phases*** in which phased deployments of capabilities will address both the ***concerns*** and ***opportunities***. Either in this Step or future Steps one or more ***operational architecture*** and ***resource architecture*** elements are defined and related to the ***actual enterprise phases***.

- **Concern** – interest in a **Strategic Phase** (Strategic Phase is synonym for System in ISO 42010) relevant to one or more of its stakeholders. (Note: a concern may be a “matter of relevance or importance to a stakeholder regarding an entity of interest” [ISO 42010] that will be addressed in an architecture)
- **View** –expresses the architecture of the system-of-interest in accordance with an architecture viewpoint (or simply, viewpoint). (Note: a view is an “information item, governed by an architecture viewpoint,

comprising part of an architecture description" [ISO 42010] that communicates some aspect of an architecture)

- **Viewpoint** –frames (to formulate or construct in a particular style or language) one or more concerns. A concern can be framed by more than one viewpoint. (Note: a viewpoint is a "convention for the creation, interpretation and use of an architecture view to frame one or more concerns" [ISO 42010] that governs the creation of views)
- **Actual Enterprise Phase** – an individual that describes the phase of an actual enterprise endeavor (Note: this is a period of time within an architecture during which capabilities are deployed that address concerns, and respond to planned courses of action)

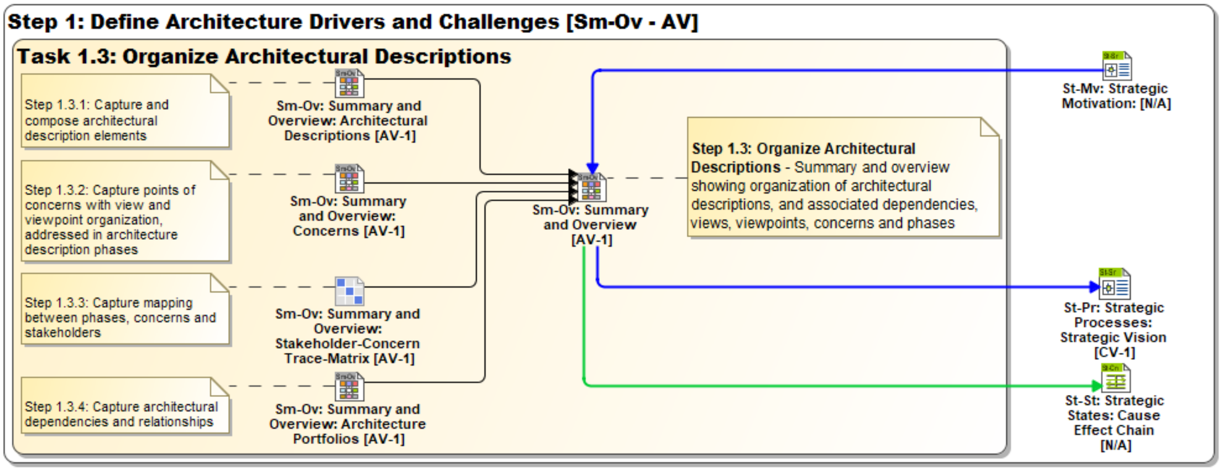

Step 1: Define Architecture Drivers and Challenges [Sm-Ov - AV]
Task 1.3: Organize Architectural Descriptions
Step 1.3.1: Capture and compose architectural description elements
Sm-Ov
Sm-Ov: Summary and Overview: Architectural Descriptions [AV-1]
Step 1.3.2: Capture points of concerns with view and viewpoint organization, addressed in architecture description phases
Sm-Ov
Sm-Ov: Summary and Overview: Concerns [AV-1]
Step 1.3.3: Capture mapping between phases, concerns and stakeholders
Sm-Ov: Summary and Overview: Stakeholder-Concern Trace-Matrix [AV-1]
Step 1.3.4: Capture architectural dependencies and relationships
Sm-Ov
Sm-Ov: Summary and Overview: Architecture Portfolios [AV-1]
Sm-Ov
Sm-Ov: Summary and Overview [AV-1]
Step 1.3: Organize Architectural Descriptions - Summary and overview showing organization of architectural descriptions, and associated dependencies, views, viewpoints, concerns and phases
St-Sr
St-Mv: Strategic Motivation: [N/A]
St-Sr
St-Pr: Strategic Processes: Strategic Vision [CV-1]
St-Cn
St-St: Strategic States: Cause Effect Chain [N/A]

Figure 4:9 - Step 1.3: Organize Architectural Descriptions

4.1.4 Strategic Tradeoffs and Decisions

Step 1.4 – Analyze Strategic Tradeoffs and Decisions – Once a summary and overview of an architecture has been captured and defined, analysis of high-level tradeoffs addressing stakeholder decision possibilities begins. ***Effects*** are identified that are made possible by each architecture based on the intrinsic drivers, challenges, and opportunity combinations. Resulting ***actual effects*** and ***actual outcomes*** are then described to help understand the final or ending results and end states that will come about due to the architecture. Following this, risks that could impact the enterprise or its stakeholders for all known points of interest are described and evaluated.

- **Actual Effect** – a real-world phenomenon that follows and is caused by some previous phenomenon. (Note: this is the realization of an **Effect**. An effect can lead to downstream effects or to one or more desired outcomes)
- **Actual Outcome** – an individual that describes something that happens or is produced as the final consequence or product and is related to one of the goals for the business or enterprise. Outcome is a special kind of effect, one that is usually at the end of a chain of effects, i.e., an "end effect".
- **Risk** – represents a situation involving exposure to danger of Affectable Elements (e.g., Assets, Processes, **Capabilities**, **Opportunities**, or **Enterprise Goals**) where the effects of such exposure can be characterized in terms of the likelihood of occurrence of a given threat and the potential adverse consequences of that threat's occurrence. (Note: A **Risk** will typically have an associated measure. The measure is used to capture the extent to which an entity is threatened by a potential circumstance or event. The **Risk** measure is typically a function of: (i) the adverse impacts that would arise if the circumstance or event occurs; and (ii) the likelihood of occurrence.)

When needed, an Architectural Description document can be developed based on the models and views generated during this Step 1. Ideally this Architectural Description document is automatically generated from the model itself using reporting scripts and model queries. Information and metadata in the document can include authorship, dates, planned timeframes, control authority, and other aspects essential to governance processes for the architecture.

Additionally, formal Business Value Model types of documents can be generated to aid in executive level decision making.

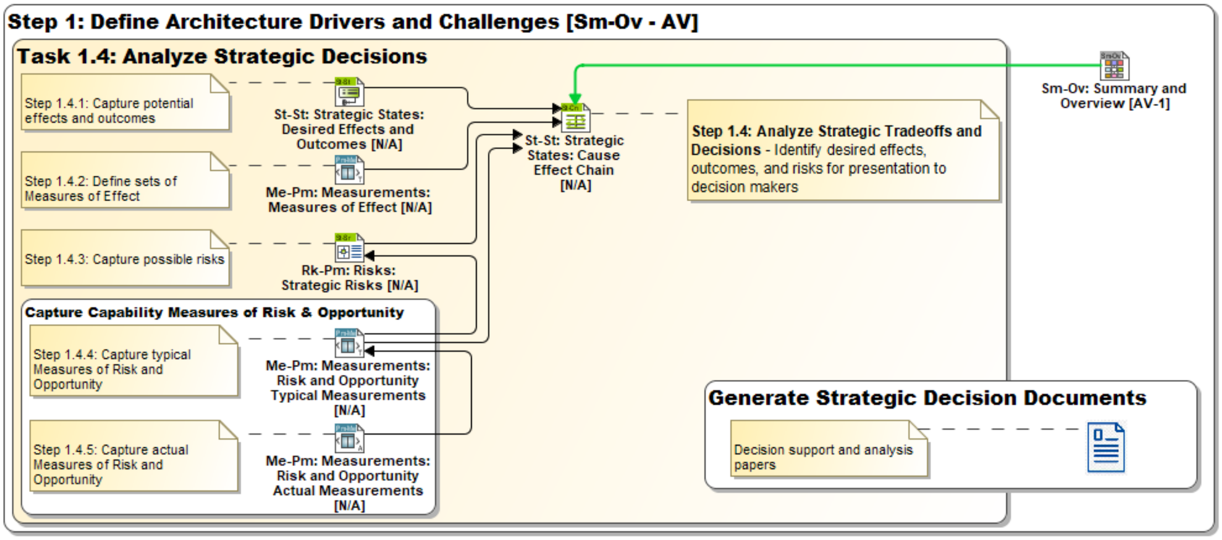
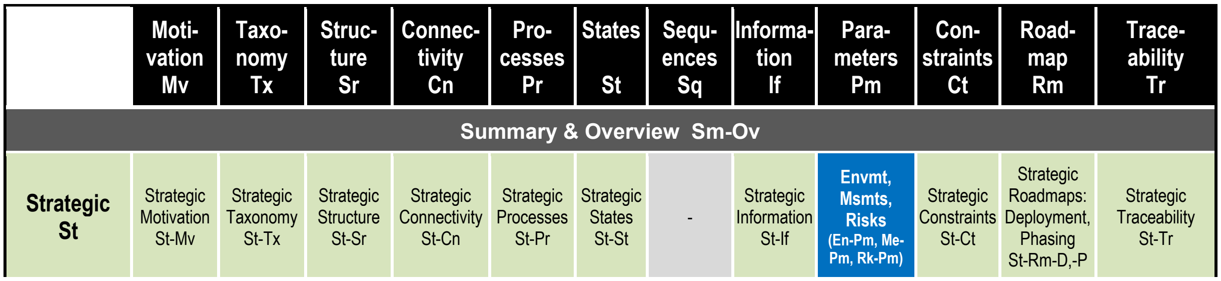

Step 1: Define Architecture Drivers and Challenges [Sm-Ov - AV]
Task 1.4: Analyze Strategic Decisions
Step 1.4.1: Capture potential effects and outcomes
St-St: Strategic States: Desired Effects and Outcomes [N/A]
Step 1.4.2: Define sets of Measures of Effect
Me-Pm: Measurements: Measures of Effect [N/A]
Step 1.4.3: Capture possible risks
Rk-Pm: Risks: Strategic Risks [N/A]
Capture Capability Measures of Risk & Opportunity
Step 1.4.4: Capture typical Measures of Risk and Opportunity
Me-Pm: Measurements: Risk and Opportunity Typical Measurements [N/A]
Step 1.4.5: Capture actual Measures of Risk and Opportunity
Me-Pm: Measurements: Risk and Opportunity Actual Measurements [N/A]
St-St: Strategic States: Cause Effect Chain [N/A]
Step 1.4: Analyze Strategic Tradeoffs and Decisions - Identify desired effects, outcomes, and risks for presentation to decision makers
Sm-Ov: Summary and Overview [AV-1]
Generate Strategic Decision Documents
Decision support and analysis papers

Figure 4:10 - Step 1.4: Analyze Strategic Decisions

4.1.5 Architecture View Summary for Step 1

The view specifications in UAF for this viewpoint are outlined here:



	Moti-vation Mv	Taxo-nomy Tx	Struc-ture Sr	Connec-tivity Cn	Pro-cesses Pr	States St	Sequ-ences Sq	Informa-tion If	Para-meters Pm	Con-straints Ct	Road-map Rm	Trace-ability Tr
Summary & Overview Sm-Ov												
Strategic St	Strategic Motivation St-Mv	Strategic Taxonomy St-Tx	Strategic Structure St-Sr	Strategic Connectivity St-Cn	Strategic Processes St-Pr	Strategic States St-St	-	Strategic Information St-If	**Envmt, Msmts, Risks (En-Pm, Me-Pm, Rk-Pm)**	Strategic Constraints St-Ct	Strategic Roadmaps: Deployment, Phasing St-Rm-D,-P	Strategic Traceability St-Tr

A summary of the lower-level steps and UAF views is shown below. The architecture views produced by each step are identified by the UAF grid designator Aa-Bb where Aa represents the Viewpoint row in the grid and Bb represents the Aspect column in the grid. DoDAF view designations are also shown in [brackets]. The name of the architecture view matches the name of the view specification in the UAFML. In some cases, a subtitle is provided to identify what kind of information is provided in an instance of that UAF view. There will be some architecture views that are not part of the UAF specification since you will sometimes need fit-for-purpose (i.e., custom) views that capture the necessary architectural information.

29	**Step 1: Define Architecture Drivers and Challenges [SmOv - AV]**	**Views**
30	**Step 1.1: Assemble Strategic Drivers** - for enterprise transformation that deal with national, department, community, joint, coalition, business, technology, and other kinds of considerations	St-Mv: Strategic Motivation: *Strategic Drivers* [N/A]
31	Step 1.1.1: Capture the strategic information elements that enable the enterprise to achieve its objectives	St-If: Strategic Information [DIV-1]
32	Step 1.1.2: Capture the strategic effects that will achieve the desired outcomes	St-St: Strategic States: *Strategic Effects* [N/A]
33	**Step 1.2: Capture Enterprise Challenges and Opportunities -** Identify challenges, opportunities, and concerns that pertain to enterprise transformation efforts	St-Mv: Strategic Motivation: [N/A]
34	Step 1.2.1: Capture strategic challenges presented by the drivers	St-Mv: Strategic Motivation: *Strategic Challenges* [N/A]
35	Step 1.2.2: Capture strategic opportunities motivated by the challenges	St-Mv: Strategic Motivation: *Strategic Opportunities* [N/A]
36	Step 1.2.3: Capture strategic architecture deployment states for enterprise transformation	St-Pr: Strategic Processes: *Architecture Deployment Phases* [N/A]
37	Step 1.2.4: Capture mapping between strategic drivers, challenges, and enduring tasks	St-Tr: Strategic Traceability: *Drivers to Enduring Tasks Mapping* [N/A]
38	**Step 1.3: Organize Architectural Descriptions** - Summary and overview showing organization of architectural descriptions, and associated dependencies, views, viewpoints, concerns and phases	Sm-Ov: Summary and Overview [AV-1]
39	Step 1.3.1: Capture and compose architectural description elements	Sm-Ov: Summary and Overview: *Architectural Descriptions* [AV-1]
40	Step 1.3.2: Capture points of concerns with view and viewpoint organization, addressed in architecture description phases	Sm-Ov: Summary and Overview: *Concerns* [AV-1]
41	Step 1.3.3: Capture mapping between phases, concerns and stakeholders	Sm-Ov: Summary and Overview: *Stakeholder-Concern Trace Matrix* [AV-1]

42	Step 1.3.4: Capture architectural dependencies and relationships	Sm-Ov: Summary and Overview: *Architecture Portfolios* [AV-1]
43	**Step 1.4: Analyze Strategic Tradeoffs and Decisions** - Identify desired effects, outcomes, and risks for presentation to decision makers	St-St: Strategic States: *Cause Effect Chain* [N/A]
44	Step 1.4.1: Capture potential effects and outcomes	St-St: Strategic States: *Desired Effects and Outcomes* [N/A]
45	Step 1.4.2: Define sets of Measure of Effect (MOE) parameters	Me-Pm: Measurements: *Measures of Effect* [N/A]
46	Step 1.4.3: Capture possible risks evoked by strategic opportunities to be pursued	Rk-Pm: Risks: *Strategic Risks* [N/A]
47	Step 1.4.4: Capture typical Measures of Risk and Opportunity	Me-Pm: Measurements: *Risk and Opportunity Typical Measurements* [N/A]
48	Step 1.4.5: Capture actual Measures of Risk and Opportunity	Me-Pm: Measurements: *Risk and Opportunity Actual Measurements* [N/A]

STEP 2 – Enterprise Strategy & Capabilities

4.2 Step 2 – Enterprise Strategy and Capabilities

Purpose. The purpose of this step is to describe the capability taxonomy, composition of capabilities, dependencies between capabilities, and evolution of the capabilities. Key stakeholders for this step are Capability Portfolio Managers. Their concerns are mainly about identification of capability gaps and shortfalls and managing the evolution of capability deployments to address those gaps and shortfalls.

Conceptual Schema. The key concepts in the Strategic Viewpoint that can be used as model elements in the architecture views and the relationships between these concepts are illustrated in the conceptual schema shown in Figure 4:11. These ***key concepts*** are highlighted in italics within the Narrative and some of the less obvious concepts are listed with the associated UAF meaning of that concept. Detailed definitions of the entities and relationships shown in the conceptual schema are provided in the UAF Modeling Language (UAFML) specification document.

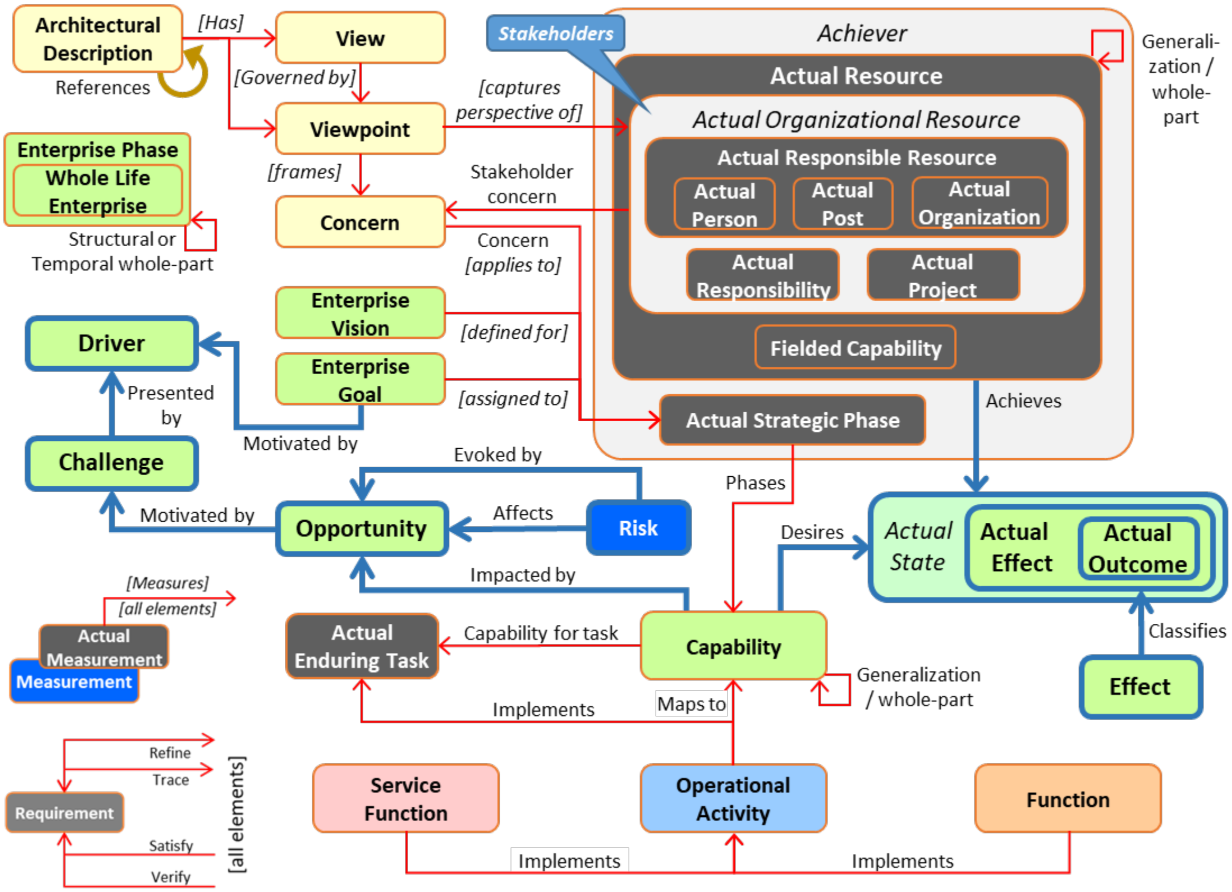


Figure 4:11 - Conceptual Schema for Enterprise Strategy and Capabilities

This Guide is intended to be used in conjunction with the UAF Sample Problem that defines architecture views for a Search and Rescue Mission. UAF Specification documents (including the Sample Problem) can be downloaded from the OMG webpage: www.omg.org/spec/UAF/About-UAF/.

Step 2.0 – Define Strategy and Capabilities – The main entry criterion for this Step is bringing forward the ***architectural description*** from Step 1 with an initial scope of ***architectures***, ***actual strategic phases***, and their ***stakeholder concerns***. These are used to define a strategic ***vision*** based on an enterprise plan for the ***whole life***

enterprise consisting of ***phases*** of deployments for ***capabilities***. ***Capabilities*** are defined that can produce ***desired effects*** meeting the ***goals*** assigned to each deployment ***phase*** of the enterprise.

- **Whole Life Enterprise** – a purposeful endeavor of any size involving people, organizations, and supporting systems made up of temporal and structural parts
- **Capability** – an enterprise's ability to achieve a desired effect realized through a combination of ways and means (e.g., **Capability Configurations**) along with specified measures

A ***Whole Life Enterprise*** can be made up of temporal parts and/or structural parts, and it often corresponds either to mission or business areas representing the essential tasks and functions of an organization, or a collection of organizations, that participate in the venture (which we usually call the "enterprise). ***Capabilities*** can achieve ***desired effects*** using ways (e.g., activities and behaviors) and means (e.g., physical and human resources) under certain ***conditions*** to perform ***enduring tasks***.

An example of a temporal part of the Whole Life Enterprise would be an enterprise phase as illustrated in Figure 4:12. An example of a structural part of the Enterprise Phase would be a value stream (modeled as a Strategic Process) composed of value stream stages. An example of a structural part of the Whole Life Enterprise would be an Enterprise Mission.

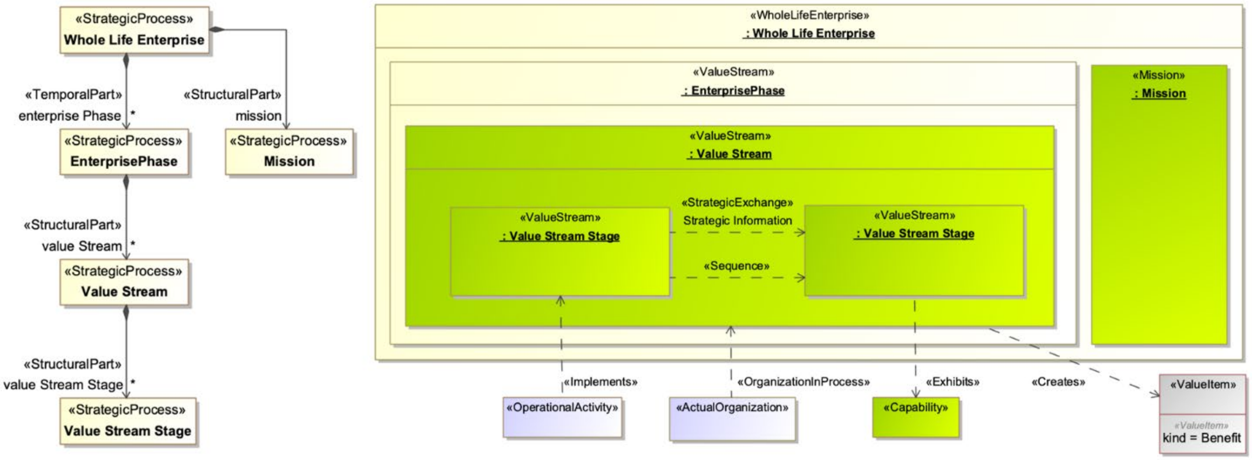


Figure 4:12 - Using Enterprise Phase as a Temporal Part and Mission as a Structural Part

Because ***capabilities*** provide and produce effects, they can be measured in terms of such an effect. A Measure of Effect (MOE) parameter describes qualitative or quantitative states or levels that directly relate to outcomes. Outcomes are sought by stakeholders, whose interests and perspectives are addressed by the ***actual measurements*** of effects. These outcome measurements are different from measurements of associated ways and means.

Capabilities map to existing or future ***operational activities*** which are traced to ***operational performers*** that can perform those activities.

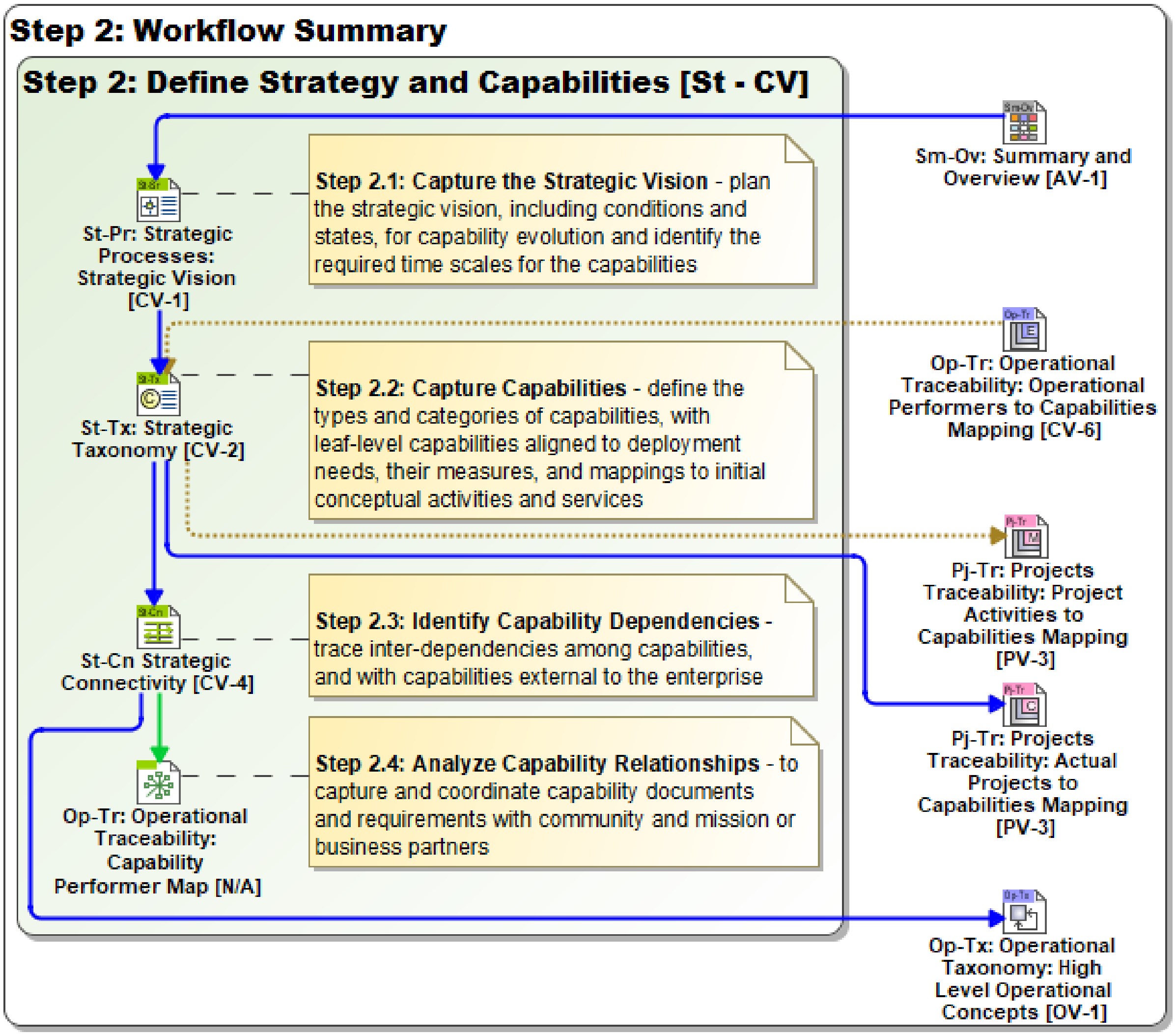

Step 2: Workflow Summary
Step 2: Define Strategy and Capabilities [St - CV]
St-Pr: Strategic Processes: Strategic Vision [CV-1]
Step 2.1: Capture the Strategic Vision - plan the strategic vision, including conditions and states, for capability evolution and identify the required time scales for the capabilities
St-Tx: Strategic Taxonomy [CV-2]
Step 2.2: Capture Capabilities - define the types and categories of capabilities, with leaf-level capabilities aligned to deployment needs, their measures, and mappings to initial conceptual activities and services
St-Cn Strategic Connectivity [CV-4]
Step 2.3: Identify Capability Dependencies - trace inter-dependencies among capabilities, and with capabilities external to the enterprise
Op-Tr: Operational Traceability: Capability Performer Map [N/A]
Step 2.4: Analyze Capability Relationships - to capture and coordinate capability documents and requirements with community and mission or business partners
Sm-Ov: Summary and Overview [AV-1]
Op-Tr: Operational Traceability: Operational Performers to Capabilities Mapping [CV-6]
Pj-Tr: Projects Traceability: Project Activities to Capabilities Mapping [PV-3]
Pj-Tr: Projects Traceability: Actual Projects to Capabilities Mapping [PV-3]
Op-Tx: Operational Taxonomy: High Level Operational Concepts [OV-1]

Figure 4:13 - Workflow Summary for Step 2: Define Strategy and Capabilities

Workflow Summary. A summary level view of the four steps involved in Step 2 workflow is shown above. The flowlines represent how one architecture view "influences" another architecture view. These lines do not represent a particular sequence of process activity execution or imply information exchanges. These should be thought of as influence diagrams rather than process diagrams. These architecture views (e.g., views, diagrams, tables) can be mapped to the architecture processes used in your organization and can be incorporated into an architecture modeling methodology.

The second-level steps in the Step 2 workflow are listed below along with the UAF views (and corresponding [DoDAF views] where applicable) associated with those steps. A complete list of the detailed steps at the third level with their corresponding views is provided at the end of this section.

49	**Step 2: Define Strategy and Capabilities [St – CV]**	**Views**
50	**Step 2.1: Capture Strategic Vision** – plan the strategic vision, including conditions and states, for capability evolution and identify the required time scales for the capabilities	St-Pr Strategic Processes: *Strategic Vision* [CV-1]

55	**Step 2.2: Capture Capabilities** – define the types and categories of capabilities, with leaf-level capabilities aligned to deployment needs, their measures, and mappings to initial conceptual activities and services	St-Tx: Strategic Taxonomy [CV-2]
63	**Step 2.3: Identify Capability Dependencies** – trace inter-dependencies among capabilities, and with capabilities external to the enterprise	St-Cn Strategic Connectivity [CV-4]
65	**Step 2.4: Analyze Capability Relationships** –capture and coordinate capability documents and requirements with community and mission or business partners	Op-Tr: Operational Traceability: *Capability Performer Map* [N/A]

4.2.1 Strategic Vision

Step 2.1 – Capture Strategic Vision – ***Enterprise visions*** are defined which may be guided by preliminary plans and courses of action developed through strategy tradeoff decisions in Step 1.4. These ***enterprise visions*** are assigned in a ***strategic motivation*** to all ***actual strategic phases*** that have been laid out, structured, and defined within the ***whole life enterprise***. More detailed and supporting ***enterprise goals*** and ***enterprise objectives*** for each ***actual strategic phase*** are defined, including broad start and end point calendar dates for each phase, what ***actual organization*** may be the recipient of, or responsible for, the phase, and for its associated future ***operational*** and ***resource architectures***. All actual strategic phases are organized in an enterprise-wide table which aids strategic planners with enterprise life cycle decision making.

Actual strategic phases may be of four kinds. An ***Actual Enterprise Phase*** is simply a period of time when a set of capabilities are deployed by an associated operational and resource architecture. An ***Enterprise Mission*** is a kind of ***Actual Enterprise Phase*** that uses set of deployed capabilities with the sense of a specific purpose to realize a vision. An ***Actual Enduring Task*** is a kind of ***Actual Enterprise Phase*** that uses a set of deployed capabilities with a sense of permanence and may not include an end date. A ***Value Stream*** is a kind of ***Actual Strategic Phase*** that uses a step or long-lived series of steps that creates value through a ***Value Item***.

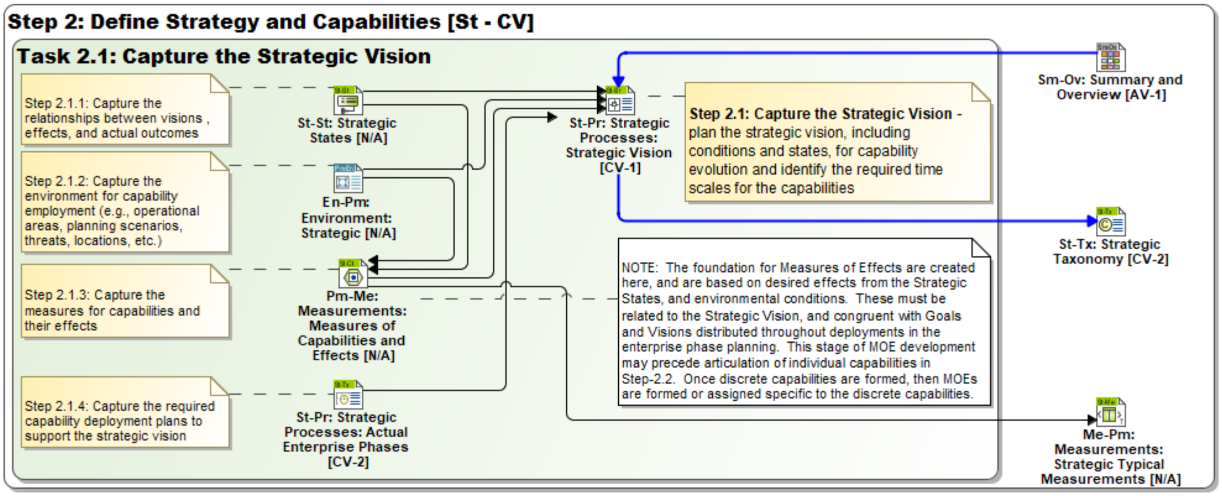
A strategic vision employing ***Value Streams*** will typically use a series of value streams to deliver worth through ***Value Items*** created by the value stream series. This method of capability planning is often associated with production-oriented enterprises mapping current and future states to improve production efficiencies.

- **Enterprise Vision** – describes the future state of the enterprise, without regard to how it is to be achieved. [BMM: OMG dtc-13-08-24]
- **Enterprise Goal** – a statement about a state or condition of the enterprise to be brought about or sustained through appropriate means. **An Enterprise Goal** amplifies an **Enterprise Vision,** that is it indicates what must be satisfied on a continuing basis to effectively attain the **Enterprise Vision**. [BMM: OMG dtc-13-08-24]
- **Enterprise Objective** – a statement of an attainable, time-targeted, and measurable target that the enterprise seeks to meet in order to achieve its goals. [BMM: 1.3]
- **Strategic Phase** –a type of a current or future state of the enterprise, mission, **Value Stream** or **Enduring Task**. (Note: there are four kinds of an actual strategic phase: an **Actual Enterprise Phase**, an **Enterprise Mission**, an **Actual Enduring Task**, or a **Value Stream**)
- **Actual Enterprise Phase** – an individual that describes the phase of an actual enterprise endeavor. (Note: this is a time period within which a set of capabilities are deployed that address concerns, and respond to planned courses of action)
- **Enterprise Mission** – captures at a high level what you will do to realize your vision
- **Actual Enduring Task** – an actual undertaking recognized by an enterprise as being essential to achieving its goals, i.e., a strategic specification of what the enterprise does
- **Value Stream** – an end-to-end collection of activities that create a result for a customer, who may be the ultimate customer or an internal end-user of the value stream. Value stream nested within another value stream may represent Value Stream Stage - a distinct, identifiable phase or step within a value stream [The Business Architecture Metamodel Guide, 2020]

- **Value Item** – an ideal, custom, or institution that an enterprise promotes or agrees with. It may be positive or negative, depending on point of view. (Note: it is a description of worth created by a **Value Stream**. There are seven kinds of a value stream: time, cost, quality, revenue, benefit, KPI, loss, or other.)
- **Actual Organization** - an actual formal or informal organizational unit, e.g., "Driving and Vehicle Licensing Agency", "UAF team Alpha". (Note: it is a particular organizational unit that can be assigned responsibility for delivering resources, performing resource functions, or handling other assignments)

Desired ***effects*** developed in Step 1 are laid out in ***desires*** relationships between initially developed ***capabilities*** and implementations of those capabilities. Other than the ***effects***, many of these elements and capabilities may not be known yet, or this may include existing ***capabilities*** that will be improved. As new capabilities are defined, they are added to these ***strategic states***, whereby effects and groupings of effects are typically aligned with ***enduring tasks*** and ***strategic phase*** constructs that help achieve the strategic vision.

- **Desires** – a tuple relating the **Desirer** (i.e., a **Capability** or **Organizational Resource**) to an **Actual State** (Note: such as an **Actual Effect**)
- **Achieves**– a tuple that exists between an **Actual State** (e.g., observed/measured during testing) of an element that attempts to achieve a desired effect and an **Achiever**. (Note: for an example, such as a **Desirer** to an **Actual Effect**)
- **Actual Effect** – a real world phenomenon that follows and is caused by some previous phenomenon (Note: this is the realization of an **Effect**. An effect can that could lead to downstream effects or to one or more desired outcomes)
- **Actual Outcome** – an individual that describes something that happens or is produced as the final consequence or product and is related to one of the goals for the business or enterprise. Outcome is a special kind of effect, one that is usually at the end of a chain of effects, i.e., an "end effect".


Step 2: Define Strategy and Capabilities [St - CV]
Task 2.1: Capture the Strategic Vision
St-St: Strategic States [N/A]
En-Pm: Environment: Strategic [N/A]
Pm-Me: Measurements: Measures of Capabilities and Effects [N/A]
St-Pr: Strategic Processes: Actual Enterprise Phases [CV-2]
St-Pr: Strategic Processes: Strategic Vision [CV-1]
Sm-Ov: Summary and Overview [AV-1]
St-Tx: Strategic Taxonomy [CV-2]
Me-Pm: Measurements: Strategic Typical Measurements [N/A]

Step 2.1.1: Capture the relationships between visions , effects, and actual outcomes

Step 2.1.2: Capture the environment for capability employment (e.g., operational areas, planning scenarios, threats, locations, etc.)

Step 2.1.3: Capture the measures for capabilities and their effects

Step 2.1.4: Capture the required capability deployment plans to support the strategic vision

Step 2.1: Capture the Strategic Vision - plan the strategic vision, including conditions and states, for capability evolution and identify the required time scales for the capabilities

NOTE: The foundation for Measures of Effects are created here, and are based on desired effects from the Strategic States, and environmental conditions. These must be related to the Strategic Vision, and congruent with Goals and Visions distributed throughout deployments in the enterprise phase planning. This stage of MOE development may precede articulation of individual capabilities in Step-2.2. Once discrete capabilities are formed, then MOEs are formed or assigned specific to the discrete capabilities.

Figure 4:14 - Step 2.1: Capture the Strategic Vision

The total environment expected for the deploying capabilities is defined in terms of ***conditions***, ***locations***, ***environments***, including kinds of environments and ***location kinds***, as well as ***geo-political*** factors. These ***conditions*** include all known attributes which will affect ***capabilities*** and will become an overarching conditional context for parameters across all other viewpoints of the architecture (e.g., Operational, Resource, Personnel, Security, etc.).

Measure of Effect sets defined in Step 1 are refined, decomposed, and applied to any pre-existing ***capabilities*** and capability areas.

4.2.2 Capability Taxonomy

Step 2.2 – Capture Capabilities – Specific capabilities are defined, related to each other through ***generalization, aggregation,*** or ***composition***, and related to ***actual strategic phases*** such as ***enduring tasks***, ***value streams, actual enterprise phases,*** or ***missions*** that they support. When large enterprises contain multiple systems that have similar capabilities which together comprise the enterprise capability, these capabilities are decomposed into parts to provide a whole capability. An overarching ***capability*** arrangement is supported by ***taxonomy tables*** and by ***enduring task mappings***. ***Capability kinds*** may be identified to differentiate strategic, operational, service, resource, personnel, security, or other types of capabilities. The kinds of capabilities can be denoted by selecting the appropriate kind attribute for that element (since there are no separate stereotypes for the different kinds).

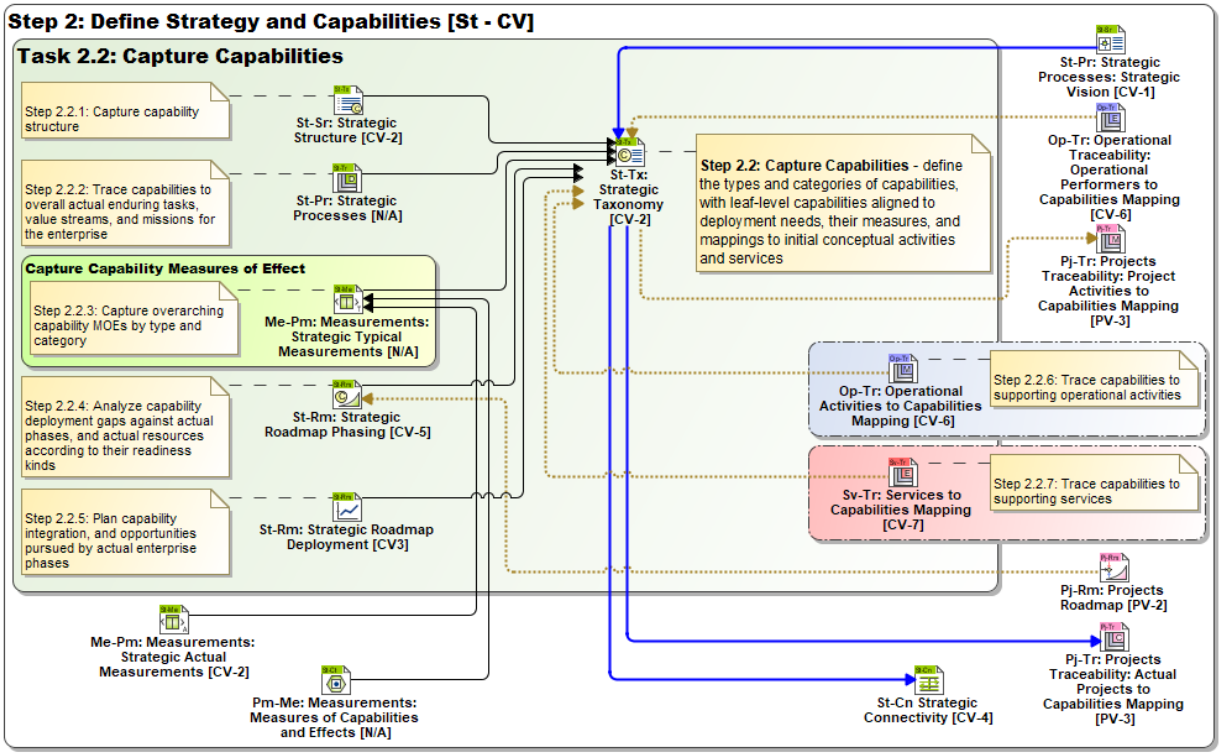
Specific tables of typical Measures of Effect are given to each capability.

- **Capability** – an enterprise's ability to achieve a desired effect realized through a combination of ways and means (e.g., **Capability Configurations**) along with specified measures
- **Actual Strategic Phase** – a phase of an **Actual Enterprise Mission**, **Value Stream** or **Actual Enduring Task** endeavor
- **Measurement** – a property of an element representing something in the physical world, expressed in amounts of a unit of measure (Note: a measurement is a reusable unit or scale which can be used to describe the value of a property qualitatively or quantitatively. A **Measurement** is just the ruler while the **Actual Measuremen**t is the ruler with an associated value/number. Thus, the ruler (measurement), as a part property of an element, is intended to be used (reused) to generate *many* actual measurements of that element over time. It may be typed by a Value Type or Enumeration. An instance of a measurement is an actual measurement which has an associated point or period of time, an intention (actual, required or estimate), and which may be named to indicate the purpose of the measurement, such as threshold or objective, or some point in a lifecycle such when the measured item, especially a resource, is in fabrication or employment.)

Initial capability ***deployment*** and ***phasing roadmaps*** are generated to start capturing points in time when planned achievement of ***capabilities*** will occur and their planned or expected deployment to ***actual organizations***. ***Capabilities*** and ***actual enterprise phases*** which have now been defined are shown in the roadmaps, along with any other existing partial architecture information from other Steps which already exist. As architecture is developed in the other Steps, these roadmaps will become complete. Key to these roadmaps are ***actual project milestones*** which define when ***resource performers*** come into existence to ***implement*** the capabilities. These milestones will convey when a resource is ***in service***, ***out of service, deployed*** or ***no longer used***. When a resource goes in or out of service, this may indicate that the resource is deploying, retiring, has ceased its existence, or has been taken offline for modifications, upgrades, or put in to a reserve or residual state.

- **Actual Project Milestone** – an event with a start date in an **Actual Project** from which progress is measured (Note: this event is described with a specific and standardized **Date Time**)
- **Resource Readiness Kind** – a particular enumeration of the type of readiness for a resource providing a capability: deployed, in service, out of service, no longer used, or other

Bridging between Step 2 and Step 3, capabilities may be immediately mapped to existing ***operational activities*** and ***services***, or new ones that don't exist may be defined to correspond to the capabilities.

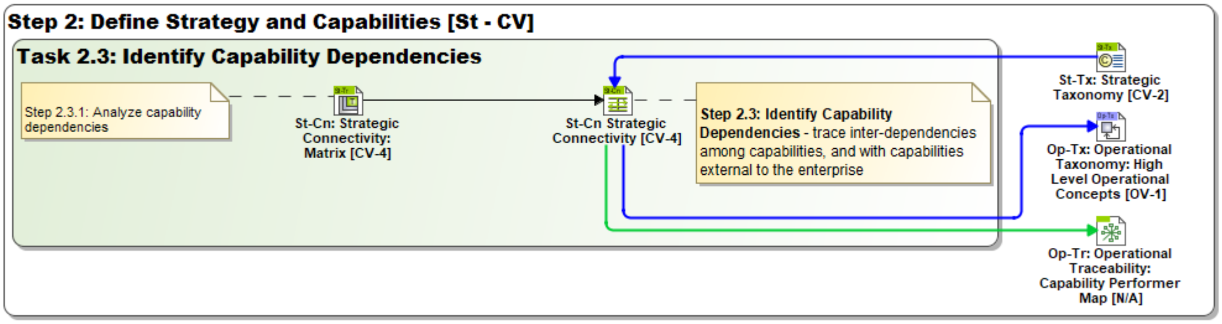


Step 2: Define Strategy and Capabilities [St - CV]
Task 2.2: Capture Capabilities
Step 2.2.1: Capture capability structure
St-Sr: Strategic Structure [CV-2]
Step 2.2.2: Trace capabilities to overall actual enduring tasks, value streams, and missions for the enterprise
St-Pr: Strategic Processes [N/A]
Capture Capability Measures of Effect
Step 2.2.3: Capture overarching capability MOEs by type and category
Me-Pm: Measurements: Strategic Typical Measurements [N/A]
Step 2.2.4: Analyze capability deployment gaps against actual phases, and actual resources according to their readiness kinds
St-Rm: Strategic Roadmap Phasing [CV-5]
Step 2.2.5: Plan capability integration, and opportunities pursued by actual enterprise phases
St-Rm: Strategic Roadmap Deployment [CV3]
St-Tx: Strategic Taxonomy [CV-2]
Step 2.2: Capture Capabilities - define the types and categories of capabilities, with leaf-level capabilities aligned to deployment needs, their measures, and mappings to initial conceptual activities and services
St-Pr: Strategic Processes: Strategic Vision [CV-1]
Op-Tr: Operational Traceability: Operational Performers to Capabilities Mapping [CV-6]
Pj-Tr: Projects Traceability: Project Activities to Capabilities Mapping [PV-3]
Op-Tr: Operational Activities to Capabilities Mapping [CV-6]
Step 2.2.6: Trace capabilities to supporting operational activities
Sv-Tr: Services to Capabilities Mapping [CV-7]
Step 2.2.7: Trace capabilities to supporting services
Pj-Rm: Projects Roadmap [PV-2]
Me-Pm: Measurements: Strategic Actual Measurements [CV-2]
Pm-Me: Measurements: Measures of Capabilities and Effects [N/A]
St-Cn Strategic Connectivity [CV-4]
Pj-Tr: Projects Traceability: Actual Projects to Capabilities Mapping [PV-3]

Figure 4:15 - Step 2.2: Capture Capabilities

4.2.3 Capability Dependencies

Step 2.3 – Identify Capability Dependencies – Capabilities of an enterprise can be related to capabilities from either within the enterprise, or from other inter-dependent enterprises. Dependency relationships are defined for these inter-dependencies in a ***strategic connectivity*** view. These relationships indicate that one capability cannot fully provide its effect without the existence of the other capability. These dependencies imply that the activities and resources used for other capabilities may either not exist with the dependent enterprise, or that capabilities within an enterprise scope are differentiated because their activities and resources are separated.



Step 2: Define Strategy and Capabilities [St - CV]
Task 2.3: Identify Capability Dependencies
Step 2.3.1: Analyze capability dependencies
St-Cn: Strategic Connectivity: Matrix [CV-4]
St-Cn Strategic Connectivity [CV-4]
Step 2.3: Identify Capability Dependencies - trace inter-dependencies among capabilities, and with capabilities external to the enterprise
St-Tx: Strategic Taxonomy [CV-2]
Op-Tx: Operational Taxonomy: High Level Operational Concepts [OV-1]
Op-Tr: Operational Traceability: Capability Performer Map [N/A]

Figure 4:16 - Step 2.3: Identify Capability Dependencies

Strategic connectivity dependencies may align with understandings, agreements, cooperation, partnerships, and other alliances between separate and distinct enterprises that join them together. The strategic architecture of an enterprise may include description of capabilities from another enterprise which are either connected through ***strategic connectivity***, or which form a separate architecture of that joint venture.

4.2.4 Capabilities Analysis and Plans

Step 2.4 – Analyze Capability Relationships – Once a strategic architecture has been captured and defined, analysis of overall structure, organization, and mappings begins. Operational activities which map to capabilities are examined to aid in activity mapping decisions, and analysis will be performed of alternative capability structure and associated operational activity layouts. Services which govern exchanges in operational activities that map to the capabilities are examined to understand gaps and comprehensiveness of service provisioning. Tradeoffs are evaluated in the analysis of alternatives to help determine differences in service consumption levels by the operational activities.

- **Maps to Capability** – a tuple denoting that an activity contributes to providing a **Capability**

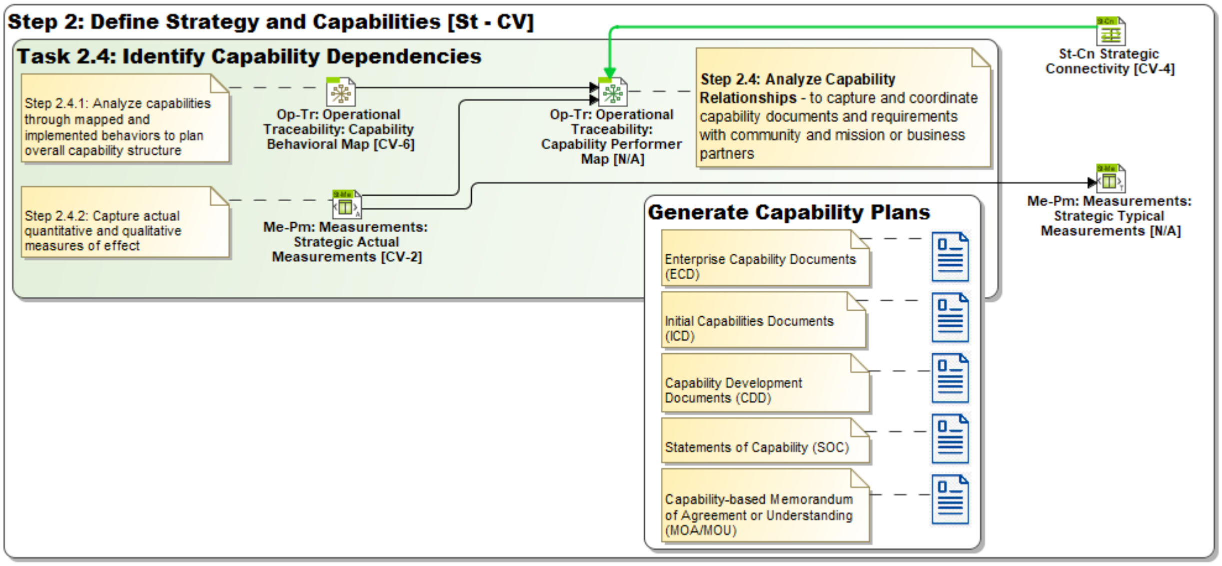

Step 2: Define Strategy and Capabilities [St - CV]
Task 2.4: Identify Capability Dependencies
Step 2.4.1: Analyze capabilities through mapped and implemented behaviors to plan overall capability structure
Op-Tr: Operational Traceability: Capability Behavioral Map [CV-6]
Op-Tr: Operational Traceability: Capability Performer Map [N/A]
Step 2.4: Analyze Capability Relationships - to capture and coordinate capability documents and requirements with community and mission or business partners
St-Cn Strategic Connectivity [CV-4]
Step 2.4.2: Capture actual quantitative and qualitative measures of effect
Me-Pm: Measurements: Strategic Actual Measurements [CV-2]
Me-Pm: Measurements: Strategic Typical Measurements [N/A]
Generate Capability Plans
Enterprise Capability Documents (ECD)
Initial Capabilities Documents (ICD)
Capability Development Documents (CDD)
Statements of Capability (SOC)
Capability-based Memorandum of Agreement or Understanding (MOA/MOU)

Figure 4:17 - Step 2.4: Analyze Capability Relationships

Typical ***measurements*** from Step 2.2 are used to generate ***actual measurements*** of the capabilities as they deploy and evolve over the enterprise life cycle[3]. Each actual measurement table is assigned start and end dates of their actual, estimated, or required points of existence, and are used to track validation, satisfaction, and realization of capability MOE changes over time.

- **Actual Measurement** – an actual value that is applied to a **Measurement** (Note: a measurement may be one of three kinds: actual, required, or estimate, and may have an associated start and/or end date)
- **Actual Measurement Kind** – an enumerated type of an actual measurement kind which is based on a required value (Note: a measurement may be one of three kinds: **Actual**, **Required**, or **Estimate**, and may have an associated start and/or end date)

Allocations from capabilities down to services, activities and their associated performers and agents is essential to determining capability plans. Capabilities may be reformed, refactored, modified, or restructured to respond to existing enterprise resources, or to force change to existing enterprise resources. Capability structure analysis aids in identification of dormant, unproductive, or antiquated resources and their associated activities, as well as services

[3] A **measurement** is merely the unit or scale for making a measure. It may be typed by a **Value Type** or **Enumeration**. An instance of a **measurement** is an **actual measurement** which has an associated point or period of time, an intention (actual, required or estimate), and which may be named to indicate the purpose of the measurement, such as threshold or objective, or some point in a lifecycle such when the measured item, especially a resource, is in fabrication or employment.

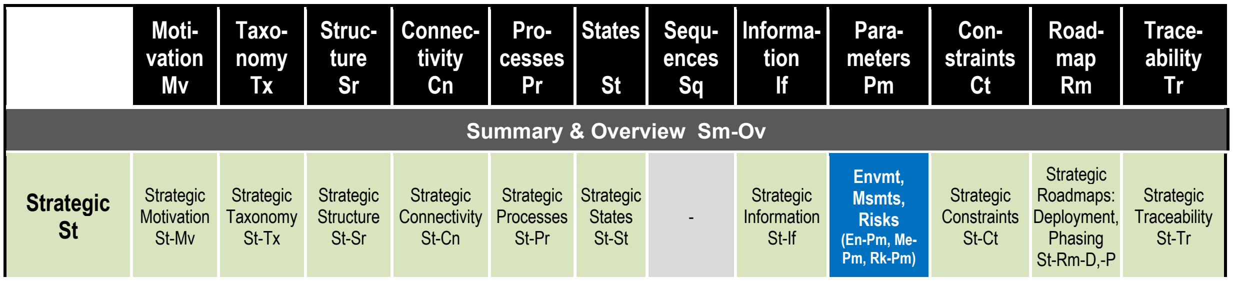
and service functions that need change or divestment. As well, this same analysis aids in designing modularity of new resources, activities and services that may better enable transformation and change.

An entire strategic architecture, or capability set, with all of its elements (capabilities, measures, goals, visions, tasks, etc.) may be considered as one option in an analysis of alternatives within the model. In other words, each alternative to be examined can be represented by a separate strategic architecture.

When needed, capability documents can be developed based on the models and views generated during this Step 2. These may include Strategic Plans, Vision Documents, formal Statements of Capability (SOC), Capability Development Documents (CDD), Initial Capability Documents (ICD), Enterprise Capability Documents (ECD), or other formal business strategic documents and plans. To support joining of enterprise endeavors, formal capability-based agreements and memorandum of understanding documents may be generated delineating capability boundaries, agreed-upon measures, and division of responsibilities between organizations participating in the enterprise or joint venture.

4.2.5 Architecture View Summary for Step 2

The view specifications in UAF for this viewpoint are outlined here:



	Moti-vation Mv	Taxo-nomy Tx	Struc-ture Sr	Connec-tivity Cn	Pro-cesses Pr	States St	Sequ-ences Sq	Informa-tion If	Para-meters Pm	Con-straints Ct	Road-map Rm	Trace-ability Tr
Summary & Overview Sm-Ov												
Strategic St	Strategic Motivation St-Mv	Strategic Taxonomy St-Tx	Strategic Structure St-Sr	Strategic Connectivity St-Cn	Strategic Processes St-Pr	Strategic States St-St	-	Strategic Information St-If	**Envmt, Msmts, Risks (En-Pm, Me-Pm, Rk-Pm)**	Strategic Constraints St-Ct	Strategic Roadmaps: Deployment, Phasing St-Rm-D,-P	Strategic Traceability St-Tr

A summary of the lower-level steps and UAF views is shown below. The architecture views produced by each step are identified by the UAF grid designator Aa-Bb where Aa represents the Viewpoint row in the grid and Bb represents the Aspect column in the grid. DoDAF view designations are also shown in [brackets]. The name of the architecture view matches the name of the view specification in the UAFML. In some cases, a subtitle is provided to identify what kind of information is provided in an instance of that UAF view. There will be some architecture views that are not part of the UAF specification since you will sometimes need fit-for-purpose (i.e., custom) views that capture the necessary architectural information.

49	**Step 2: Define Strategy and Capabilities [St – CV]**	**Views**
50	**Step 2.1: Capture Strategic Vision** – plan the strategic vision, including conditions and states, for capability evolution and identify the required time scales for the capabilities	St-Mv: Strategic Processes: *Strategic Vision* [CV-1]
51	Step 2.1.1: Capture the relationships between visions, effects, and actual outcomes	St-St: Strategic States [N/A]
52	Step 2.1.2: Capture the environment for capability employment (e.g., operational areas, planning scenarios, threats, locations, etc.)	En-Pm: Environment: *Strategic* [N/A]
53	Step 2.1.3: Capture the measures for capabilities and their effects	Pm-Me: Measurements: *Measures of Capabilities and Effects* [N/A]
54	Step 2.1.4: Capture the required capability deployment plans to support the strategic vision	St-Pr: Strategic Processes: *Actual Enterprise Phases* [CV-2]
55	**Step 2.2: Capture Capabilities** – define the types and categories of capabilities, with leaf-level capabilities	St-Tx: Strategic Taxonomy [CV-2]

	aligned to deployment needs, their measures, and mappings to initial conceptual activities and services	
56	Step 2.2.1: Capture capability structure	St-Sr: Strategic Structure [CV-2]
57	Step 2.2.2: Trace capabilities to overall actual enduring tasks, value streams, and missions for the enterprise	St-Pr: Strategic Processes [N/A]
58	Step 2.2.3: Capture overarching capability MOEs by type and category	Me-Pm: Measurements: *Strategic Typical Measurements* [N/A]
59	Step 2.2.4: Analyze capability deployment gaps against actual phases, and actual resources according to their readiness kinds	St-Rm: Strategic Roadmap Phasing [CV-3]
60	Step 2.2.5: Plan capability integration, and opportunities pursued by actual enterprise phases	St-Rm: Strategic Roadmap Deployment [CV-5]
61	Step 2.2.6: Trace capabilities to supporting operational activities	Op-Tr: Operational Traceability: *Operational Activities to Capabilities Mapping* [CV-6]
62	Step 2.2.7: Trace capabilities to supporting services	Sv-Tr: Services Traceability: *Services to Capabilities Mapping* [CV-7]
63	**Step 2.3: Identify Capability Dependencies** – trace inter-dependencies among capabilities, and with capabilities external to the enterprise	St-Cn Strategic Connectivity [CV-4]
64	Step 2.3.1: Analyze capability dependencies	St-Cn: Strategic Connectivity: *Matrix* [CV-4]
65	**Step 2.4: Analyze Capability Relationships** – to capture and coordinate capability documents and requirements with community and mission/business partners	Op-Tr: Operational Traceability: *Capability Performer Map* [N/A]
66	Step 2.4.1: Analyze capabilities through mapped and implemented behaviors to plan overall capability structure	Op-Tr: Operational Traceability: *Capability Behavioral Map* [CV-6]
67	Step 2.4.2: Capture actual quantitative and qualitative measures of effect	Me-Pm: Measurements: *Strategic Actual Measurements* [CV-2]

STEP 3 – Operational Architectures

4.3 Step 3 – Operational Architectures

Purpose. The purpose of this step is to describe the requirements, operational behavior, structure, and exchanges required to support (i.e., exhibit) capabilities. Key stakeholders for this step are Executives, Business Architects, Business Managers, Operations Managers and Mission Directors. Their concerns are mainly about what operational actors and activities are needed to meet enterprise objectives and what is the logical architecture of the enterprise.

Conceptual Schema. The key concepts in the Operational Viewpoint that can be used as model elements in the architecture views and the relationships between these concepts are illustrated in the conceptual schema shown in Figure 4:18. These ***key concepts*** are highlighted in italics within the Narrative and some of the less obvious concepts are listed with the associated UAF meaning of that concept. Detailed definition of the entities and relationships shown in the conceptual schema are provided in the UAFML specification document.

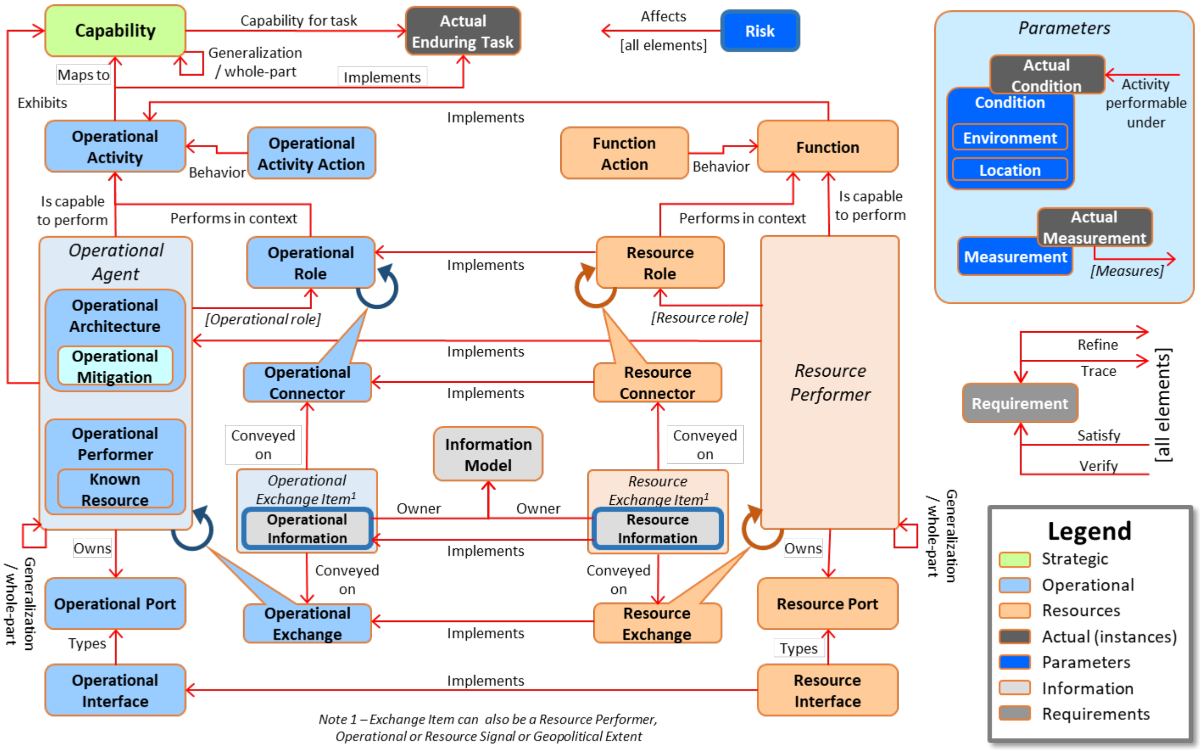


Figure 4:18 - Conceptual Schema for Operational Architectures

The operational architecture concepts are shown on the left in blue and the resource and personnel architecture concepts are shown on the right in orange and white, respectively. ***Resource performers*** and the ***functions*** performed by those performers will "implement" the ***operational agents*** and ***operational activities*** defined in the Operational Architecture.

Likewise, ***resource roles, resource connectors, resource exchange items, resource exchanges,*** and ***resource interfaces*** will implement the analogous things in the ***operational architecture***. Decisions about which resource elements (and what kinds of these elements) will implement operational elements are made in Steps 5 and 6 that deal with enterprise Resources and Personnel, respectively.

It is common to consider the operational architecture to be a "logical" architecture in the sense that physical implementation decisions are deferred to downstream architecture decisions and tradeoffs (a solution-independent architecture). Logical in this sense means connecting ideas in a sensible way, based on the rules of logic or formal argument. In other words, the logical architecture is what reasonably "follows" from the drivers, challenges, opportunities, desired effects, and capabilities defined in Steps 1 and 2.

This Guide is intended to be used in conjunction with the UAF Sample Problem that defines architecture views for a Search and Rescue Mission. UAF Specification documents (including the Sample Problem) can be downloaded from the OMG webpage: www.omg.org/spec/UAF/About-UAF/.

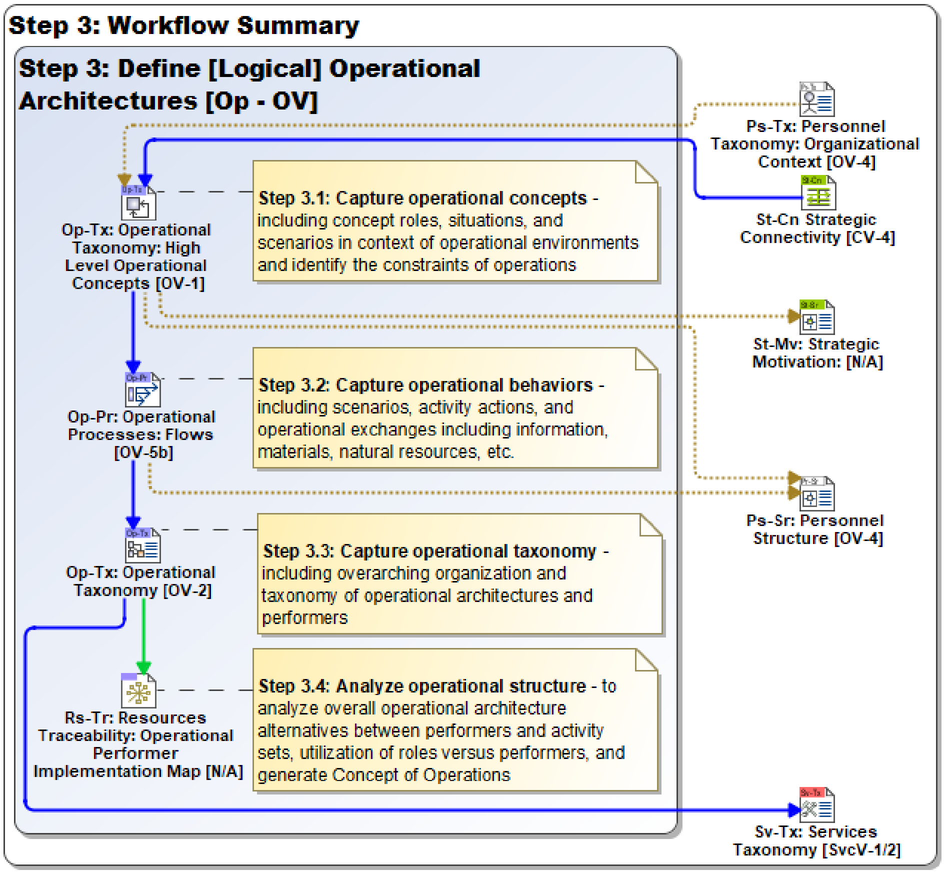
Step 3.0 – Define [Logical] Operational Architectures – The main entry criterion for this Step is bringing forward the ***capabilities*** from Step 2 with associated ***measures***, ***visions***, ***goals,*** and their strategic context. These are used to define associated concepts of operation using logical (unimplemented) ***operational agents***, performing activities that compose an ***operational architecture*** element which provides those ***capabilities***.

UAF has a special relationship called "Implements." An operational concept is a solution to the capabilities. A resource architecture is a solution to the operational architecture. The "implements" usage in UAF is sometimes confusing. For example, it must be understood that Personnel is not the same as or equivalent to Operational. That is the key here. The "unimplemented" as applied to ***operational agents*** is intentional in this regard.

- **Operational Agent** – an abstract type grouping **Operational Architecture** and **Operational Performer** (Note: this is an entity that can interact with other operational agents to perform operational activities. As an abstract grouping element for **Operational Performers** and **Operational Architectures** it cannot appear as an element on a diagram. See conceptual schema above.)
- **Operational Architecture** – a type used to denote a model of the Architecture, described from the Operational perspective (Note: this represents a large composition or aggregation of operational agents, which is itself an operational agent, that is described from the operational perspective)
- **Operational Performer** – a logical entity that **Is Capable To Perform Operational Activities** which produce, consume and process **Resources** (Note: this represents an individual or simple operational agent)

An Operational Architecture defines operational *behavior* elements (e.g., processes, states, sequences) and allocates these to operational *structure* elements (e.g., operational architectures, performers, known resources to be used in the operational setting, roles, connectors, exchange items, exchanges, ports and interfaces). The behavior elements define operational expectations for the ***operational activities*** that map to ***capabilities*** that will in turn help achieve enterprise goals.

Workflow Summary. A summary level view of the four steps involved in Step 3 workflow is shown below. The flowlines represent how one architecture view "influences" another architecture view. These lines do not represent a particular sequence of process activity execution or imply information exchanges. These should be thought of as influence diagrams rather than process diagrams. These architecture views (e.g., views, diagrams, tables) can be mapped to the architecture processes used in your organization and can be incorporated into an architecture modeling methodology.


Step 3: Workflow Summary
Step 3: Define [Logical] Operational Architectures [Op - OV]
Op-Tx: Operational Taxonomy: High Level Operational Concepts [OV-1]
Op-Pr: Operational Processes: Flows [OV-5b]
Op-Tx: Operational Taxonomy [OV-2]
Rs-Tr: Resources Traceability: Operational Performer Implementation Map [N/A]
Ps-Tx: Personnel Taxonomy: Organizational Context [OV-4]
St-Cn Strategic Connectivity [CV-4]
St-Mv: Strategic Motivation: [N/A]
Ps-Sr: Personnel Structure [OV-4]
Sv-Tx: Services Taxonomy [SvcV-1/2]

Step 3.1: Capture operational concepts - including concept roles, situations, and scenarios in context of operational environments and identify the constraints of operations

Step 3.2: Capture operational behaviors - including scenarios, activity actions, and operational exchanges including information, materials, natural resources, etc.

Step 3.3: Capture operational taxonomy - including overarching organization and taxonomy of operational architectures and performers

Step 3.4: Analyze operational structure - to analyze overall operational architecture alternatives between performers and activity sets, utilization of roles versus performers, and generate Concept of Operations

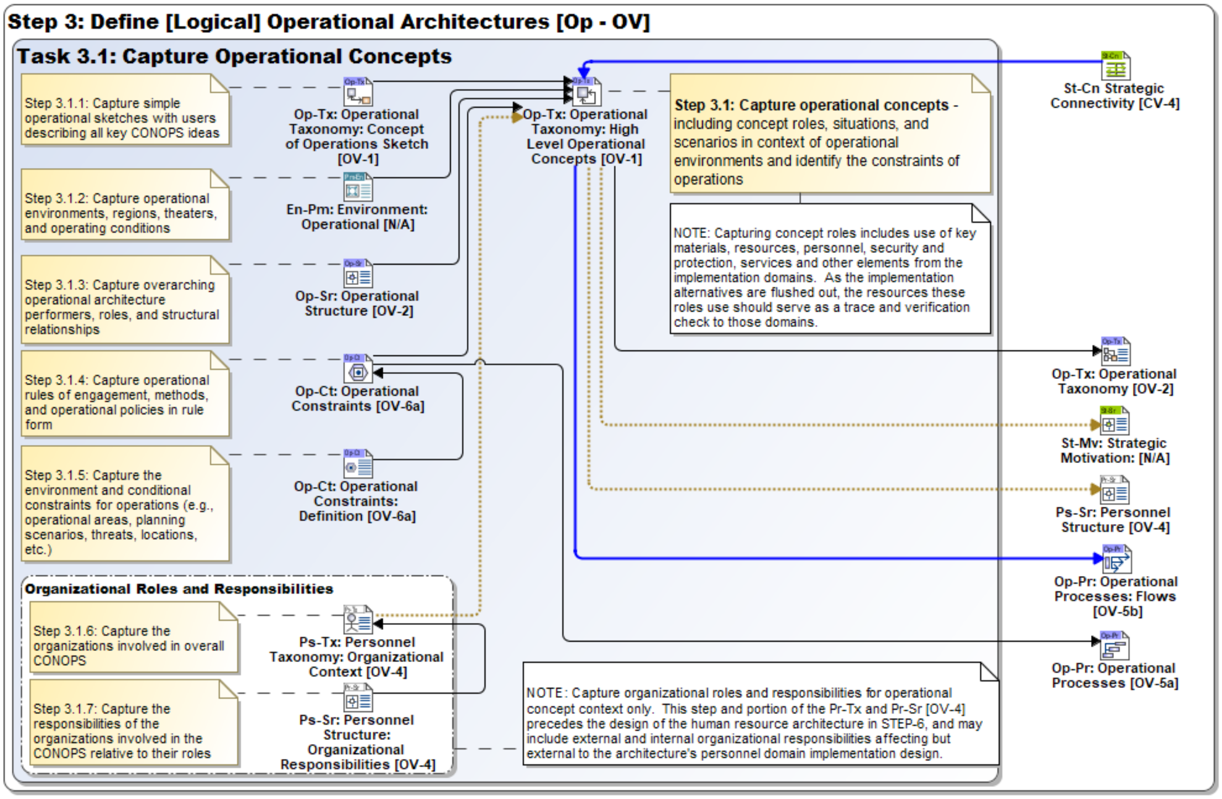
Figure 4:19 - Workflow Summary for Step 3: Define [Logical] Operational Architectures

The second-level steps in the Step 3 workflow are listed below along with the UAF views (and corresponding [DoDAF views] where applicable) associated with those steps. A complete list of the detailed steps at the third level with their corresponding views is provided at the end of this section.

68	**Step 3: Define [Logical] Operational Architectures [Op - OV]**	**Views**
69	**Step 3.1: Capture operational concepts** - including concept roles, situations, and scenarios in context of operational environments and identify the constraints of operations	Op-Tx: Operational Taxonomy: *High Level Operational Concepts* [OV-1]
77	**Step 3.2: Capture operational behaviors** - including scenarios, activity actions, and operational exchanges including information, materials, natural resources, etc.	Op-Pr: Operational Processes: *Flows* [OV-5b]
85	**Step 3.3: Capture operational taxonomy** - including overarching organization and taxonomy of operational architectures and performers	Op-Tx: Operational Taxonomy [OV-2]
90	**Step 3.4: Analyze operational structure** - to analyze overall operational architecture alternatives between performers and activity sets, utilization of roles versus performers, and generate Concept of Operations	Rs-Tr: Resources Traceability: *Operational Performer Implementation Map* [N/A]

4.3.1 Operational Concepts

Step 3.1 – Capture Operational Concepts – An overarching set of performers are described in ***high-level operational concepts*** by their participant's roles as ***concept items*** and their ***connections*** with each other, in a scoped context composed of ***conditions***, ***environments***, and ***locations***. All ***rules***, ***policies*** and other ***operational constraints*** are listed and applied to ***operational agents***, and then later expanded to their associated actions and exchanges. When ***concept items*** come from pre-existing ***known resources*** as well as other ***resources*** and ***organizations***, those are captured as well since they represent a known scoping ***constraint***.



Step 3: Define [Logical] Operational Architectures [Op - OV]
Task 3.1: Capture Operational Concepts
Step 3.1.1: Capture simple operational sketches with users describing all key CONOPS ideas
Op-Tx: Operational Taxonomy: Concept of Operations Sketch [OV-1]
Step 3.1.2: Capture operational environments, regions, theaters, and operating conditions
En-Pm: Environment: Operational [N/A]
Step 3.1.3: Capture overarching operational architecture performers, roles, and structural relationships
Op-Sr: Operational Structure [OV-2]
Step 3.1.4: Capture operational rules of engagement, methods, and operational policies in rule form
Op-Ct: Operational Constraints [OV-6a]
Step 3.1.5: Capture the environment and conditional constraints for operations (e.g., operational areas, planning scenarios, threats, locations, etc.)
Op-Ct: Operational Constraints: Definition [OV-6a]
Organizational Roles and Responsibilities
Step 3.1.6: Capture the organizations involved in overall CONOPS
Ps-Tx: Personnel Taxonomy: Organizational Context [OV-4]
Step 3.1.7: Capture the responsibilities of the organizations involved in the CONOPS relative to their roles
Ps-Sr: Personnel Structure: Organizational Responsibilities [OV-4]
Op-Tx: Operational Taxonomy: High Level Operational Concepts [OV-1]
Step 3.1: Capture operational concepts - including concept roles, situations, and scenarios in context of operational environments and identify the constraints of operations
NOTE: Capturing concept roles includes use of key materials, resources, personnel, security and protection, services and other elements from the implementation domains. As the implementation alternatives are flushed out, the resources these roles use should serve as a trace and verification check to those domains.
NOTE: Capture organizational roles and responsibilities for operational concept context only. This step and portion of the Pr-Tx and Pr-Sr [OV-4] precedes the design of the human resource architecture in STEP-6, and may include external and internal organizational responsibilities affecting but external to the architecture's personnel domain implementation design.
St-Cn Strategic Connectivity [CV-4]
Op-Tx: Operational Taxonomy [OV-2]
St-Mv: Strategic Motivation: [N/A]
Ps-Sr: Personnel Structure [OV-4]
Op-Pr: Operational Processes: Flows [OV-5b]
Op-Pr: Operational Processes [OV-5a]

Figure 4:20 - Step 3.1: Capture Operational Concepts

Known ***responsibility*** designations of ***organizations*** or ***personnel*** are assigned. The set of ***operational agents***, scoped by participating ***concept items***, are then structured into logical relationships where they are grouped or made parts of each other. A review is done to ensure all ***capabilities*** and their ***conditions*** and contexts have been covered or addressed within the ***high-level operational concepts***.

- **High-Level Operational Concept** – describes the Resources and Locations required to meet an operational scenario from an integrated systems point of view. It is used to communicate overall quantitative and qualitative system characteristics to stakeholders. (Note: this is an element containing an integrated view of an operational scenario of participants, stakeholders, conditions, resources, and their conceptual roles with each other. It can describe the concept roles for Resources and Locations required to meet an operational scenario from an integrated systems point of view. It is used to communicate overall quantitative and qualitative system characteristics to stakeholders.)
- **Concept Item** – abstract, an item which may feature in a **High-Level Operational Concept** (Note: this is an element representing the part played by a logical or physical performer, asset, or condition, which guides the accounting of necessary operational agents in an operational architecture)
- **Known Resource** – asserts that a known **Resource Performer** constrains the implementation of the Operational Performer that plays the role in the **Operational Architecture**. (Note: this is typically a pre-

existing entity, such as a physical resource or other operational agent which participates in an operational scenario, and is already known and described outside the context of the operational architecture)

- **Condition** – type that defines the **Location**, **Environment**, and/or **Geopolitical Extent**
- **Operational Constraint** – a **Rule** governing an operational architecture element i.e., **Operational Performer**, **Operational Activity**, **Operational Information**, etc. (Note: as a type of rule, it may be enumerated by one of the following **Rule Kinds**: structural assertion, action assertion, derivation, contract, constraint, guidance, security policy, or caveat.)

4.3.2 Operational Activity Behavior

Step 3.2 – Capture Operational Behaviors – One or more activities are mapped to each ***capability***, corresponding to, and covering all the operational concepts, as the basis for assembling a complete description of ***operational activities***. This description may be arranged by activity groupings, operational agents capable to perform them, or some other useful organization.

- **Operational Activity** – an activity that captures a logical process, specified independently of how the process is carried out (Note: an activity may contain a view of a logical process flow)
- **Standard Operational Activity** – a sub-type of **Operational Activity** that is a standard operating procedure. (Note: this may be an operational activity which dictates or meets some standard tactic, technique, or procedure for the enterprise)

When developing a business architecture, the personnel behavior functions and sequenced timelines, and their implementation of operational activities and service functions, are used to capture the business aspects of the enterprise. Business Process Model and Notation (BPMN)[4] is an alternative diagram for these views in the UAFML.

Process flow diagrams are constructed for all ***operational activities*** including ***operational activity actions*** classified by ***operational activities*** and process control-flow mechanisms such as decision nodes and forks. The ***operational activity actions*** are grouped into swim-lanes to create ***operational agents***. When operational agents are already known, they are assigned to swim-lanes and the actions they are capable to perform.[5]

All of the ***operational agents*** are associated with each other, in an operational connectivity diagram, wherever ***operational exchanges*** exist between them. ***Operational exchanges*** and their ***operational exchange items*** are placed on all associations. Multiple associations may exist between operational agents to group different individual or multiple kinds of exchanges in terms of time, sequence, interface definition, or kinds of items that are exchanged.

- **Operational Exchange** – asserts that a flow can exist between **Operational Performers** (i.e., flows of information, people, material, or energy)
- **Operational Exchange item** – an abstract grouping for elements that defines the types of elements that can be exchanged between **Operational Performers** and conveyed by an **Operational Exchange** (Note: An **Operational Exchange Item** may be **Operational Information**, an **Operational Signal**, a resource performer, or a **Geopolitical Extent Type**)

[4] ***ISO 19510:2013 – Information technology — Object Management Group Business Process Model and Notation*** establishes a set of notations and semantics for collaboration, process, and choreography diagrams to communicate process information to other business users, process implementers, customers, and suppliers (see also OMG standard BPMN 2.0.2)

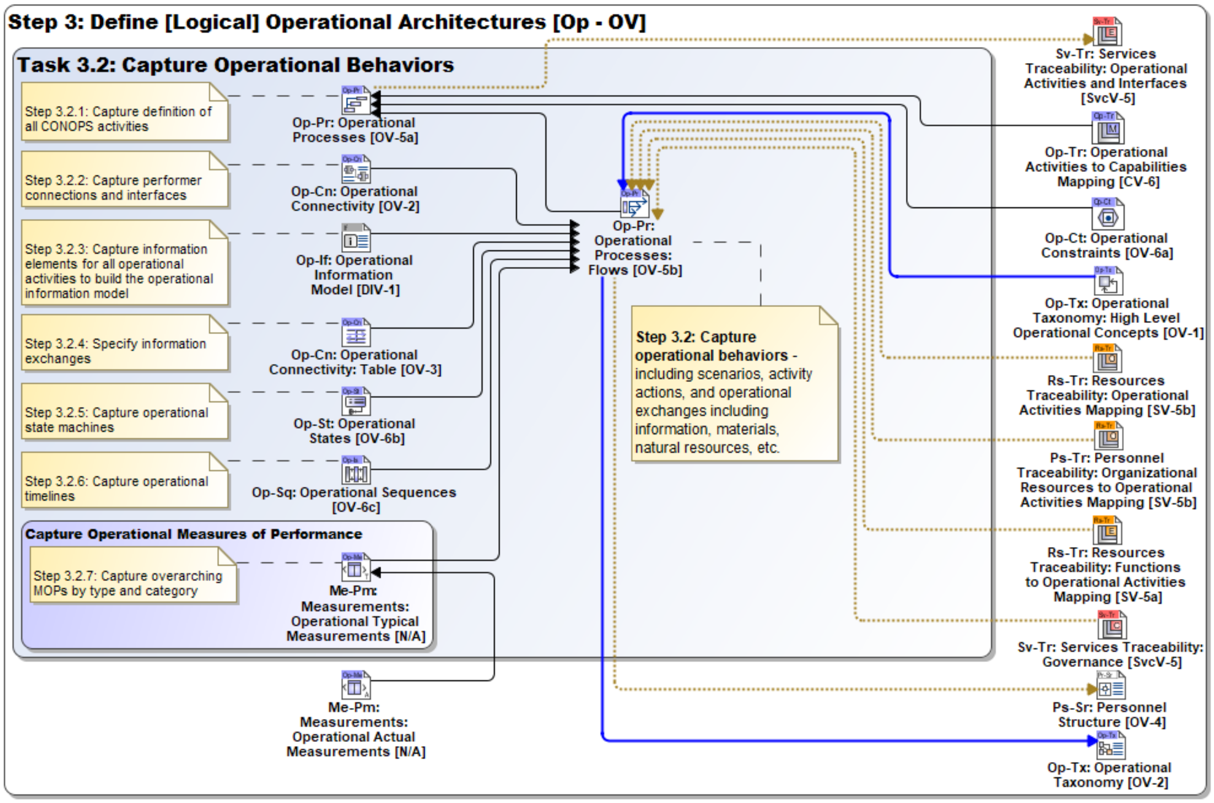
[5] This guide presents a sequence reflecting traditional functional analysis in systems engineering, where processes (activities or functions) are defined first, then grouped and bundled into swim lanes with control flows, which are then assigned to performers. In UAFML exchange flows cannot be created between processes which have not yet had a performer assigned. This constraint with the language may cause architects to design and create performers prematurely, leading to inefficient groupings and allocations of processes to performers. Once an architecture proceeds to implementation, this order may change if implementation starts with performers that have been designed in the operational Viewpoint, or it may not change if implementation starts with functions that implement activities.

An operational information ***conceptual information model*** (CIM) is created to define the ***operational information*** elements which are exchanged. The CIM may include taxonomies, structure, associations, and conveyance of the operational information elements.

- **Operational Information** – an item of information that flows between **Operational Performers** and is produced and consumed by the **Operational Activities** that the **Operational Performers** are capable to perform (see **Is Capable To Perform**)

The process flows are refined, when needed, with ***operational state descriptions*** for performers and sequenced timelines of ***operational messages*** and ***operational methods***. ***Operational interfaces*** may be defined and declared for interface points of any operational agent. ***Operational connections*** are added to the process flow to realize the ***operational exchange items*** (which for the operational architecture may consist of ***operational information elements, resource performers, operational signals***, and ***geopolitical extents***). State descriptions and sequenced timelines may now be developed to supplement or refine the item exchanges.

- **Operational Message** – a message for use in an **Operational Interaction Scenario** which carries any of the subtypes of **Operational Exchange** (Note: it is a sequenced message between two operational agents which may convey **Operational Exchanges** or **Operational Methods**)
- **Operational Method** – a behavioral feature of an Operational Agent whose behavior is specified in an **Operational Activity**
- **Operational Interface** – a declaration that specifies a contract between the **Operational Performer** it is related to, and any other **Operational Performers** it can interact with



Step 3: Define [Logical] Operational Architectures [Op - OV]
Task 3.2: Capture Operational Behaviors
Step 3.2.1: Capture definition of all CONOPS activities
Op-Pr: Operational Processes [OV-5a]
Step 3.2.2: Capture performer connections and interfaces
Op-Cn: Operational Connectivity [OV-2]
Step 3.2.3: Capture information elements for all operational activities to build the operational information model
Op-If: Operational Information Model [DIV-1]
Step 3.2.4: Specify information exchanges
Op-Cn: Operational Connectivity: Table [OV-3]
Step 3.2.5: Capture operational state machines
Op-St: Operational States [OV-6b]
Step 3.2.6: Capture operational timelines
Op-Sq: Operational Sequences [OV-6c]
Capture Operational Measures of Performance
Step 3.2.7: Capture overarching MOPs by type and category
Me-Pm: Measurements: Operational Typical Measurements [N/A]
Me-Pm: Measurements: Operational Actual Measurements [N/A]
Op-Pr: Operational Processes: Flows [OV-5b]
Step 3.2: Capture operational behaviors - including scenarios, activity actions, and operational exchanges including information, materials, natural resources, etc.
Sv-Tr: Services Traceability: Operational Activities and Interfaces [SvcV-5]
Op-Tr: Operational Activities to Capabilities Mapping [CV-6]
Op-Ct: Operational Constraints [OV-6a]
Op-Tx: Operational Taxonomy: High Level Operational Concepts [OV-1]
Rs-Tr: Resources Traceability: Operational Activities Mapping [SV-5b]
Ps-Tr: Personnel Traceability: Organizational Resources to Operational Activities Mapping [SV-5b]
Rs-Tr: Resources Traceability: Functions to Operational Activities Mapping [SV-5a]
Sv-Tr: Services Traceability: Governance [SvcV-5]
Ps-Sr: Personnel Structure [OV-4]
Op-Tx: Operational Taxonomy [OV-2]

Figure 4:21 - Step 3.2: Capture Operational Behaviors

Typical ***measures*** of performance are defined for the ***performers*** and their ***activities***, from which ***actual measurements*** can be made and taken. ***Measures of performance*** (MOPs) may include parametric diagrams when needed. Measures of overall activity and performers should demonstrate satisfaction of ***capability measures of effectiveness*** (MOEs) either directly or indirectly through examination or correlation of ***operational activities*** that map to a ***capability***.

A review is done to ensure all ***operational concepts*** have been covered by ***operational activities***, and that all ***concept items*** have now been covered by use of their ***performers*** that classify their ***concept items***.

4.3.3 Operational Taxonomy and Structure

Step 3.3 – Capture Operational Taxonomy and Structure – All ***operational agents*** identified by their roles as ***concept items*** identified in Step 3.1 are accounted for in a ***taxonomy***. This view captures understanding of ***generalizations***, particularly when a higher-level reference enterprise architecture is setting contextual guidance, or when specialization of operational agents is necessary. Internal structures of ***operational agents*** within the taxonomy are developed when ***operational connectors***, internal features and characteristics of that agent must be known, including ***operational roles*** they have as part of a greater ***operational agent***. One operational agent may perform different operational roles, and one operational role may be performed by different operational agents.

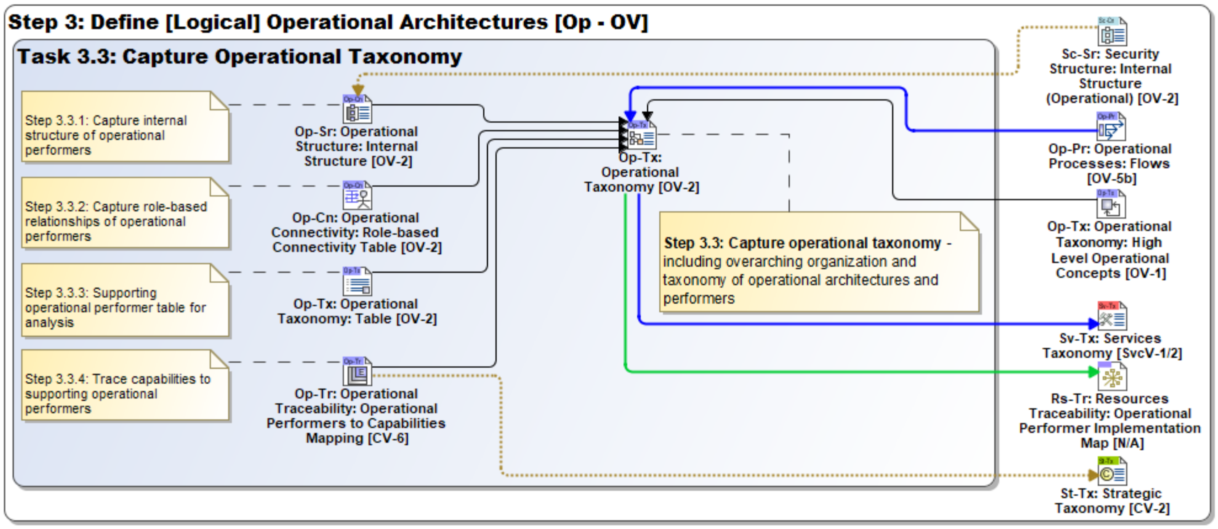

Step 3: Define [Logical] Operational Architectures [Op - OV]
Task 3.3: Capture Operational Taxonomy
Step 3.3.1: Capture internal structure of operational performers
Step 3.3.2: Capture role-based relationships of operational performers
Step 3.3.3: Supporting operational performer table for analysis
Step 3.3.4: Trace capabilities to supporting operational performers
Op-Sr
Op-Sr: Operational Structure: Internal Structure [OV-2]
Op-Cn
Op-Cn: Operational Connectivity: Role-based Connectivity Table [OV-2]
Op-Tx
Op-Tx: Operational Taxonomy: Table [OV-2]
Op-Tr
Op-Tr: Operational Traceability: Operational Performers to Capabilities Mapping [CV-6]
Op-Tx
Op-Tx: Operational Taxonomy [OV-2]
Step 3.3: Capture operational taxonomy - including overarching organization and taxonomy of operational architectures and performers
Sc-Cr
Sc-Sr: Security Structure: Internal Structure (Operational) [OV-2]
Op-Pr
Op-Pr: Operational Processes: Flows [OV-5b]
Op-Tx
Op-Tx: Operational Taxonomy: High Level Operational Concepts [OV-1]
Sv-Tx
Sv-Tx: Services Taxonomy [SvcV-1/2]
Rs-Tr: Resources Traceability: Operational Performer Implementation Map [N/A]
St-Tx
St-Tx: Strategic Taxonomy [CV-2]

Figure 4:22 - Step 3.3: Capture Operational Taxonomy and Structure

Examination of the taxonomy is necessary to scope and perform assessment of the trade space and an analysis of alternatives, including differences between composition and aggregation in structure. A tracing of what capabilities are exhibited by ***operational agents*** is then automatically generated and examined based on the mapping of those ***activities*** to ***capabilities***, to identify orphaned ***operational agents*** or conflicts in overall ***operational activity*** mappings to ***capabilities***. Environmental conditions from Step 3.1 are associated with these exhibited relationships between the ***operational agents*** and the ***capabilities***.

- **Operational Role** – Usage of an **Operational Performer** or **Operational Architecture** in the context of another **Operational Performer** or **Operational Architecture**. Creates a whole-part relationship. (Note: this element represents the part played by one operational agent in another. In particular, an **Operational Role** may be captured as a **Problem Domain** within an **Operational Architecture**)
- **Operational Connector** – a Connector that goes between **Operational Roles** representing a need to exchange Resources. It can carry a number of **Operational Exchanges**
- **Exhibits** – a tuple that exists between a Capable Element and a **Capability** that it meets under specific environmental conditions. (Note: in this case it is the relationship of an **Operational Agent** to a **Capability** implied by its mapped **Operational Activity** and which may be associated with an environmental condition)

4.3.4 Operational Analysis and Roles

Step 3.4 – Analyze Operational Structure – Once an operational architecture has been captured and defined, analysis to prepare implementation possibilities begins. ***Resource implementations*** are examined for both

operational agents and their ***operational roles*** to aid in implementation decisions, and analysis of alternatives. ***Operational activities*** and their inherent structure and composition are examined for potential ***resource implementations***.

- **Implements** – a tuple that defines how an element in the upper layer of abstraction is implemented by a semantically equivalent element (for example tracing the **Functions** to the **Operational Activities**) in the lower level of abstraction. (Note: often this is a relationship between resource elements that outfit or equip operational elements)

Resource performers (defined in Step 5) and the functions performed by those performers will "implement" the operational agents and operational activities defined in the operational architecture. Likewise, resource roles, connectors, resource exchange items, resource exchanges, and resource interfaces will implement the analogous things in the operational architecture.

Operational agents, their ***operational roles***, and their ***actions*** and ***operational activities*** are refined as necessary to adjust for tradeoff decisions, efficiencies, feasibility, and other factors in the operational architecture design, as well as in response to resource implementation decisions based on cost, feasibility and other factors.

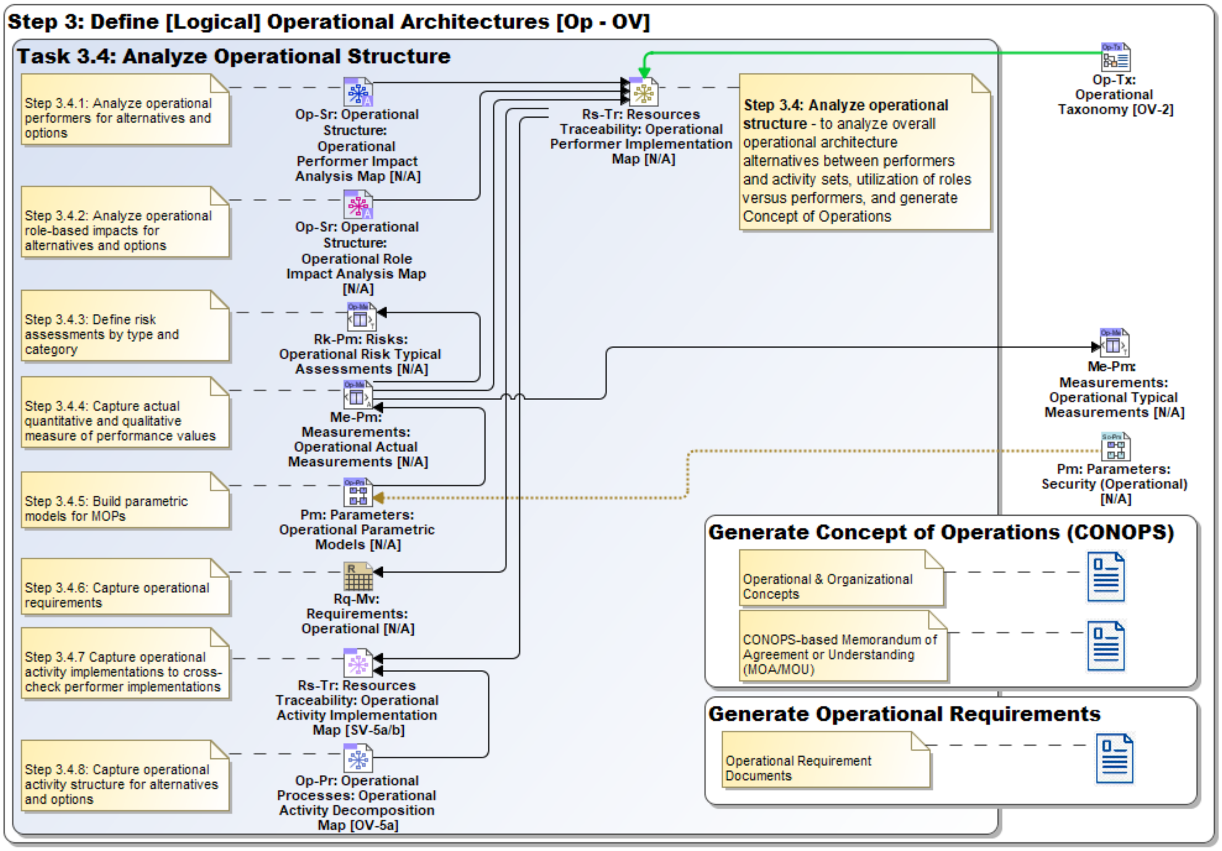

Step 3: Define [Logical] Operational Architectures [Op - OV]
Task 3.4: Analyze Operational Structure
Step 3.4.1: Analyze operational performers for alternatives and options
Op-Sr: Operational Structure: Operational Performer Impact Analysis Map [N/A]
Step 3.4.2: Analyze operational role-based impacts for alternatives and options
Op-Sr: Operational Structure: Operational Role Impact Analysis Map [N/A]
Step 3.4.3: Define risk assessments by type and category
Rk-Pm: Risks: Operational Risk Typical Assessments [N/A]
Step 3.4.4: Capture actual quantitative and qualitative measure of performance values
Me-Pm: Measurements: Operational Actual Measurements [N/A]
Step 3.4.5: Build parametric models for MOPs
Pm: Parameters: Operational Parametric Models [N/A]
Step 3.4.6: Capture operational requirements
Rq-Mv: Requirements: Operational [N/A]
Step 3.4.7 Capture operational activity implementations to cross-check performer implementations
Rs-Tr: Resources Traceability: Operational Activity Implementation Map [SV-5a/b]
Step 3.4.8: Capture operational activity structure for alternatives and options
Op-Pr: Operational Processes: Operational Activity Decomposition Map [OV-5a]
Rs-Tr: Resources Traceability: Operational Performer Implementation Map [N/A]
Step 3.4: Analyze operational structure - to analyze overall operational architecture alternatives between performers and activity sets, utilization of roles versus performers, and generate Concept of Operations
Op-Tx: Operational Taxonomy [OV-2]
Me-Pm: Measurements: Operational Typical Measurements [N/A]
Pm: Parameters: Security (Operational) [N/A]
Generate Concept of Operations (CONOPS)
Operational & Organizational Concepts
CONOPS-based Memorandum of Agreement or Understanding (MOA/MOU)
Generate Operational Requirements
Operational Requirement Documents

Figure 4:23 - Step 3.4: Analyze Operational Structure

The primary exit criterion is the completion of an Operational Architecture whose ***activities*** and ***performers*** cover all ***capabilities*** and a determination that all ***capability measures of effect*** (either directly, numerically, or by analysis) can be met.

An entire Operational Architecture, with all of its elements (agents, behaviors, etc.) may be considered as one option in an analysis of alternatives within the model. In other words, each alternative to be examined can be represented by a separate Operational Architecture. Selection of an Operational Architecture may be based on a set of assessment criteria for satisfying the capability needs defined in the Strategic Architecture.

Typical measurements of performance (MOPs) from Step 3.2 may be used to generate ***actual measurements*** to either capture analysis of these alternatives or records of analysis for ***performers*** over time. If the operational architecture continues as a reference architecture over time, ***actual measurements*** may form a life cycle record of that Operational Architecture as it evolves.

Portions of the Operational Architecture which serve only to guide a resource implementation one time may not need more than one record of ***actual measurements***, which may act to inform requirements in lieu of moving on to designing resources and services architectures, or they may act to inform a resource architecture design that is then used as a reference for evaluating bids and resource solution architectures from resource developers.

A ***risk*** analysis assesses the impact of events that may ***affect*** operational assets. Measures and ***actual measurements*** of these ***risks*** are included with the operational measurement analysis.

When needed, a Concept of Operations (CONOPS) document can be developed based on the models and views generated during this Step 3. Ideally this CONOPS document is automatically generated from the model itself using reporting scripts and model queries. Sometimes it also necessary to develop a Concept of Use (CONUSE) and a Concept of Employment (CONEMP) to describe how the fielded capabilities (in the form of ***capability configurations***) will be used to conduct operations and how they will be employed to achieve mission or business goals and imperatives.

Additionally, a set of operational ***requirements*** can be generated on the basis of the elements within the Operational Architecture, which are intended to be service, resource, personnel, and security implementation agnostic. Requirements tracing from the Operational Architecture can be satisfied by service, resource, personnel, and security solutions.

Requirements are developed and related to architectural elements relevant to the generation of the requirement. These are related using ***trace, verify, satisfy***, or ***refine*** relationships for linking to relevant elements in the architecture.

- **Requirement** – a statement that identifies a system, product, or process characteristic or constraint, which is unambiguous, clear, unique, consistent, stand-alone (not grouped), and verifiable, and is deemed necessary for stakeholder acceptability (INCOSE 2010)
- **Refine** – a relationship from an architectural element which refines a text-based requirement

4.3.5 Architecture View Summary for Step 3

The view specifications in UAF for this viewpoint are outlined here:

	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Operational Op	Requirements Rq-Mv	Operational Taxonomy Op-Tx	Operational Structure Op-Sr	Operational Connectivity Op-Cn	Operational Processes Op-Pr	Operational States Op-St	Operational Sequences Op-Sq	Operational Information Model (Op-If)	Envmt, Msmts, Risks (En-Pm, Me-Pm, Rk-Pm)	Operational Constraints Op-Ct	-	Operational Traceability Op-Tr

A summary of the lower-level steps and UAF views is shown below. The architecture views produced by each step are identified by the UAF grid designator Aa-Bb where Aa represents the Viewpoint row in the grid and Bb represents the Aspect column in the grid. DoDAF view designations are also shown in [brackets], when applicable. The name of the architecture view matches the name of the view specification in the UAFML. In some cases, a subtitle is provided to identify what kind of information is provided in an instance of that UAF view. There will be some architecture views that are not part of the UAF specification since you will sometimes need fit-for-purpose (i.e., custom) views that capture the necessary architectural information.

68	**Step 3: Define [Logical] Operational Architectures [Op - OV]**	**Views**
69	**Step 3.1: Capture operational concepts** - including concept roles, situations, and scenarios in context of operational environments and identify the constraints of operations	Op-Tx: Operational Taxonomy: *High Level Operational Concepts* [OV-1]
70	Step 3.1.1: Capture simple operational sketches with users describing all key CONOPS ideas	Op-Tx: Operational Taxonomy: *Concept of Operations Sketch* [OV-1]
71	Step 3.1.2: Capture operational environments, regions, theaters, and operating conditions	En-Pm: Environment: *Operational* [N/A]
72	Step 3.1.3: Capture overarching operational architecture performers, roles, and structural relationships	Op-Sr: Operational Structure [OV-2]
73	Step 3.1.4: Capture operational rules of engagement, methods, and operational policies in rule form	Op-Ct: Operational Constraints [OV-6a]
74	Step 3.1.5: Capture the environment and conditional constraints for operations (e.g., operational areas, planning scenarios, threats, locations, etc.)	Op-Ct: Operational Constraints: *Definition* [OV-6a]
75	Step 3.1.6: Capture the organizations involved in the overall CONOPS	Ps-Tx: Personnel Taxonomy: *Organizational Context* [OV-4]
76	Step 3.1.7: Capture the responsibilities of the organizations involved in the CONOPS relative to their roles	Ps-Sr: Personnel Structure: *Organizational Responsibilities* [OV-4]
77	**Step 3.2: Capture operational behaviors** - including scenarios, activity actions, and operational exchanges including information, materials, natural resources, etc.	Op-Pr: Operational Processes: *Flows* [OV-5b]
78	Step 3.2.1: Capture definition of all CONOPS activities	Op-Pr: Operational Processes [OV-5a]
79	Step 3.2.2: Capture performer connections and interfaces	Op-Cn: Operational Connectivity [OV-2]
80	Step 3.2.3: Capture information elements for all operational activities to build the operational information model	Op-If: Operational Information Model [DIV-1]
81	Step 3.2.4: Specify information exchanges	Op-Cn: Operational Connectivity: *Table* [OV-3]
82	Step 3.2.5: Capture operational state machines	Op-St: Operational States [OV-6b]
83	Step 3.2.6: Capture operational timelines	Op-Sq: Operational Sequences [OV-6c]
84	Step 3.2.7: Capture typical MOPs by type and category	Me-Pm: Measurements: *Operational Typical Measurements* [N/A]
85	**Step 3.3: Capture operational taxonomy** - including overarching organization and taxonomy of operational architectures and performers	Op-Tx: Operational Taxonomy [OV-2]
86	Step 3.3.1: Capture internal structure of operational performers	Op-Sr: Operational Structure: *Internal Structure* [OV-2]
87	Step 3.3.2: Capture role-based relationships of operational performers	Op-Cn: Operational Connectivity: *Role-based Connectivity Table* [OV-3]
88	Step 3.3.3: Supporting operational performer table for analysis	Op-Tx: Operational Taxonomy: *Table* [OV-2]
89	Step 3.3.4: Trace capabilities to supporting operational performers	Op-Tr: Operational Traceability: *Operational Performers to Capabilities Mapping* [CV-6]
90	**Step 3.4: Analyze operational structure** - to analyze overall operational architecture alternatives between performers and activity sets, utilization of roles versus performers, and generate Concept of Operations	Rs-Tr: Resources Traceability: *Operational Performer Implementation Map* [N/A]

91	Step 3.4.1: Analyze operational performers for alternatives and options	Op-Sr: Operational Structure: *Operational Performer Impact Analysis Map* [N/A]
92	Step 3.4.2: Analyze operational role-based impacts for alternatives and options	Op-Sr: Operational Structure: *Operational Role Impact Analysis Map* [N/A]
93	Step 3.4.3: Define risk assessments by type and category	Rk-Pm: Risks: *Operational Risk Typical Assessments* [N/A]
94	Step 3.4.4: Capture actual quantitative and qualitative measure of performance values	Me-Pm: Measurements: *Operational Actual Measurements* [N/A]
95	Step 3.4.5: Build parametric models for MOPs	Pm: Parameters: *Operational Parametric Models* [N/A]
96	Step 3.4.6: Capture operational requirements	Rq-Mv: Requirements: *Operational* [N/A]
97	Step 3.4.7 Capture operational activity implementations to cross-check performer implementations	Rs-Tr: Resources Traceability: *Operational Activity Implementation* Map [SV-5a/b]
98	Step 3.4.8: Capture operational activity structure for alternatives and options	Op-Pr: Operational Processes: *Operational Activity Decomposition Map* [OV-5a]

STEP 4 – Service Architectures

4.4 Step 4 – Service Architectures

Purpose. The purpose of this step is to define services and to specify required and provided service levels for the services needed to exhibit capabilities and to support operational activities. Key stakeholders for this step are Enterprise Architects, Solution Providers, Systems Engineers, Software Architects and Business Architects. Their concerns are mainly about what are the specifications of services required to exhibit capabilities.

Conceptual Schema. The key concepts in the Services Viewpoint that can be used as model elements in the architecture views and the relationships between these concepts are illustrated in the conceptual schema shown in Figure 4:24. These ***key concepts*** are highlighted in italics within the Narrative and some of the less obvious concepts are listed with the associated UAF meaning of that concept. Detailed definition of the entities and relationships shown in the conceptual schema are provided in the UAFML specification document.

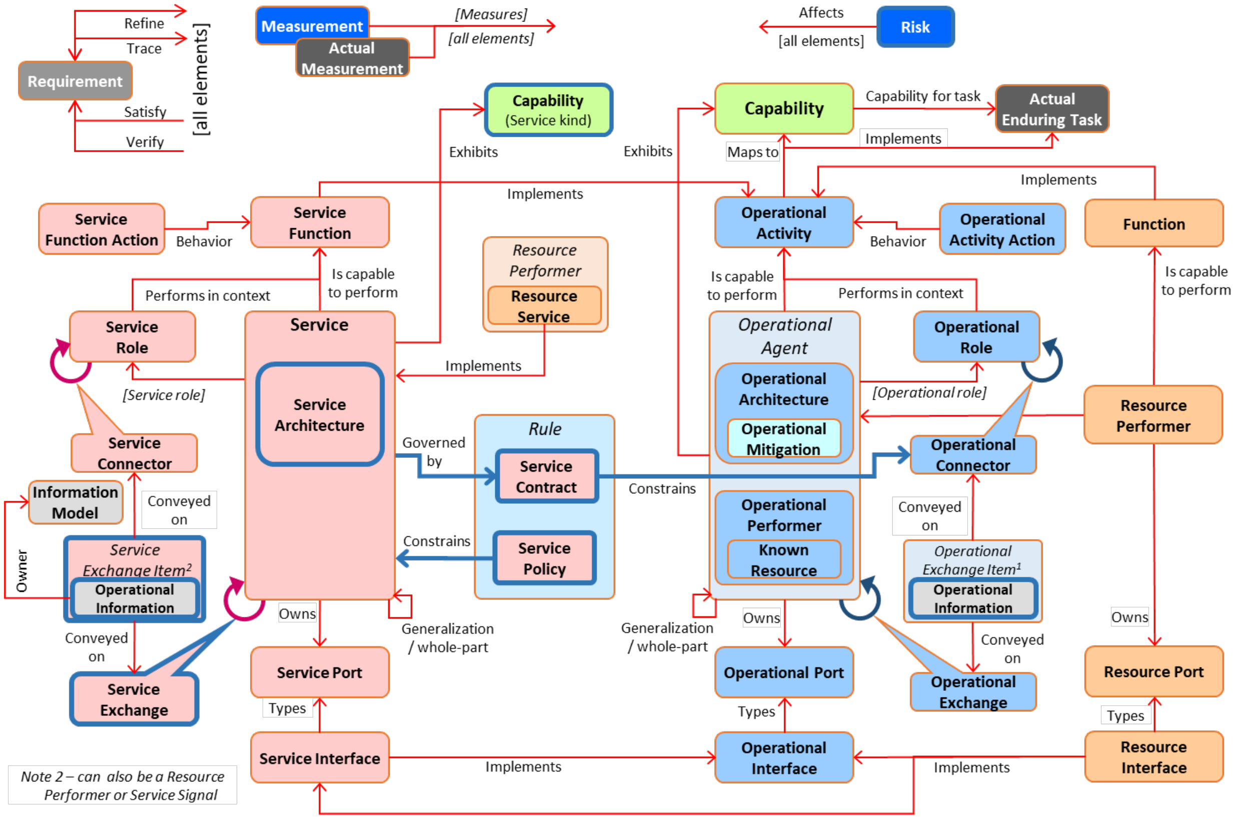


Figure 4:24 - Conceptual Schema for Service Architectures

The service architecture concepts are shown on the left in red and the operational and resource architecture concepts are shown on the right in blue and orange, respectively. ***Resource interfaces*** will "implement" the ***service interfaces*** and ***operational agents*** will exchange information or resources which ***constrain*** various ***service contracts*** that ***govern*** underlying ***services***. These ***services*** are also ***constrained*** by ***service policies***.

Service architecture service roles, service connectors, service exchange items, service exchanges, service ports, and service architectures will comprise the services used by an Operational Architecture. Service ports are typed by a service interface.

This Guide is intended to be used in conjunction with the UAF Sample Problem that defines architecture views for a Search and Rescue Mission. UAF Specification documents (including the Sample Problem) can be downloaded from the OMG webpage: www.omg.org/spec/UAF/About-UAF/.

Step 4.0 – Define Service Architectures – The main entry criterion for this Step is bringing forward the ***operational activities*** from Step 3 with associated ***performers***, ***operational exchanges***, ***measures*** and their operational context. These are used to define potential or actual opportunities for service use, i.e., identification of services to be used by ***operational activities*** that support operational needs for information or resources that can be provided on the basis of a ***service*** specification**.** These services can also be used as the components of a designated ***service architecture***.

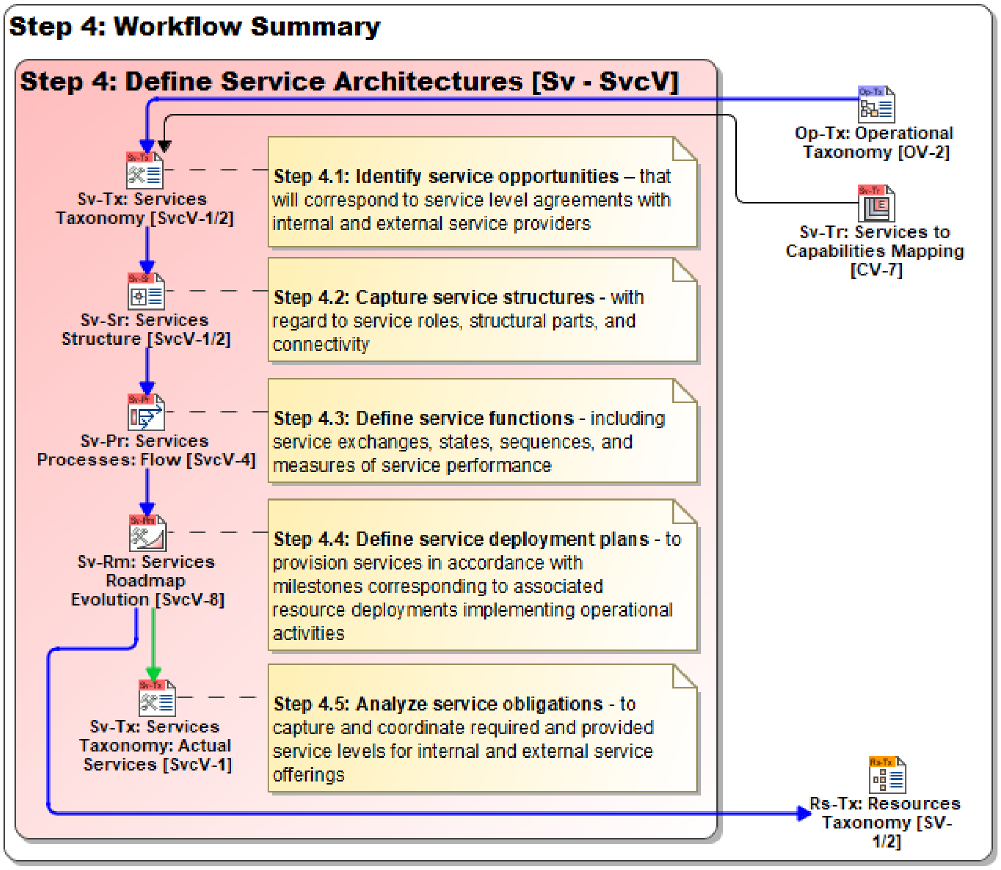
- **Service** – the specification of a set of functionalities provided by one element for the use of others (Note: this is often a set of functionalities that can be provided for the use of operational activities)
- **Service Architecture** – an element used to denote a model of the Architecture, described from the **Services** perspective (Note: this may include a composition or aggregation of services that is described from the service perspective)

Workflow Summary. A summary level view of the five steps involved in Step 4 workflow is shown below. The flowlines represent how one architecture view "influences" another architecture view. These lines do not represent a particular sequence of process activity execution or imply information exchanges. These should be thought of as influence diagrams rather than process diagrams. These architecture views (e.g., views, diagrams, tables) can be mapped to the architecture processes used in your organization and can be incorporated into an architecture modeling methodology.

The ***service architecture*** defines service *behavior* elements (e.g., processes, states, sequences) and allocates these to service *structure* elements (e.g., other service architectures and services to be used in the service setting, roles, connectors, exchange items, exchanges, ports and interfaces). The behavior elements define service functions for the ***services*** that are provisioned and ***governed by*** various ***service contracts*** which will ***constrain*** connections in ***operational exchanges***.

- **Service Contract** – a constraint governing the use of one or more **Services** (Note: this is often a declaration of a contractual governance relationship for the provision of a **Service** conveying **Operational Information, Service Signals** or resources for an **Operational Activity and between operational performers**)
- **Governed By** – a tuple that exists between the **Service Contract** and the **Service** that it governs (Note: this is the governance of a **Service** by a **Service Contract** which constrains an **Operational Connector** within one or more **Operational Exchanges**)

The ***service architecture*** can be defined from several perspectives. The "black box" perspective is where the service is being viewed in terms of its inputs and outputs, without knowledge of its internal workings. The "white box" perspective is the opposite where the inner components or logic are available for inspection, or if necessary, these inner elements are defined by the **service architecture** model. A "gray box" perspective is somewhere in between where there is partial consideration of the inner structures and behaviors. Usually if the services to be used are external to the organizations involved then black box models are sufficient. But if the services are internal to the organizations involved then sometimes the architecture needs to have some influence on the inner workings of the service to ensure a balanced and robust solution will be realized across the enterprise as a whole.


Step 4: Workflow Summary
Step 4: Define Service Architectures [Sv - SvcV]
Sv-Tx: Services Taxonomy [SvcV-1/2]
Sv-Sr: Services Structure [SvcV-1/2]
Sv-Pr: Services Processes: Flow [SvcV-4]
Sv-Rm: Services Roadmap Evolution [SvcV-8]
Sv-Tx: Services Taxonomy: Actual Services [SvcV-1]
Op-Tx: Operational Taxonomy [OV-2]
Sv-Tr: Services to Capabilities Mapping [CV-7]
Rs-Tx: Resources Taxonomy [SV-1/2]

Step 4.1: Identify service opportunities – that will correspond to service level agreements with internal and external service providers

Step 4.2: Capture service structures - with regard to service roles, structural parts, and connectivity

Step 4.3: Define service functions - including service exchanges, states, sequences, and measures of service performance

Step 4.4: Define service deployment plans - to provision services in accordance with milestones corresponding to associated resource deployments implementing operational activities

Step 4.5: Analyze service obligations - to capture and coordinate required and provided service levels for internal and external service offerings

Figure 4:25 - Workflow Summary for Step 4: Define Service Architectures

The second-level steps in the Step 4 workflow are listed below along with the UAF views (and corresponding [DoDAF views] where applicable) associated with those steps. A complete list of the detailed steps at the third level with their corresponding views is provided at the end of this section.

99	Step 4: Define Service Architectures [Sv - SvcV]	Views
100	**Step 4.1: Identify service opportunities** – that will correspond to service level agreements with internal and external service providers	Sv-Tx: Services Taxonomy [SvcV-1]
107	**Step 4.2: Capture service structures** - with regard to service roles, structural parts, and connectivity	Sv-Sr: Services Structure [SvcV-1/2]
111	**Step 4.3: Define service functions** - including service exchanges, states, sequences, and measures of service performance	Sv-Pr: Services Processes: *Flow* [SvcV-4]
118	**Step 4.4: Define service deployment plans** - to provision services in accordance with milestones corresponding to associated resource deployments implementing operational activities	Sv-Rm: Services Roadmap-Evolution [ScvV-8]
120	**Step 4.5: Analyze service obligations** - to capture and coordinate service level agreements and requirements for internal and external service providers and consumers	Sv-Tx: Services Taxonomy: *Actual Services* [SvcV-1]

4.4.1 Service Opportunities

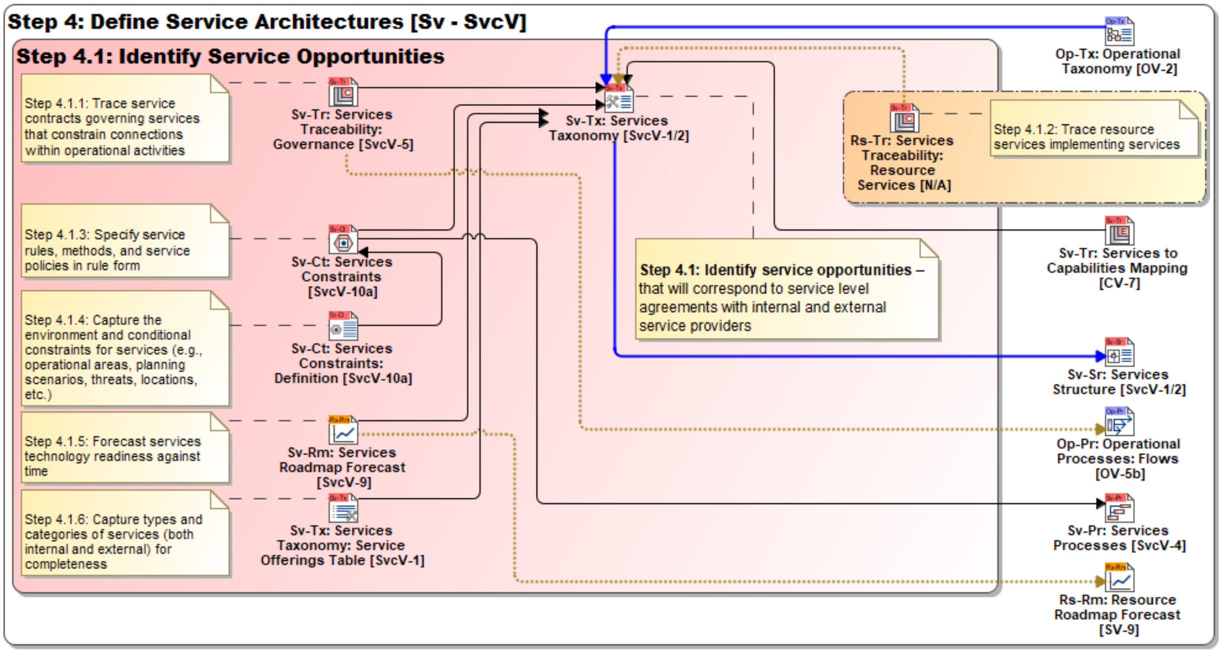
Step 4.1 – Identify Service Opportunities – A set of ***services*** are described, which have been identified from Step 2 and Step 3, where services are needed or anticipated that will contribute to or ***exhibit*** capabilities, as they convey information, signals and resources on connections consumed or provided by ***operational activities*** that map to those ***capabilities***. Additionally, all ***services*** known to be implemented by ***resource services*** are identified. These ***services***

are accounted for in a ***taxonomy***. This captures the understanding of ***generalizations***, particularly when a reference service architecture is setting contextual guidance, or service offerings are already known.

Rules, and other ***service policies*** are identified and applied to ***services***, and then later expanded to their associated actions and exchanges. As necessary, a service parametric model can be used to evaluate or provide ***service policies*** which constrain the services. When resource technologies in Step 5.1 are used by services that are planned for future availability, a ***forecast***, is assembled to show when key resources utilized by ***services*** are available for periods applicable to usage demands for the services.

A review is done to ensure all ***capabilities*** and their mapped ***operational activities*** and contexts have been covered or addressed, when appropriate, within a ***services taxonomy*** table.

- **Service Policy** – a constraint governing the use of one or more **Services** (Note: this is a type of rule or constraint stemming from a guidance, contract, or other source)
- **Exhibits** – a tuple that exists between a Capable Element and a Capability that it meets under specific environmental conditions (Note: in this case this may be a relationship of a **Service** to a **Capability** (service kind) and which may be associated with an environmental condition)



Step 4: Define Service Architectures [Sv - SvcV]
Step 4.1: Identify Service Opportunities
Step 4.1.1: Trace service contracts governing services that constrain connections within operational activities
Sv-Tr: Services Traceability: Governance [SvcV-5]
Sv-Tx: Services Taxonomy [SvcV-1/2]
Rs-Tr: Services Traceability: Resource Services [N/A]
Step 4.1.2: Trace resource services implementing services
Op-Tx: Operational Taxonomy [OV-2]
Step 4.1.3: Specify service rules, methods, and service policies in rule form
Sv-Ct: Services Constraints [SvcV-10a]
Step 4.1: Identify service opportunities – that will correspond to service level agreements with internal and external service providers
Sv-Tr: Services to Capabilities Mapping [CV-7]
Step 4.1.4: Capture the environment and conditional constraints for services (e.g., operational areas, planning scenarios, threats, locations, etc.)
Sv-Ct: Services Constraints: Definition [SvcV-10a]
Sv-Sr: Services Structure [SvcV-1/2]
Op-Pr: Operational Processes: Flows [OV-5b]
Step 4.1.5: Forecast services technology readiness against time
Sv-Rm: Services Roadmap Forecast [SvcV-9]
Sv-Pr: Services Processes [SvcV-4]
Step 4.1.6: Capture types and categories of services (both internal and external) for completeness
Sv-Tx: Services Taxonomy: Service Offerings Table [SvcV-1]
Rs-Rm: Resource Roadmap Forecast [SV-9]

Figure 4:26 - Step 4.1: Identify Service Opportunities

4.4.2 Service Structures

Step 4.2 – Capture Service Structures – Internal structure of ***services*** within the taxonomy are developed when internal features and characteristics of those services must be known, including ***service roles*** they have as part of a greater ***service architecture***. One service may perform different service roles, and one service role may be performed by different services. Examination of this is necessary to scope and perform assessment of the trade space and conduct an analysis of alternatives, including differences between composition and aggregation in the service structure. Formal ***service interfaces*** may be defined and declared for interface points of any service, which are implemented by resource interfaces, or which can implement operational interfaces.

- **Service Role** – a behavioral feature of a Service whose behavior is specified in a **Service Function**
- **Service Interface** – a contract that defines the **Service Methods** and **Service Signals** that the **Service** realizes (Note: this may be for a formal declaration of a contractual level interface with a **Service**)
- **Implements** – a tuple that defines how an element in the upper layer of abstraction is implemented by a semantically equivalent element (for example tracing the **Functions** to the **Operational Activities**) in the

lower level of abstraction (Note: in this case a relationship between a **Resource Interface** that will outfit or equip a **Service Interface**, or between a **Service Interface** or **Service Function** that will outfit or equip an **Operational Interface** or **Operational Activity**)

Services are associated with each other to identify service relationships which will support ***service exchanges***, ***service connectors***, ***service contracts***, and connections which may be required for service provisioning.

A services ***conceptual information model*** (CIM) is created to define the ***operational information*** elements which are exchanged. The CIM may include taxonomies, structure, associations, and conveyance of the operational information elements.

Service exchanges (new in UAF v1.2) are placed on all associations with groupings of ***service exchange items*** in accordance with logical actions or sequences of the service performers. (Note that a "service performer" means a Service or Service Architecture acting as an "agent" that performs some service functions for benefit of operational performers or other services. These service performers can handle service messages and have service methods.)

Multiple associations may exist between services which represent different types of exchanges in terms of time, sequence, interface definition, or kinds of items that are exchanged (which for the service architecture consist of ***operational information*** elements***, resource performers,*** and ***signals***).

- **Service Exchange** – asserts that a flow can exist between **Services** (i.e., flows of information, people, materiel, or energy)
- **Service Exchange Item** – an abstract grouping for elements that defines the types of elements that can be exchanged between **Services** and conveyed by a **Service Exchange** (Note: this may be one of several **Service Exchange Kinds** such as a Material Exchange, Organizational Exchange, Energy Exchange, Information Exchange, or a Configuration Exchange).

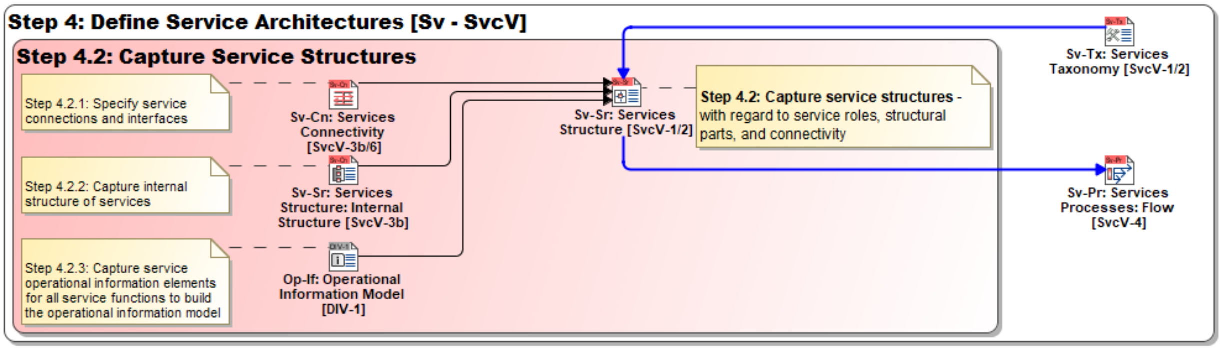

Step 4: Define Service Architectures [Sv - SvcV]
Step 4.2: Capture Service Structures
Step 4.2.1: Specify service connections and interfaces
Sv-Cn: Services Connectivity [SvcV-3b/6]
Step 4.2.2: Capture internal structure of services
Sv-Sr: Services Structure: Internal Structure [SvcV-3b]
Step 4.2.3: Capture service operational information elements for all service functions to build the operational information model
Op-If: Operational Information Model [DIV-1]
Sv-Sr: Services Structure [SvcV-1/2]
Step 4.2: Capture service structures - with regard to service roles, structural parts, and connectivity
Sv-Tx: Services Taxonomy [SvcV-1/2]
Sv-Pr: Services Processes: Flow [SvcV-4]

Figure 4:27 - Step 4.2: Capture Service Structures

4.4.3 Service Functional Behavior

Step 4.3 – Define Service Functions – Process flow diagrams are constructed for all ***service functions*** including ***service function actions*** classified by ***service functions*** and process control-flow mechanisms such as decision nodes and forks. When services are already known or defined in Step 3.1, they are assigned to swim-lanes and the actions they are capable to perform. Otherwise, ***service function actions*** are grouped into swim-lanes to create ***services*** which must be structured in Step 4.1 – 4.2.

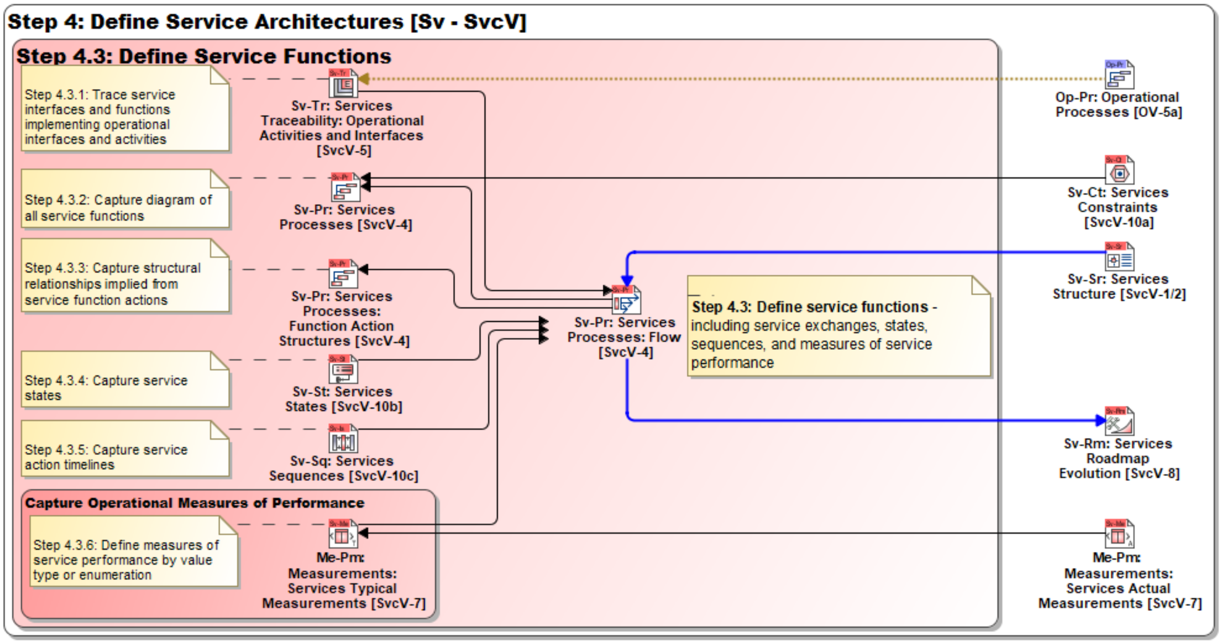
The process flows are refined, when needed, with ***service state descriptions*** for **services** and sequenced timelines of ***service messages*** and ***service methods***. ***Service connections*** are added to the process flow to realize the ***service exchange items*** (which for the service architecture may consist of ***operational information elements, resource performers,*** and ***service signals***). State descriptions and sequenced timelines may now be developed to supplement or refine the item exchanges.

- **Service Message** – a message for use in a Service Event-Trace (Note: this is a sequenced message between two services which may convey **Service Exchanges** or **Service Methods**)
- **Service Method** – a behavioral feature of a **Service** whose behavior is specified in a **Service Function**

Typical ***measures*** of performance are defined for the ***services*** and their ***functions***. Service ***measures of performance*** (MOPs) may include parametric diagrams when needed. Measures of overall functions and services should demonstrate satisfaction of agreed upon service levels either directly, or indirectly through examination or correlation of the MOPs.

- **Measurement** – a property of an element representing something in the physical world, expressed in amounts of a unit of measure

A review is done to ensure all service usage demands have been covered by ***services***, and ***service functions***. If service functions have been designated, one or more functions are mapped from each ***service***, corresponding to and covering all the ***service contracts***, to ensure a complete library of ***service functions***. This library may be arranged by service groupings, services capable to perform them, or some other useful organizational scheme.



Step 4: Define Service Architectures [Sv - SvcV]
Step 4.3: Define Service Functions
Step 4.3.1: Trace service interfaces and functions implementing operational interfaces and activities
Sv-Tr: Services Traceability: Operational Activities and Interfaces [SvcV-5]
Op-Pr: Operational Processes [OV-5a]
Step 4.3.2: Capture diagram of all service functions
Sv-Pr: Services Processes [SvcV-4]
Sv-Ct: Services Constraints [SvcV-10a]
Step 4.3.3: Capture structural relationships implied from service function actions
Sv-Pr: Services Processes: Function Action Structures [SvcV-4]
Sv-Sr: Services Structure [SvcV-1/2]
Sv-Pr: Services Processes: Flow [SvcV-4]
Step 4.3: Define service functions - including service exchanges, states, sequences, and measures of service performance
Step 4.3.4: Capture service states
Sv-St: Services States [SvcV-10b]
Step 4.3.5: Capture service action timelines
Sv-Sq: Services Sequences [SvcV-10c]
Sv-Rm: Services Roadmap Evolution [SvcV-8]
Capture Operational Measures of Performance
Step 4.3.6: Define measures of service performance by value type or enumeration
Me-Pm: Measurements: Services Typical Measurements [SvcV-7]
Me-Pm: Measurements: Services Actual Measurements [SvcV-7]

Figure 4:28 - Step 4.3: Define Service Functions

4.4.4 Service Deployment Plans

Step 4.4 – Define Service Deployment Plans – When services and service technologies are planned for future availability, a ***services roadmap*** is assembled to show when ***services*** will be ***released*** for employment, will be ***withdrawn***, or will change over time. This roadmap allows planners to understand when service offerings are deployed, in service, out of service, or no longer used. Actual milestones are created for services to denote their availability on a ***timeline***.

- **Version Released at Milestone** – an actual milestone category showing a version of an element to be released (Note: in this case a **Service**)
- **Version Withdrawn at Milestone** – an actual milestone category showing a version of an element to be withdrawn (Note: in this case a **Service**)

Service provisioning is controlled by ***version succession*** of ***versions of configurations*** within ***whole life configurations*** for services. Versions of configuration are structured within whole life configurations to organize deployments of service offerings. Various service offerings can ***succeed*** others as versions of the offerings are changed. The timing for employment of the services in ***actual enterprise phases*** is understood through the ***operational*** and ***resource architectures*** of those phases.

- **Whole Life Configuration** – a set of Versioned Elements (Note: in this case **Services** from a service provider)
- **Version of Configuration** – a property of a **Whole Life Configuration**, used in version control of a Versioned Element. It asserts that a Versioned Element is a version of a **Whole Life Configuration**. (Note: in this case a particular version of a **Service**)
- **Version Succession** – a tuple between two **Version Of Configurations** that denotes that one **Version Of Configuration** follows from another

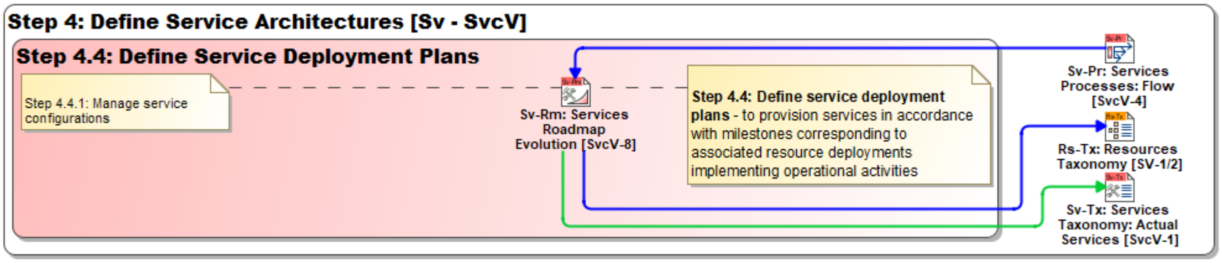

Step 4: Define Service Architectures [Sv - SvcV]
Step 4.4: Define Service Deployment Plans
Step 4.4.1: Manage service configurations
Sv-Rm
Sv-Rm: Services Roadmap Evolution [SvcV-8]
Step 4.4: Define service deployment plans - to provision services in accordance with milestones corresponding to associated resource deployments implementing operational activities
Sv-Pr
Sv-Pr: Services Processes: Flow [SvcV-4]
Rs-Tx
Rs-Tx: Resources Taxonomy [SV-1/2]
Sv-Tx
Sv-Tx: Services Taxonomy: Actual Services [SvcV-1]

Figure 4:29 - Step 4.4: Define Service Deployment Plans

4.4.5 Service Analysis and Obligations

Step 4.5 – Analyze Service Obligations – Once a service architecture has been captured and defined, organization of the service agreement possibilities can begin. Service structure is analyzed in order to bundle or organize service level offerings. When needed, a Service Level Agreement (SLA) document can be developed based on the models and views generated during this Step 4. Ideally this SLA document is automatically generated from the model itself using reporting scripts and model queries.

- **Actual Service** – an individual **Service** (Note: an instance of a **Service** in the real world)

Sometimes it also necessary to develop a Service Offering Catalog to describe how actual services (in the form of ***provided service levels***) will be used to set SLAs and how they will be employed to achieve mission or business goals and imperatives. Typical ***measurements*** from Step 4.3 are used to generate ***actual measurements*** of the ***required*** and ***provided service levels*** as they deploy and evolve over the enterprise life cycle. Each actual measurement table is assigned start and end dates of their actual, estimated, or required points of existence, and are used to track validation, satisfaction, and realization of service level performance changes over time.

A ***risk*** analysis assesses the impact of events that may ***affect*** service assets. Measures and ***actual measurements*** of these ***risks*** are included with the service measurement analysis.

- **Provided Service Level** – a sub type of **Actual Service** that details a specific service level delivered by the provider
- **Required Service Level** – A sub type of **Actual Service** that details a specific service level required of the provider
- **Actual Measurement** – an actual value that is applied to a **Measurement** (Note: a measurement may be one of three kinds: actual, required, or estimate, and may have an associated start and/or end date)
- **Required** – an enumerated type of an actual measurement kind which is based on a required value

An entire service architecture, with all of its elements (contracts, policies, ports, interfaces, behaviors, etc.) may be considered as one option in an analysis of alternatives within the model. In other words, each alternative to be examined can be represented by a separate service architecture.

Additionally, a set of service ***requirements*** can be generated based on the elements within a Service Architecture, which are intended to be resource and personnel implementation agnostic. Requirements tracing from the Service Architecture can be satisfied by service, resource, personnel, and security solutions.

Requirements are developed and related to architectural elements relevant to the generation of the requirement. These are related using ***trace, verify, satisfy***, or ***refine*** relationships for linking to relevant elements in the architecture.

- **Requirement** – a statement that identifies a system, product, or process characteristic or constraint, which is unambiguous, clear, unique, consistent, stand-alone (not grouped), and verifiable, and is deemed necessary for stakeholder acceptability (INCOSE 2010)
- **Refine** – a relationship from an architectural element which refines a text-based requirement

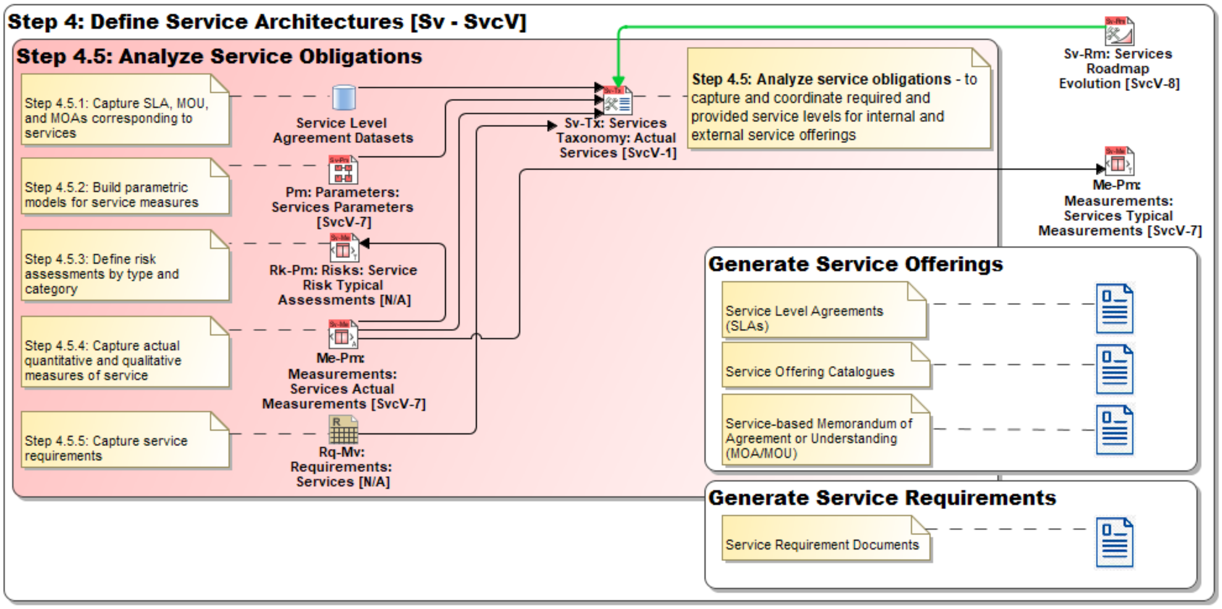

Step 4: Define Service Architectures [Sv - SvcV]
Step 4.5: Analyze Service Obligations
Step 4.5.1: Capture SLA, MOU, and MOAs corresponding to services
Service Level Agreement Datasets
Step 4.5.2: Build parametric models for service measures
Pm: Parameters: Services Parameters [SvcV-7]
Step 4.5.3: Define risk assessments by type and category
Rk-Pm: Risks: Service Risk Typical Assessments [N/A]
Step 4.5.4: Capture actual quantitative and qualitative measures of service
Me-Pm: Measurements: Services Actual Measurements [SvcV-7]
Step 4.5.5: Capture service requirements
Rq-Mv: Requirements: Services [N/A]
Sv-Tx: Services Taxonomy: Actual Services [SvcV-1]
Step 4.5: Analyze service obligations - to capture and coordinate required and provided service levels for internal and external service offerings
Sv-Rm: Services Roadmap Evolution [SvcV-8]
Me-Pm: Measurements: Services Typical Measurements [SvcV-7]
Generate Service Offerings
Service Level Agreements (SLAs)
Service Offering Catalogues
Service-based Memorandum of Agreement or Understanding (MOA/MOU)
Generate Service Requirements
Service Requirement Documents

Figure 4:30 - Step 4.5: Analyze Service Obligations

4.4.6 Architecture View Summary for Step 4

The view specifications in UAF for this viewpoint are outlined here:

	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Services Sv	Requirements Rq-Mv	Services Taxonomy **Sv-Tx**	Services Structure Sv-Sr	Services Connectivity Sv-Cn	Services Processes Sv-Pr	Services States Sv-St	Services Sequences Sv-Sq	Operational Information Model (Op-If)	**Envmt, Msmts, Risks (En-Pm, Me-Pm, Rk-Pm)**	Services Constraints Sv-Ct	Services Roadmap Sv-Rm	Services Traceability Sv-Tr

A summary of the lower-level steps and UAF views is shown below. The architecture views produced by each step are identified by the UAF grid designator Aa-Bb where Aa represents the Viewpoint row in the grid and Bb represents the Aspect column in the grid. DoDAF view designations are also shown in [brackets] when applicable. The name of the architecture view matches the name of the view specification in the UAFML. In some cases, a subtitle is provided to identify what kind of information is provided in an instance of that UAF view. There will be some architecture views that are not part of the UAF specification since you will sometimes need fit-for-purpose (i.e., custom) views that capture the necessary architectural information.

99	**Step 4: Define Service Architectures [Sv - SvcV]**	**Views**
100	**Step 4.1: Identify service opportunities** – that will correspond to service level agreements with internal and external service providers	Sv-Tx: Services Taxonomy [SvcV-1/2]
101	Step 4.1.1: Trace service contracts governing services that constrain connections within operational activities	Sv-Tr: Services Traceability: *Governance* [SvcV-5]
102	Step 4.1.2: Trace resource services implementing services	Rs-Tr: Services Traceability: *Resource Services* [N/A]
103	Step 4.1.3: Specify service rules, methods, and service policies in rule form	Sv-Ct: Services Constraints [SvcV-10a]
104	Step 4.1.4: Capture the environment and conditional constraints for services (e.g., operational areas, planning scenarios, threats, locations, etc.)	Sv-Ct: Services Constraints: Definition [SvcV-10a]
105	Step 4.1.5: Forecast services technology readiness against time	Sv-Rm: Services Roadmap Forecast [SvcV-9]
106	Step 4.1.6: Capture types and categories of services (both internal and external) for completeness	Sv-Tx: Services Taxonomy: *Service Offerings Table* [SvcV-1/2]
107	**Step 4.2: Capture service structures** - with regard to service roles, structural parts, and connectivity	Sv-Sr: Services Structure [SvcV-1/2]
108	Step 4.2.1: Specify service connections and interfaces	Sv-Cn: Services Connectivity [SvcV-3b/6]
109	Step 4.2.2: Capture internal structure of services	Sv-Sr: Services Structure: *Internal Structure* [SvcV-3b]
110	Step 4.2.3: Capture service operational information elements for all service functions to build the operational information model	Op-If: Operational Information Model [DIV-1]
111	**Step 4.3: Define service functions** - including service exchanges, states, sequences, and measures of service performance	Sv-Pr: Services Processes: *Flow* [SvcV-4]
112	Step 4.3.1: Trace service interfaces and functions implementing operational interfaces and activities	Sv-Tr: Services Traceability: *Operational Activities and Interfaces* [SvcV-5]
113	Step 4.3.2: Capture diagram of all service functions	Sv-Pr: Services Processes [SvcV-4]
114	Step 4.3.3: Capture structural relationships implied from service function actions	Sv-Pr: Services Processes: *Function Action Structures* [SvcV-4]
115	Step 4.3.4: Capture service states	Sv-St: Services States [SvcV-10b]
116	Step 4.3.5: Capture service action timelines	Sv-Sq: Services Sequences [SvcV-10c]
117	Step 4.3.6: Define measures of service performance by value type or enumeration	Me-Pm: Measurements: *Services Typical Measurements* [SvcV-7]
118	**Step 4.4: Define service deployment plans** - to provision services in accordance with milestones corresponding to associated resource deployments implementing operational activities	Sv-Rm: Services Roadmap Evolution [ScvV-8]
119	Step 4.4.1: Manage service configurations	Sv-Rm: Services Roadmap [SvcV-8]
120	**Step 4.5: Analyze service obligations** - to capture and coordinate required and provided service levels for internal and external service offerings	Sv-Tx: Services Taxonomy: *Actual Services* [SvcV-1]
121	Step 4.5.1: Capture SLA, MOU, and MOAs corresponding to services	*Service Level Agreement Datasets*
122	Step 4.5.2: Build parametric models for required service level measures	Pm: Parameters: *Services Parameters* [SvcV-7]
123	Step 4.5.3: Define risk assessments by type and category	Rk-Pm: Risks: *Service Risk Typical Assessments* [N/A]

124	Step 4.5.4: Capture required, estimated or actual quantitative and qualitative measures of service performance	Me-Pm: Measurements: *Services Actual Measurements* [SvcV-7]
125	Step 4.5.5: Capture service requirements	Rq-Mv: Requirements: *Services* [N/A]

STEP 5 – Resource Architectures

4.5 Step 5 – Resource Architectures

Purpose. The purpose of this step is to capture a solution architecture consisting of various resources, such as software, artifacts, capability configurations and natural resources that implement the operational elements and requirements in the operational architecture. Further design of a resource is typically detailed in SysML or UML. Key stakeholders for this step are Systems Engineers, Resource Owners, Implementers, Solution Providers, and Information Technology (IT) Architects. Their concerns are mainly about definition of solution architectures to implement operational elements and requirements.

Conceptual Schema. The key concepts in the Resources Viewpoint that can be used as model elements in the architecture views and the relationships between these concepts are illustrated in the conceptual schema shown in Figure 4:31. These ***key concepts*** are highlighted in italics within the Narrative and some of the less obvious concepts are listed with the associated UAF meaning of that concept. Detailed definition of the entities and relationships shown in the conceptual schema are provided in the UAFML specification document.

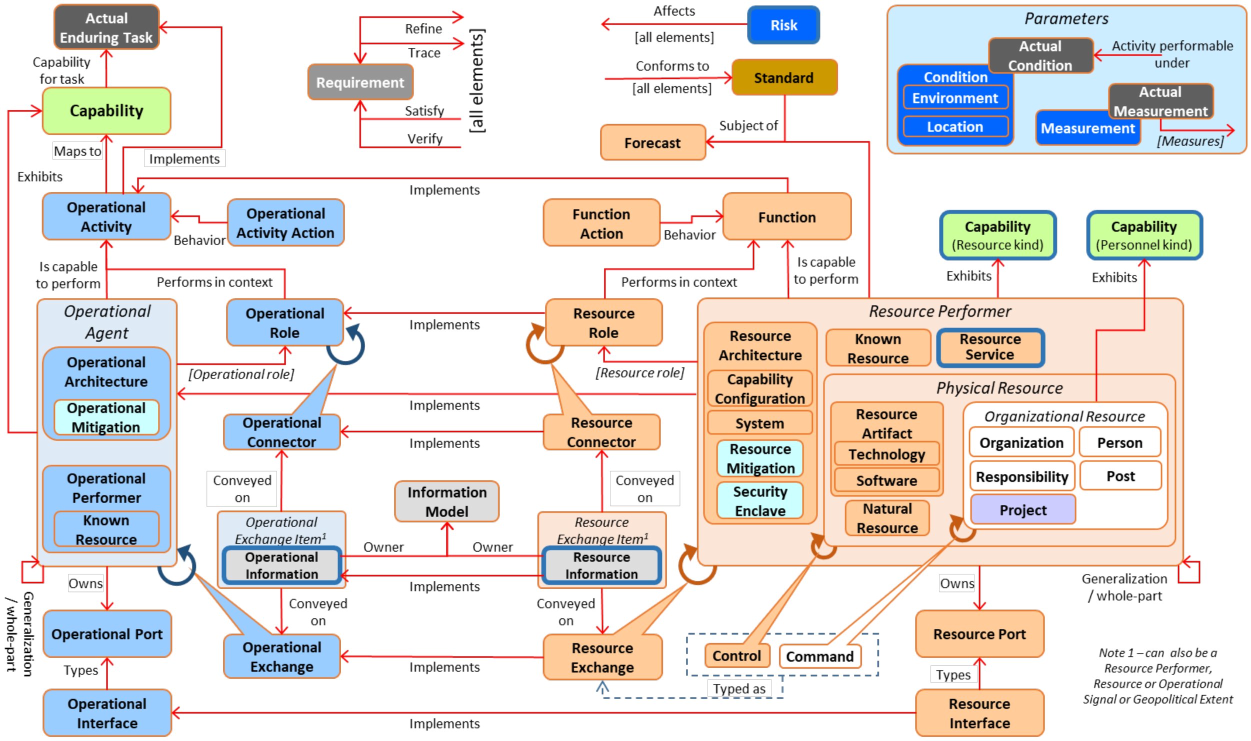


Figure 4:31 - Conceptual Schema for Resource Architectures

The operational architecture concepts are shown on the left in blue and the resource and personnel architecture concepts are shown on the right in orange and white, respectively. ***Resource performers*** and the ***functions*** performed by those performers will "implement" the ***operational agents*** and ***operational activities*** defined in the Operational Architecture.

Likewise, ***resource roles, resource connectors, resource exchange items, resource exchanges,*** and ***resource interfaces*** will implement the analogous things in the ***operational architecture***. Decisions about which resource elements (and what kinds of these elements) will implement operational elements are made in this Step 5. Resource mitigations can be devised to deal with resource security risks, and these will be addressed in Step 7.

It is common to consider the resource architecture to be an "implementation" architecture in the sense that architecture decisions and tradeoffs are being made when "implementing" the operational elements and service elements defined in Steps 3 and 4. Implementation in this sense means providing a practical means for accomplishing something.

This Guide is intended to be used in conjunction with the UAF Sample Problem that defines architecture views for a Search and Rescue Mission. UAF Specification documents (including the Sample Problem) can be downloaded from the OMG webpage: www.omg.org/spec/UAF/About-UAF/.

Step 5.0 – Define [Implementation] Resource Architectures – The main entry criterion for this Step is bringing forward the ***operational activities*** from Step 3 with associated ***performers***, ***operational exchanges***, ***measures*** and their operational context. These are used to define potential or actual ***resources*** for operational use, i.e., identification of resource ***functions*** to be used by ***operational activities*** that support operational needs for information or resources that can be provided on the basis of a ***resource architecture*** specification**.** These ***resources*** can also be used as the components of a designated ***resource architecture***.

An additional entry criterion for this Step, when applicable, is bringing forward the ***services*** from Step 4 with associated ***service interfaces*** and their operational usage or implementation context. The Operational Architecture is now examined for ***resource architecture*** implementations using various ***resource performers*** including, for example, ***resource artifacts***, ***capability configurations***, ***systems***, and ***natural resources***. These ***resource performers*** along with their associated ***exchanges*** and ***functions*** will form alternatives that are compared in trade-offs to support ***implementation*** decisions. When examining the resource performer options, the use of ***organizational resources*** is considered in Step 6 for specific trade-offs on how ***organizations***, ***posts***, and other human resources will be utilized, which could involve setting desired levels of automation.

- **Resource Performer** – an abstract grouping of elements that can perform **Functions** (Note: this is an entity that is capable of interacting with other resource performers to perform resource **Functions** and may include **Resource Architectures** and **Resource Artifacts**, among other things. See conceptual schema above.)
- **Resource Architecture** – a type used to denote a model of the Architecture, described from the **Resource Performer** perspective (Note: this typically represents a large composition or aggregation of resource performers (which is itself a resource performer) that can interact with other resources and perform functions)
- **Implements** – a tuple that defines how an element in the upper layer of abstraction is implemented by a semantically equivalent element (for example tracing the **Functions** to the **Operational Activities**) in the lower level of abstraction (Note: in this case these include implements relationships between resource elements that outfit or equip operational elements)

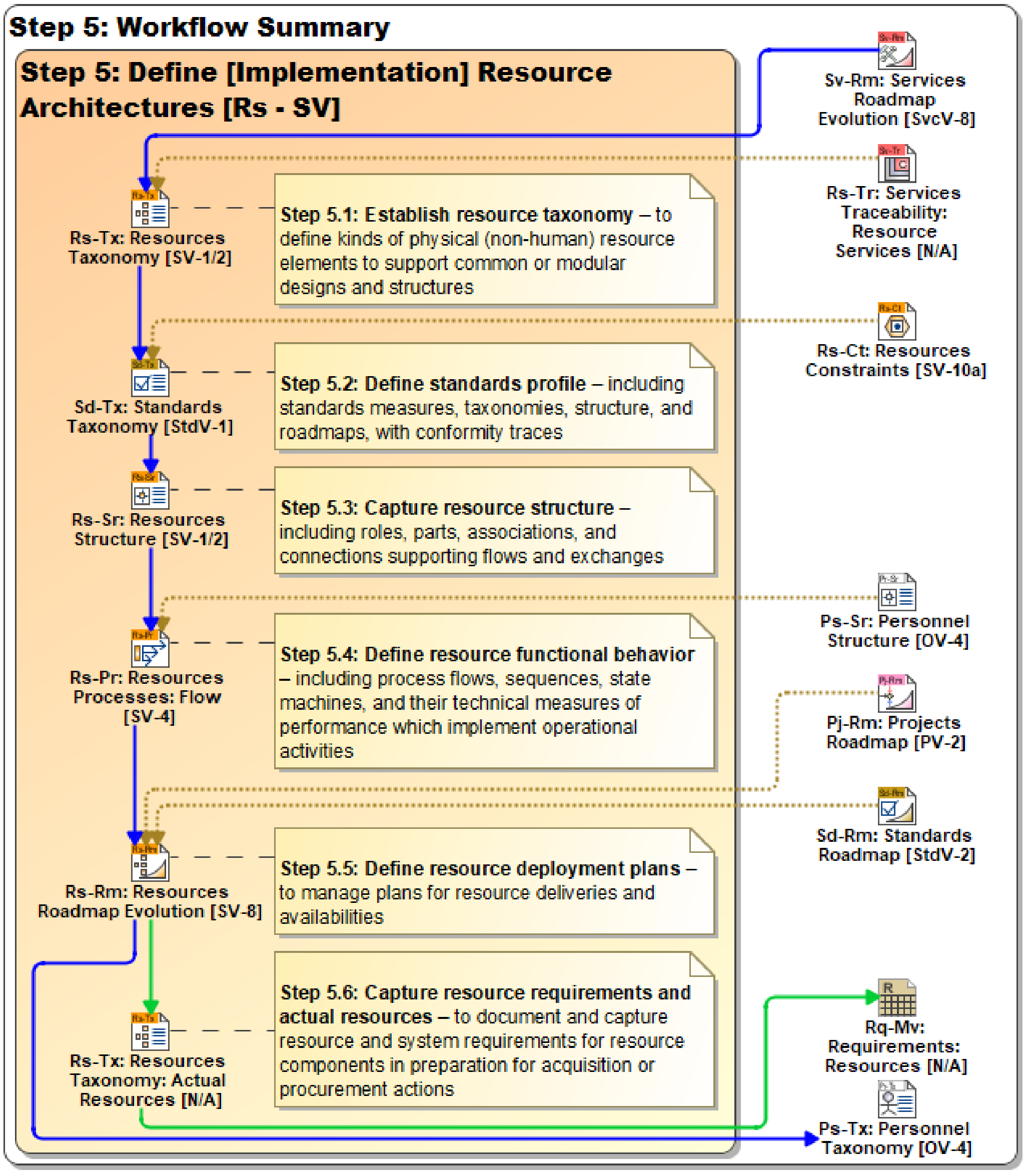

Step 5: Workflow Summary
Step 5: Define [Implementation] Resource Architectures [Rs - SV]
Rs-Tx: Resources Taxonomy [SV-1/2]
Step 5.1: Establish resource taxonomy – to define kinds of physical (non-human) resource elements to support common or modular designs and structures
Sd-Tx: Standards Taxonomy [StdV-1]
Step 5.2: Define standards profile – including standards measures, taxonomies, structure, and roadmaps, with conformity traces
Rs-Sr: Resources Structure [SV-1/2]
Step 5.3: Capture resource structure – including roles, parts, associations, and connections supporting flows and exchanges
Rs-Pr: Resources Processes: Flow [SV-4]
Step 5.4: Define resource functional behavior – including process flows, sequences, state machines, and their technical measures of performance which implement operational activities
Rs-Rm: Resources Roadmap Evolution [SV-8]
Step 5.5: Define resource deployment plans – to manage plans for resource deliveries and availabilities
Rs-Tx: Resources Taxonomy: Actual Resources [N/A]
Step 5.6: Capture resource requirements and actual resources – to document and capture resource and system requirements for resource components in preparation for acquisition or procurement actions
Sv-Rm: Services Roadmap Evolution [SvcV-8]
Rs-Tr: Services Traceability: Resource Services [N/A]
Rs-Ct: Resources Constraints [SV-10a]
Ps-Sr: Personnel Structure [OV-4]
Pj-Rm: Projects Roadmap [PV-2]
Sd-Rm: Standards Roadmap [StdV-2]
Rq-Mv: Requirements: Resources [N/A]
Ps-Tx: Personnel Taxonomy [OV-4]

Figure 4:32 - Workflow Summary for Step 5: Define [Implementation] Resource Architectures

A resource architecture defines resource *behavior* elements (e.g., processes, states, sequences) and allocates these to resource *structure* elements (e.g., resource configurations and artifacts, to be used in roles, connectors, exchange items, exchanges, ports, and interfaces). The behavior elements define resource ***functions*** that implement ***operational activities*** that will in turn help achieve enterprise goals.

Workflow Summary. A summary level view of the six steps involved in Step 5 workflow is shown above. The flowlines represent how one architecture view "influences" another architecture view. These lines do not represent a particular sequence of process activity execution or imply information exchanges. These should be thought of as influence diagrams rather than process diagrams. These architecture views (e.g., views, diagrams, tables) can be mapped to the architecture processes used in your organization and can be incorporated into an architecture modeling methodology.

The second-level steps in the Step 5 workflow are listed below along with the UAF views (and corresponding [DoDAF views] where applicable) associated with those steps. A complete list of the detailed steps at the third level with their corresponding views is provided at the end of this section.

126	**Step 5: Define [Implementation] Resource Architectures [Rs - SV]**	**Views**
127	**Step 5.1: Establish resource taxonomy** – to define kinds of physical (non-human) resource elements to support common or modular designs and structures	Rs-Tx: Resources Taxonomy [SV-1/2]
135	**Step 5.2: Define standards profile** – including standards measures, taxonomies, structure, and roadmaps, with conformity traces	Sd-Tx: Standards Taxonomy [StdV-1]
142	**Step 5.3: Capture resource structure** – including roles, parts, associations, and connections supporting flows and exchanges	Rs-Sr: Resources Structure [SV-1/2]
149	**Step 5.4: Define resource functional behavior** – including process flows, sequences, state machines, and their technical measures of performance which implement operational activities	Rs-Pr: Resources Processes: *Flow* [SV-4]
155	**Step 5.5: Define resource deployment plans** – to manage plans for resource deliveries and availabilities	Rs-Rm: Resources Roadmap Evolution [SV-8]
156	**Step 5.6: Capture resource requirements** – analyze resource alternatives to document and capture resource and system requirements for resource components in preparation for acquisition or procurement actions	Rs-Tx: Resources Taxonomy: *Actual Resources* [N/A]

4.5.1 Resource Taxonomy

Step 5.1 – Establish resource taxonomy – A set of ***resource performers*** are described, including any that have been preliminarily identified from Step 3.2, where resources are needed or anticipated that will ***implement operational agents***. These ***resource performers*** are accounted for in a ***resource taxonomy*** and may include ***resource artifacts*** (such as ***technology*** or ***software***), ***natural resources***, ***capability configurations***, ***systems***, ***known resources***, or entire ***resource architectures***. This step captures understanding of ***generalizations***, particularly when a reference resource architecture is setting contextual guidance, resource categories are established, or particular resource item plans are already known.

- **Capability Configuration** – a composite structure representing the physical and human resources (and their interactions) in an enterprise, assembled to meet a capability
- **System** – An integrated set of elements, subsystems, or assemblies that accomplish a defined objective, including products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements (INCOSE SE Handbook V4, 2015)
- **Resource Artifact** – a type of man-made object that contains no human beings (e.g., satellite, radio, petrol, gasoline, etc.) (Note: this includes subtypes such as **Software** and **Technology**)
- **Natural Resource** – a type of physical resource that occurs in nature such as oil, water, gas or coal (Note: this may also include other natural resources such as solar energy, ballast rock and electromagnetic spectrum)
- **Known Resource** – asserts that a known resource performer constrains the implementation of the **Operational Performer** that plays a particular role in the **Operational Architecture** (Note: a **Known Resource** may be a pre-existing entity, such as a physical resource or other operational agent which participates in an operational scenario and is already known and described outside the context of the **Operational Architecture**)
- **Resource Service** – a service that a Resource Performer provides to support higher level **Service** or **Operational Activity** (Note: employee provisioning, backup and recovery, storage, and self-service help desk are examples of **Resource Services** which are a set of functionalities that can be provided for the use of resource functions, which can implement an enterprise **Service** in the Services Viewpoint)

Resource architectures, along with their sub-type ***capability configurations*** and ***systems,*** are elements intended to describe composite structures of a complex nature that include humans and processes. The ***system*** element should not be used to describe a purely hardware and/or software device such as a satellite, aircraft, ship, a control suite, or other such resource artifacts. Such resource artifacts, like a ship, need people to be of any use and should instead be modeled as a structural part of a system (e.g., as a ***resource artifact*** element) rather than as a ***system*** element. A ***system*** may be composed of multiple ***capability configurations*** to represent the full nature of complexity in a system, and the fact that it may be capable of many things. It is usually helpful to make a clear distinction between *resource capabilities* (e.g., what a system is able to do something for someone or for some other system) and *operational capabilities* (e.g., what users are able to do when provided with various solutions in the form of artifacts, software, technologies, organizations, persons, etc.).

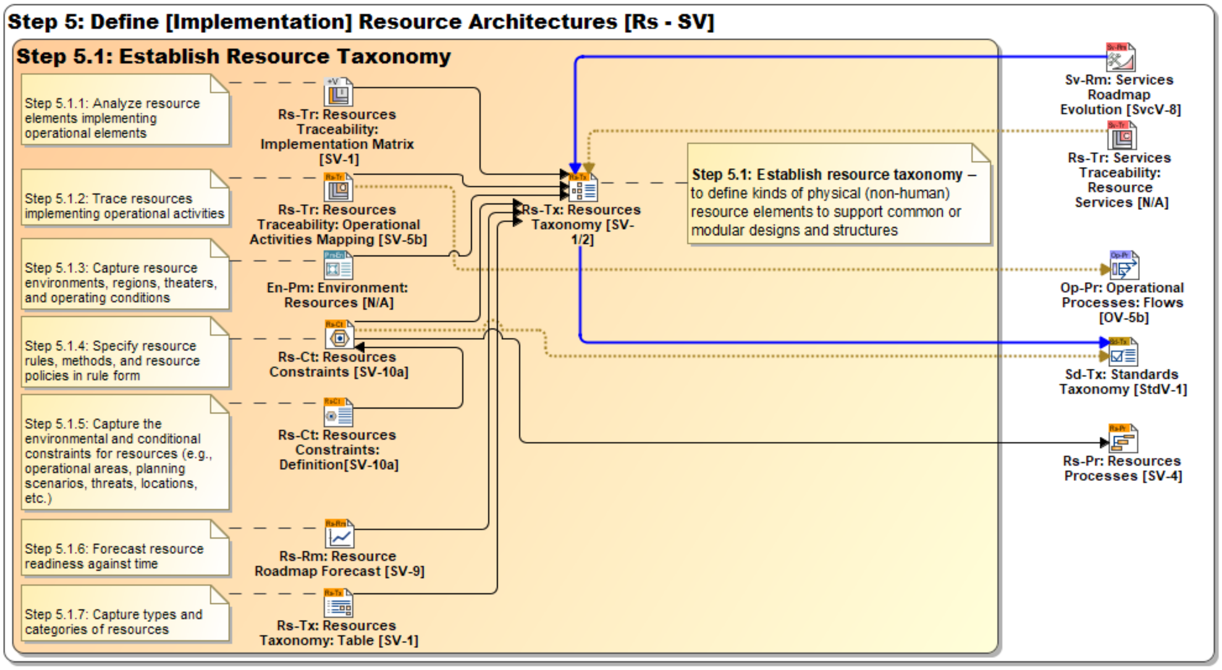

Step 5: Define [Implementation] Resource Architectures [Rs - SV]
Step 5.1: Establish Resource Taxonomy
Step 5.1.1: Analyze resource elements implementing operational elements
Step 5.1.2: Trace resources implementing operational activities
Step 5.1.3: Capture resource environments, regions, theaters, and operating conditions
Step 5.1.4: Specify resource rules, methods, and resource policies in rule form
Step 5.1.5: Capture the environmental and conditional constraints for resources (e.g., operational areas, planning scenarios, threats, locations, etc.)
Step 5.1.6: Forecast resource readiness against time
Step 5.1.7: Capture types and categories of resources
Rs-Tr: Resources Traceability: Implementation Matrix [SV-1]
Rs-Tr: Resources Traceability: Operational Activities Mapping [SV-5b]
En-Pm: Environment: Resources [N/A]
Rs-Ct: Resources Constraints [SV-10a]
Rs-Ct: Resources Constraints: Definition[SV-10a]
Rs-Rm: Resource Roadmap Forecast [SV-9]
Rs-Tx: Resources Taxonomy: Table [SV-1]
Rs-Tx: Resources Taxonomy [SV-1/2]
Step 5.1: Establish resource taxonomy – to define kinds of physical (non-human) resource elements to support common or modular designs and structures
Sv-Rm: Services Roadmap Evolution [SvcV-8]
Rs-Tr: Services Traceability: Resource Services [N/A]
Op-Pr: Operational Processes: Flows [OV-5b]
Sd-Tx: Standards Taxonomy [StdV-1]
Rs-Pr: Resources Processes [SV-4]

Figure 4:33 - Step 5.1: Establish Resource Taxonomy

The total environment developed in Step 2.1 expected for the deploying capabilities in terms of ***conditions***, ***locations***, ***environments***, including kinds of environments and ***kinds*** of locations, as well as ***geo-political*** factors is brought forward to assess their effects on resources, and becomes an overarching conditional context for resource parameters.

Rules and other resource policies are listed and applied to resource performers and then later expanded to their associated functions, roles and information elements. Where necessary, a resource parametric model can be used to evaluate or define resource policies. A resource technology ***forecast*** is assembled to show when key resources are forecast for periods applicable to implementation needs for the operational architecture.

A review is conducted to ensure all resources have been covered or addressed within the ***resources taxonomy table***.

- **Resource Constraint** – a rule governing the structural or functional aspects of an implementation
- **Forecast** – a dependency relationship that specifies a transition from one Resource Performer, **Standard**, or **Competence** to another future one, related to an **Actual Enterprise Phase** to give it a temporal context

4.5.2 Standards Profile

Step 5.2 – Define standards profile – A standards profile is created to define the technical ***standards*** applicable to the ***resource architecture***, identifying and listing applicable portions of existing or emerging standards, including ***protocols*** and ***protocol stacks***. Standards may also apply to any element in the architecture as necessary, but should not be used in lieu of a ***rule kind***, including policies, or ***standard operational activities***. ***Mandate*** and ***retirement dates*** should be included for all existing or anticipated standards, and the actual organization that ***ratifies*** the standard.

- **Standard** – a ratified and peer-reviewed specification that is used to guide or constrain the architecture. A **Standard** may be applied to any element in the architecture. (Note: a Standard may have a Mandated Date, a Retired Date, and may be Ratified By and **Actual Organization**)
- **Protocol** – a **Standard** for communication over a network, which may be composite, represented as a **Protocol Stack** made up of **Protocol Layers**
- **Protocol Stack** – a sub-type of **Protocol** that contains the **Protocol Layers**, defining a complete stack (Note: of protocols used in a communications network configuration)

A trace is made showing standards that resources must ***conform to***, as well as known implementations of protocols by ***resource connectors*** or ***resource ports***. As resource architecture is further developed in Step 5.3 and later, standards traces are updated.

- **Conforms to** – a dependency relationship that relates an element to a **Standard** that the element is conforming to
- **Protocol Implementation** – an abstract type grouping architectural elements that can implement **Protocols** (Note: such as a **Resource Connector** or **Resource Port**)

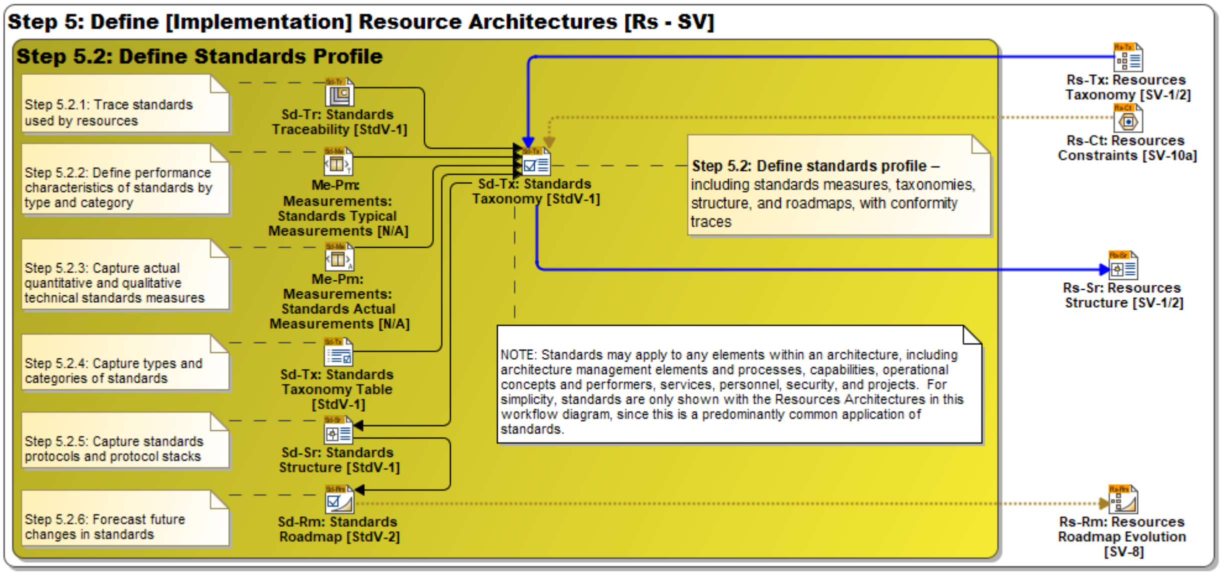
When protocols are described, they should be structured in terms of their protocol stack and set as adjacent protocol layers.

- **Protocol Layer** – usage of a **Protocol** in the context of another **Protocol** creating a whole-part relationship

Standards ***measures*** are defined to specify standard values which have been ***ratified*** by ***actual organizations***. Typical standard measures may be used to generate ***actual measurements*** to track changes to standards values over time. Each actual measurement table is assigned start and end dates of their actual, estimated, or required points of existence, and are used to track standards changes over time

- **Measurement** – a property of an element representing something in the physical world, expressed in amounts of a unit of measure

When ***standards*** are planned for future availability, a ***standards roadmap*** is assembled to show when ***standards*** will be ***released*** for employment, will be ***withdrawn***, or will be changed over time. The standards roadmap is often dependent on external factors, such as governing bodies for industry standards and regulatory agencies for government standards. This roadmap allows planners to understand when standards have been or will be ***ratified***, or no longer used. A ***timeline*** for the standards may include the associated resource architecture, or it may stand independently.



Step 5: Define [Implementation] Resource Architectures [Rs - SV]
Step 5.2: Define Standards Profile
Step 5.2.1: Trace standards used by resources
Sd-Tr: Standards Traceability [StdV-1]
Step 5.2.2: Define performance characteristics of standards by type and category
Me-Pm: Measurements: Standards Typical Measurements [N/A]
Step 5.2.3: Capture actual quantitative and qualitative technical standards measures
Me-Pm: Measurements: Standards Actual Measurements [N/A]
Step 5.2.4: Capture types and categories of standards
Sd-Tx: Standards Taxonomy Table [StdV-1]
Step 5.2.5: Capture standards protocols and protocol stacks
Sd-Sr: Standards Structure [StdV-1]
Step 5.2.6: Forecast future changes in standards
Sd-Rm: Standards Roadmap [StdV-2]
Sd-Tx: Standards Taxonomy [StdV-1]
Step 5.2: Define standards profile – including standards measures, taxonomies, structure, and roadmaps, with conformity traces
NOTE: Standards may apply to any elements within an architecture, including architecture management elements and processes, capabilities, operational concepts and performers, services, personnel, security, and projects. For simplicity, standards are only shown with the Resources Architectures in this workflow diagram, since this is a predominantly common application of standards.
Rs-Tx: Resources Taxonomy [SV-1/2]
Rs-Ct: Resources Constraints [SV-10a]
Rs-Sr: Resources Structure [SV-1/2]
Rs-Rm: Resources Roadmap Evolution [SV-8]

Figure 4:34 - Step 5.2: Define Standards Profile

4.5.3 Resource Structure

Step 5.3 – Capture Resource Structure – Internal structure of resource performers within the taxonomy is developed with internal features and characteristics, including ***resource roles*** they will have as part of a greater resource performer. One resource may perform different resource roles, and one resource role may be performed by different resource performers. Examination of this is necessary to scope and perform assessment of the trade space and conduct an analysis of alternatives, including differences between composition and aggregation in resource performer structure. Formal ***resource interfaces*** may be defined and declared for interface points of any resource performer port, which implement service or operational interfaces.

- **Resource Role** – usage of a Resource Performer in the context of another Resource Performer creating a whole-part relationship (Note: this is an element representing the part played by one resource in a particular context, governed by a **Role Kind**, such as Part, Component, Used Configuration, Human Resource, Platform, System, Sub Organization, Post Role, Responsibility Role, Equipment, Sub System Part, Hosted Software, Artifact Component, Natural Resource Component, or Other kind of role)
- **Resource Interface** – a declaration that specifies a contract between the Resource Performers it is related to and any other Resource Performers it can interact with. It is also intended to be an implementation of a specification of an Interface in the Business and/or Service layer. (Note: this may include the idea of a formal declaration of a contractual level interface)

All of the resource performers are associated with each other to identify resource performer relationships which will support ***resource exchanges***, and ***resource connections***.

- **Resource Exchange** – asserts that a flow can exist between Resource Performers (i.e., flows of data, people, material, or energy)
- **Resource Exchange item** – an abstract grouping for elements that defines the types of elements that can be exchanged between Resource Performers and conveyed by a **Resource Exchange** (Note: these may be a Resource Performer, **Resource Information**, or a **Geopolitical Extent Type**)
- **Resource Connector** – a channel for exchange between two **Resource Roles**

A resource ***logical information model*** (LIM) is created to define the ***resource information elements*** which are exchanged. ***Resource exchanges*** are placed on all associations with groupings of ***resource exchange items*** in accordance with logical actions or sequences of the resource performers. Multiple associations may exist between resource performers which represent different types of exchanges in terms of time, sequence, interface definition, or kinds of items that are exchanged (which for the resource architecture consists of ***resource information*** elements***,*** ***resource performers,*** and ***signals***).

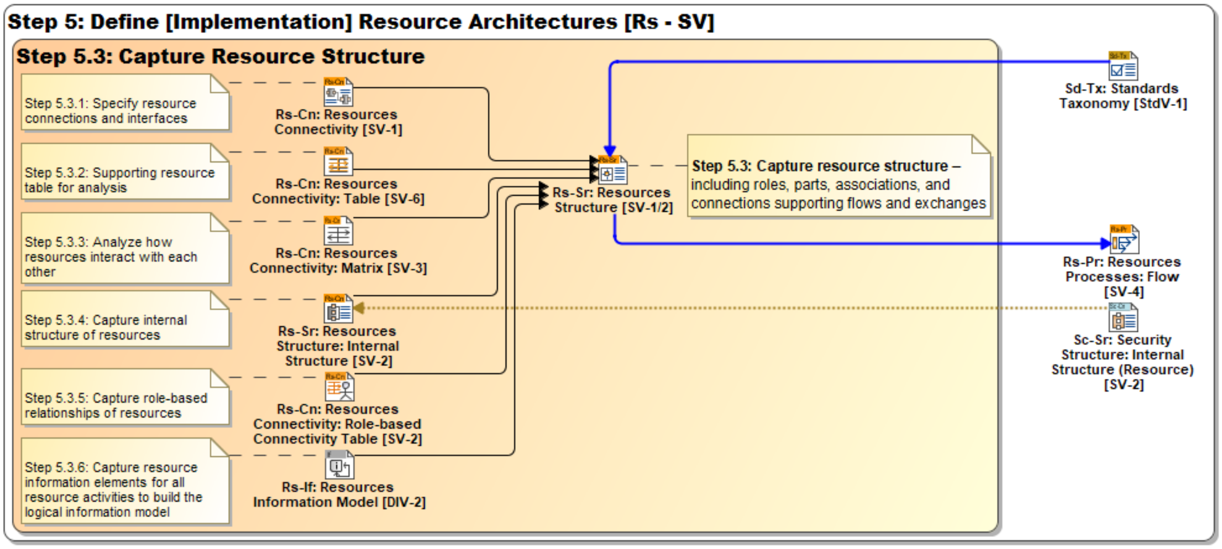

Step 5: Define [Implementation] Resource Architectures [Rs - SV]
Step 5.3: Capture Resource Structure
Step 5.3.1: Specify resource connections and interfaces
Rs-Cn: Resources Connectivity [SV-1]
Step 5.3.2: Supporting resource table for analysis
Rs-Cn: Resources Connectivity: Table [SV-6]
Step 5.3.3: Analyze how resources interact with each other
Rs-Cn: Resources Connectivity: Matrix [SV-3]
Step 5.3.4: Capture internal structure of resources
Rs-Sr: Resources Structure: Internal Structure [SV-2]
Step 5.3.5: Capture role-based relationships of resources
Rs-Cn: Resources Connectivity: Role-based Connectivity Table [SV-2]
Step 5.3.6: Capture resource information elements for all resource activities to build the logical information model
Rs-If: Resources Information Model [DIV-2]
Rs-Sr: Resources Structure [SV-1/2]
Step 5.3: Capture resource structure – including roles, parts, associations, and connections supporting flows and exchanges
Sd-Tx: Standards Taxonomy [StdV-1]
Rs-Pr: Resources Processes: Flow [SV-4]
Sc-Sr: Security Structure: Internal Structure (Resource) [SV-2]

Figure 4:35 - Step 5.3: Capture Resource Structure

4.5.4 Resource Functional Behavior

Step 5.4 – Define Resource Functional Behavior – Process flow diagrams are constructed for all key ***resource functions*** including ***resource function actions*** classified by ***resource functions*** and process control-flow mechanisms such as decision nodes and forks. When ***resource performers*** are already known or defined in Step 5.1, they are assigned to swim-lanes and the actions they are capable to perform. Otherwise, ***function actions*** are grouped into swim-lanes to create ***resource performers*** which must be structured in Step 5.1 – 5.3.

The process flows are refined, when needed, with ***resource state descriptions*** for performers and sequenced timelines of ***resource messages*** and ***resource methods***. ***Resource connections*** are added to the process flow to realize the ***resource exchange items*** (which for the resource architecture may consist of ***resource information elements, resource performers, resource signals***, and ***geopolitical extents***). State descriptions and sequenced timelines may now be developed to supplement or refine the item exchanges.

- **Resource Message** – a message for use in a Resource Event-Trace which carries any of the subtypes of **Resource Exchange** (Note: this may include **Resource Methods**)
- **Resource Method** – a behavioral feature of a **Resource Performer** whose behavior is specified in a **Function**

Technical performance ***measures*** are defined for the resource performers and their ***functions***. Technical performance ***measures*** (TPMs) may include parametric diagrams when needed. Measures of overall functions and resource performers should demonstrate satisfaction of operational implementations either directly or indirectly through examination or correlation of resource TPMs with the operational MOPs.

- **Measurement** – a property of an element representing something in the physical world, expressed in amounts of a unit of measure

When developing a business architecture, the personnel behavior functions and sequenced timelines, and their implementation of operational activities and service functions, are used to capture the business aspects of the enterprise. Business Process Model and Notation (BPMN)[6] is an alternative diagram for these views in the UAFML.

A review is conducted to ensure all relevant ***operational activities*** and their performers have been ***implemented*** by ***functions***, and their ***resource performers***. Additionally, functional structural relationships are examined through intervening ***function actions***, to check for redesign or simplification, as well as possible duplication in implementation coverage of operational activities. Functions may be viewed either in complex meta-chain maps or in simple diagrams to ensure a complete description of ***functions*** is established and understood. This description may be arranged by function groupings, resource performers capable to perform them, or some other useful organization.

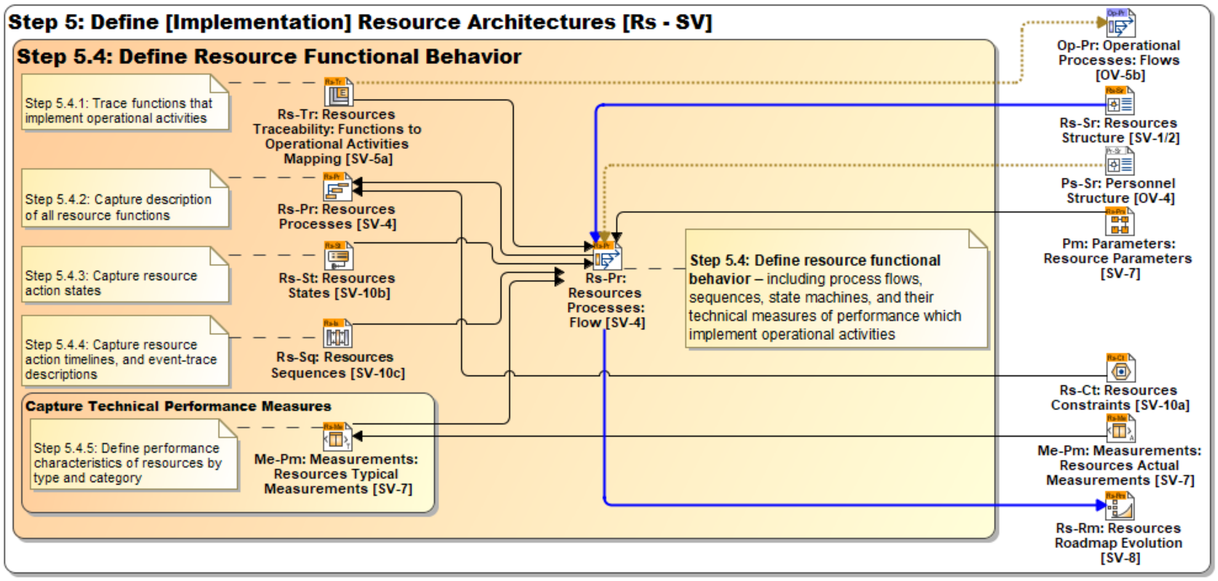

Step 5: Define [Implementation] Resource Architectures [Rs - SV]
Step 5.4: Define Resource Functional Behavior
Step 5.4.1: Trace functions that implement operational activities
Rs-Tr: Resources Traceability: Functions to Operational Activities Mapping [SV-5a]
Step 5.4.2: Capture description of all resource functions
Rs-Pr: Resources Processes [SV-4]
Step 5.4.3: Capture resource action states
Rs-St: Resources States [SV-10b]
Step 5.4.4: Capture resource action timelines, and event-trace descriptions
Rs-Sq: Resources Sequences [SV-10c]
Capture Technical Performance Measures
Step 5.4.5: Define performance characteristics of resources by type and category
Me-Pm: Measurements: Resources Typical Measurements [SV-7]
Rs-Pr: Resources Processes: Flow [SV-4]
Step 5.4: Define resource functional behavior – including process flows, sequences, state machines, and their technical measures of performance which implement operational activities
Op-Pr: Operational Processes: Flows [OV-5b]
Rs-Sr: Resources Structure [SV-1/2]
Ps-Sr: Personnel Structure [OV-4]
Pm: Parameters: Resource Parameters [SV-7]
Rs-Ct: Resources Constraints [SV-10a]
Me-Pm: Measurements: Resources Actual Measurements [SV-7]
Rs-Rm: Resources Roadmap Evolution [SV-8]

Figure 4:36 - Step 5.4: Define Resource Functional Behavior

4.5.5 Resource Deployment Plans

Step 5.5 – Define Resource Deployment Plans – When resources and resource technologies are planned for future availability, a ***resources roadmap*** is assembled to show when resource performers will be ***released*** for employment, will be ***withdrawn***, or will be changed over time. This roadmap allows resource acquisition planners to understand when resource performers for implementation are deployed, in service, out of service, or no longer used. Actual milestones are created for service specifications to denote their availability on a ***timeline***.

- **Version Released at Milestone** – an actual milestone category showing a version of an element to be released (Note: in this case a Resource)
- **Version Withdrawn at Milestone** – an actual milestone category showing a version of an element to be withdrawn (Note: in this case a Resource)

[6] ***ISO 19510:2013 – Information technology — Object Management Group Business Process Model and Notation*** establishes a set of notations and semantics for collaboration, process, and choreography diagrams to communicate process information to other business users, process implementers, customers and suppliers (see also OMG standard BPMN 2.0.2)

Resource provisioning is governed by ***version succession*** of ***versions of configurations*** within ***whole life configurations*** for resources. Versions of configuration are structured within whole life configurations to organize the deployment of resources. Various resource performers ***succeed*** others as versions of the resources are changed.

- **Whole Life Configuration** – a set of Versioned Elements (Note: in this case **Resources**)
- **Version of Configuration** – a property of a **Whole Life Configuration**, used in version control of a Versioned Element. It asserts that a Versioned Element is a version of a **Whole Life Configuration**. (Note: in this case a particular version of a Resource)
- **Version Succession** – a tuple between two **Version Of Configurations** that denotes that one **Version Of Configuration** follows from another

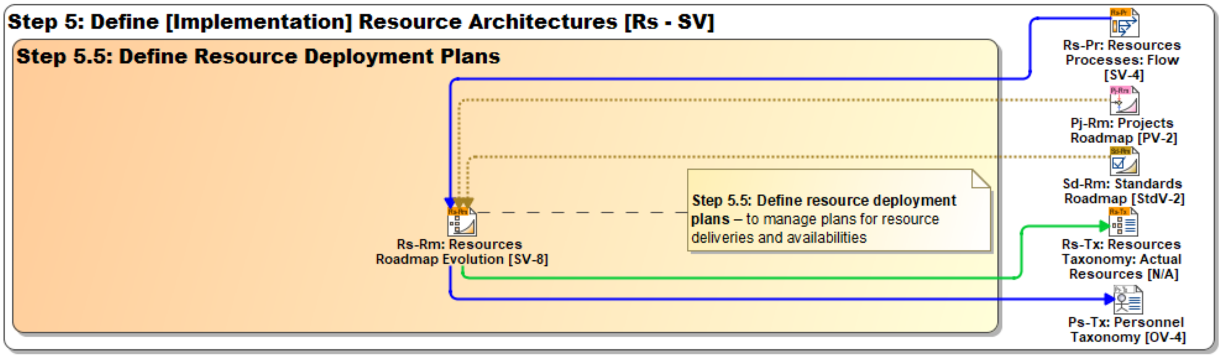

Step 5: Define [Implementation] Resource Architectures [Rs - SV]
Step 5.5: Define Resource Deployment Plans
Rs-Rm: Resources Roadmap Evolution [SV-8]
Step 5.5: Define resource deployment plans – to manage plans for resource deliveries and availabilities
Rs-Pr: Resources Processes: Flow [SV-4]
Pj-Rm: Projects Roadmap [PV-2]
Sd-Rm: Standards Roadmap [StdV-2]
Rs-Tx: Resources Taxonomy: Actual Resources [N/A]
Ps-Tx: Personnel Taxonomy [OV-4]

Figure 4:37 - Step 5.5: Define Resource Deployment Plans

4.5.6 Physical Resource Analysis and Requirements

Step 5.6 – Perform Analysis and Capture Resource Requirements – Once a resource architecture has been captured and defined, organization of resource implementation possibilities can begin. Resource structure is analyzed to bundle or organize resource deployments, or to examine variations and specializations in resource utilization. Impact analysis is conducted on resource and resource role implementations. Typically, feasibility, cost and other factors may be tied to the resource elements to drive other kinds of analysis.

- **Actual Resource** – an individual, fully-realized Resource Performer (Note: in other words, an instance of a **Resource** in the real world, including a **Fielded Capability** or an **Actual Organizational Resource**)

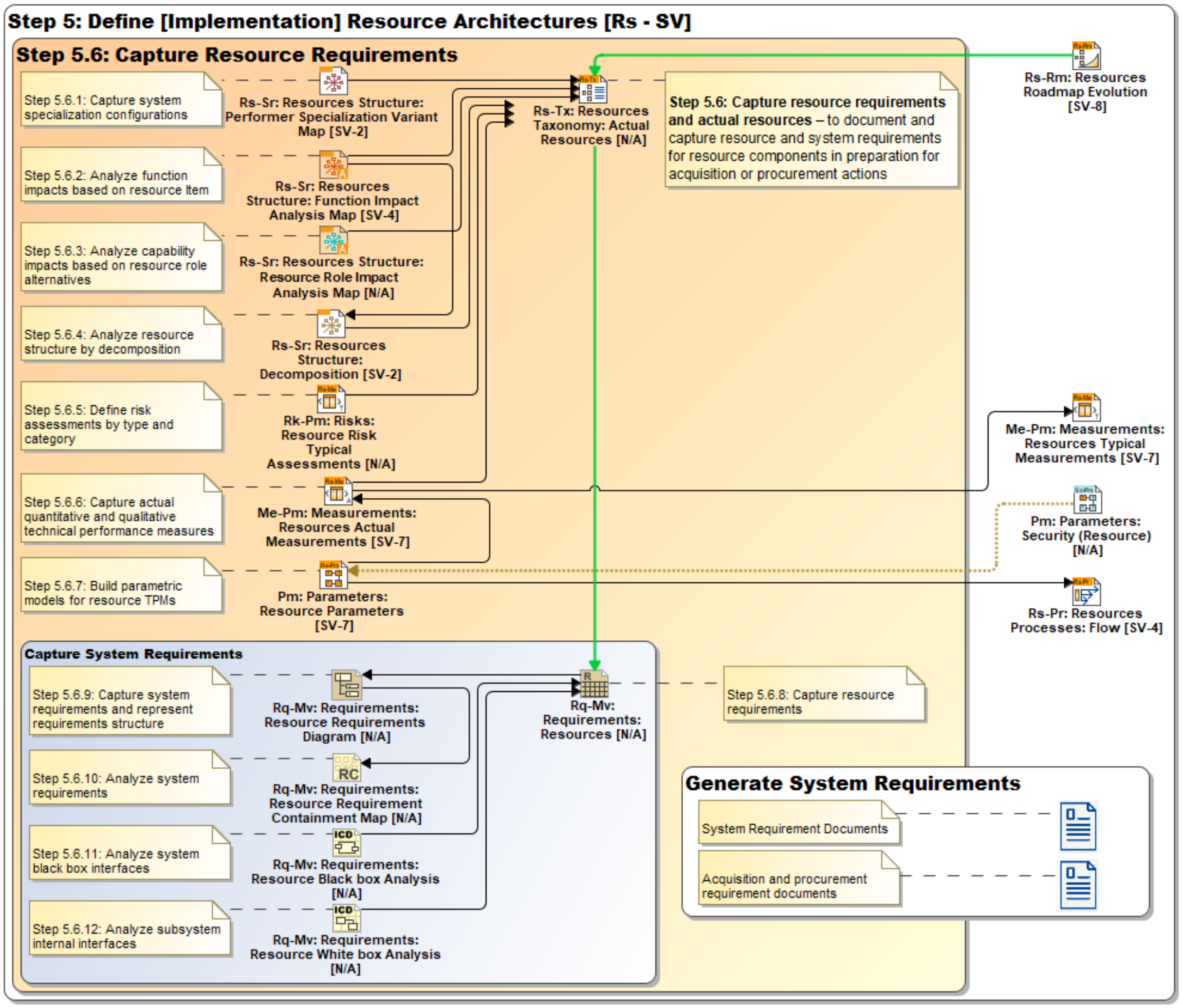

Step 5: Define [Implementation] Resource Architectures [Rs - SV]
Step 5.6: Capture Resource Requirements
Step 5.6.1: Capture system specialization configurations
Rs-Sr: Resources Structure: Performer Specialization Variant Map [SV-2]
Step 5.6.2: Analyze function impacts based on resource Item
Rs-Sr: Resources Structure: Function Impact Analysis Map [SV-4]
Step 5.6.3: Analyze capability impacts based on resource role alternatives
Rs-Sr: Resources Structure: Resource Role Impact Analysis Map [N/A]
Step 5.6.4: Analyze resource structure by decomposition
Rs-Sr: Resources Structure: Decomposition [SV-2]
Step 5.6.5: Define risk assessments by type and category
Rk-Pm: Risks: Resource Risk Typical Assessments [N/A]
Step 5.6.6: Capture actual quantitative and qualitative technical performance measures
Me-Pm: Measurements: Resources Actual Measurements [SV-7]
Step 5.6.7: Build parametric models for resource TPMs
Pm: Parameters: Resource Parameters [SV-7]
Rs-Tx: Resources Taxonomy: Actual Resources [N/A]
Step 5.6: Capture resource requirements and actual resources – to document and capture resource and system requirements for resource components in preparation for acquisition or procurement actions
Rs-Rm: Resources Roadmap Evolution [SV-8]
Me-Pm: Measurements: Resources Typical Measurements [SV-7]
Pm: Parameters: Security (Resource) [N/A]
Rs-Pr: Resources Processes: Flow [SV-4]
Capture System Requirements
Step 5.6.9: Capture system requirements and represent requirements structure
Rq-Mv: Requirements: Resource Requirements Diagram [N/A]
Step 5.6.10: Analyze system requirements
Rq-Mv: Requirements: Resource Requirement Containment Map [N/A]
Step 5.6.11: Analyze system black box interfaces
Rq-Mv: Requirements: Resource Black box Analysis [N/A]
Step 5.6.12: Analyze subsystem internal interfaces
Rq-Mv: Requirements: Resource White box Analysis [N/A]
Rq-Mv: Requirements: Resources [N/A]
Step 5.6.8: Capture resource requirements
Generate System Requirements
System Requirement Documents
Acquisition and procurement requirement documents

Figure 4:38 - Step 5.6: Capture Resource Requirements

An entire resource architecture with all of its elements (agents, behaviors, etc.) may be set as one option in an analysis of alternatives between others. Each alternative to be examined can be represented by a separate resource architecture. Typical technical performance measures (TPMs) from Step 5.4 may be used to generate ***actual measurements*** to either capture analysis of these alternatives or records of analysis for resource performers over time. Each actual measurement table is assigned start and end dates of their actual, estimated, or required points of existence, and are used to track validation, satisfaction, and realization of technical performance changes over time.

A ***risk*** analysis assesses the impact of events that may ***affect*** resource assets. Measures and ***actual measurements*** of these ***risks*** are included with the resource measurement analysis.

- **Actual Measurement** – a real or estimated value or enumerated account taken by a ***measurement*** at a specific point in time (Note: a measurement may be one of three kinds: actual, required, or estimate, and may have an associated start and/or end date)
- **Required** – an enumerated type of an actual measurement kind which is based on a required value

If the resource architecture is used as a reference architecture to evaluate bids for contracts from resource developers, ***actual measurements*** may form the basis of evaluation for bids received on parts or for all of the resource architecture.

When needed, a System Requirement Document (SRD) can be developed based on the models and views generated during this Step 5. Ideally this SRD document is automatically generated from the model itself using reporting scripts and model queries. Additionally, the architectural description itself may be published or shared with vendors and bidders to communicate purchase plans, requests for information, and requests for proposals. When ***requirements*** are shown embedded in architectural context, bidders and vendors will have a better contextual understanding for the requirements and may use approved architectural descriptions to stem off their own resource designs using SysML, UML and other languages compatible with a UAF-based architecture.

Requirements are developed and related to architectural elements relevant to the generation of the requirement. These are related using ***trace, verify, satisfy***, or ***refine*** relationships for linking to relevant elements in the architecture.

- **Requirement** – a statement that identifies a system, product or process characteristic or constraint, which is unambiguous, clear, unique, consistent, stand-alone (not grouped), and verifiable, and is deemed necessary for stakeholder acceptability (INCOSE 2010)
- **Refine** – a relationship from an architectural element which refines a text-based requirement

4.5.7 Architecture View Summary for Step 5

The view specifications in UAF for this viewpoint are outlined here:

	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Resources Rs	Requirements Rq-Mv	Resources Taxonomy Rs-Tx	Resources Structure Rs-Sr	Resources Connectivity Rs-Cn	Resources Processes Rs-Pr	Resources States Rs-St	Resources Sequences Rs-Sq	Resources Information Model (Rs-If)	Envmt, Msmts, Risks (En-Pm, Me-Pm, Rk-Pm)	Resources Constraints Rs-Ct	Resources Roadmaps: Evolution, Forecast Rs-Rm-E, -F	Resources Traceability Rs-Tr
Standards Sd	-	Standards Taxonomy Sd-Tx	Standards Structure Sd-Sr	-	-	-	-	-	Envmt, Msmts, Risks (En-Pm, Me-Pm, Rk-Pm)	-	Standards Roadmap Sd-Rm	Standards Traceability Sd-Tr

A summary of the lower-level steps and UAF views is shown below. The architecture views produced by each step are identified by the UAF grid designator Aa-Bb where Aa represents the Viewpoint row in the grid and Bb represents the Aspect column in the grid. DoDAF view designations are also shown in [brackets], when applicable. The name of the architecture view matches the name of the view specification in the UAFML. In some cases, a subtitle is provided to identify what kind of information is provided in an instance of that UAF view. There will be some architecture views that are not part of the UAF specification since you will sometimes need fit-for-purpose (i.e., custom) views that capture the necessary architectural information.

126	**Step 5: Define [Implementation] Resource Architectures [Rs - SV]**	**Views**
127	**Step 5.1: Establish resource taxonomy** – to define kinds of physical (non-human) resource elements to support common or modular designs and structures	Rs-Tx: Resources Taxonomy [SV-1/2]
128	Step 5.1.1: Analyze resource elements implementing operational elements	Rs-Tr: Resources Traceability: *Implementation Matrix* [SV-5b]
129	Step 5.1.2: Trace resources implementing operational activities	Rs-Tr: Resources Traceability: *Operational Activities Mapping* [SV-5b]
130	Step 5.1.3: Capture resource environments, regions, theaters, and operating conditions	En-Pm: Environment: *Resources* [N/A]
131	Step 5.1.4: Specify resource rules, methods, and resource policies in rule form	Rs-Ct: Resources Constraints [SV-10a]
132	Step 5.1.5: Capture the environmental and conditional constraints for resources (e.g., operational areas, planning scenarios, threats, locations, etc.)	Rs-Ct: Resources Constraints: *Definition* [SV-10a]
133	Step 5.1.6: Forecast resource readiness against time	Rs-Rm: Resource Roadmap Forecast [SV-9]
134	Step 5.1.7: Capture types and categories of resources	Rs-Tx: Resources Taxonomy: *Table* [SV-1]
135	**Step 5.2: Define standards profile** – including standards measures, taxonomies, structure, and roadmaps, with conformity traces	Sd-Tx: Standards Taxonomy [StdV-1]
136	Step 5.2.1: Trace standards used by resources	Sd-Tr: Standards Traceability [StdV-1]
137	Step 5.2.2: Define performance characteristics of standards by type and category	Me-Pm: Measurements: *Standards Typical Measurements* [N/A]
138	Step 5.2.3: Capture actual quantitative and qualitative technical standards measures	Me-Pm: Measurements: *Standards Actual Measurements* [N/A]
139	Step 5.2.4: Capture types and categories of standards	Sd-Tx: Standards Taxonomy: *Table* [StdV-1]
140	Step 5.2.5: Capture standards protocols and protocol stacks	Sd-Sr: Standards Structure [StdV-1]
141	Step 5.2.6: Forecast future changes in standards	Sd-Rm: Standards Roadmap [StdV-2]

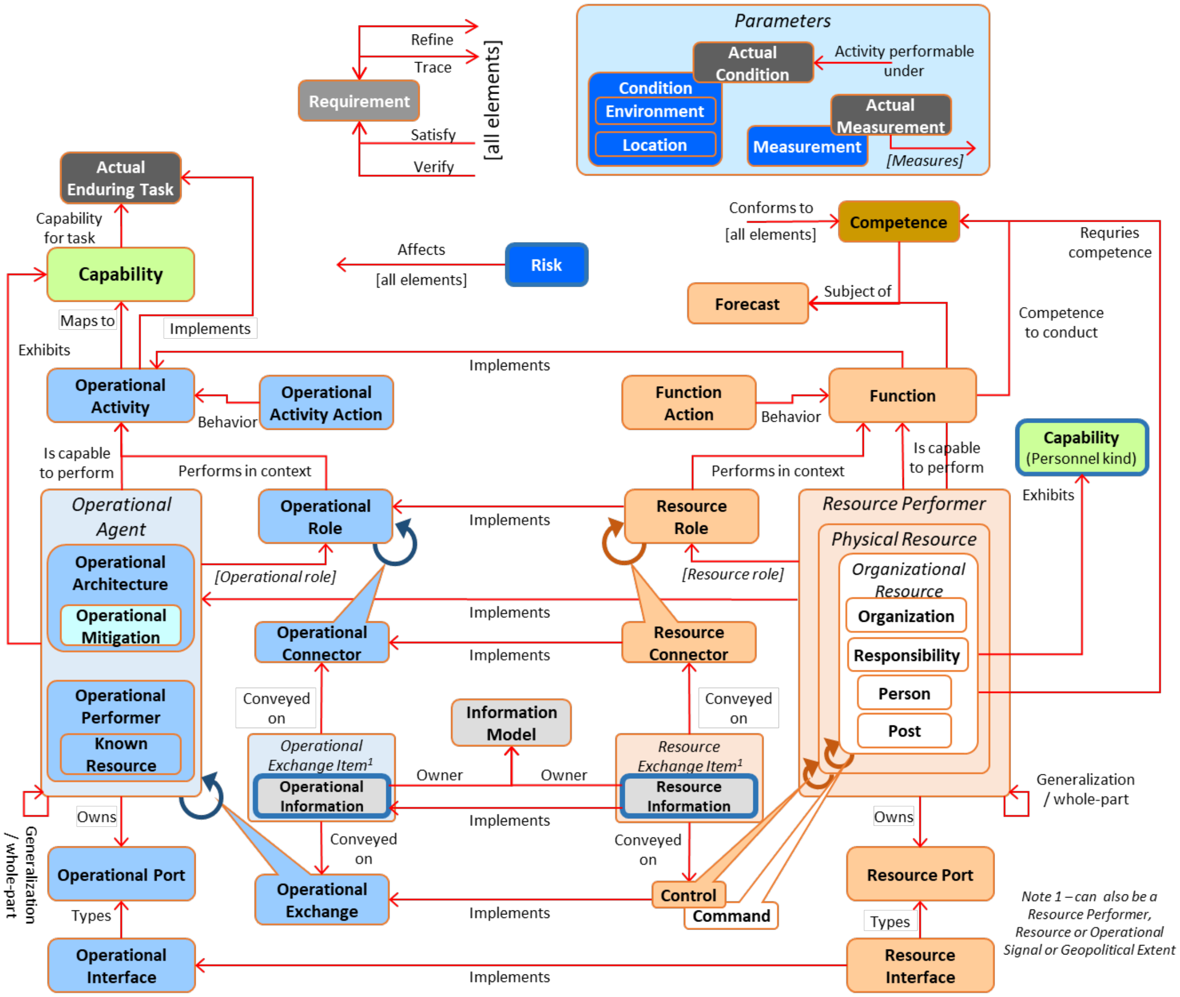
142	**Step 5.3: Capture resource structure** – including roles, parts, and associations and connections supporting flows and exchanges	Rs-Sr: Resources Structure [SV-1/2]
143	Step 5.3.1: Specify resource connections and interfaces	Rs-Cn: Resources Connectivity [SV-1]
144	Step 5.3.2: Supporting resource table for analysis	Rs-Cn: Resources Connectivity: *Table* [SV-6]
145	Step 5.3.3: Analyze how resources interact with each other	Rs-Cn: Resources Connectivity: *Matrix* [SV-3]
146	Step 5.3.4: Capture internal structure of resources	Rs-Sr: Resources Structure: *Internal Structure* [SV-2]
147	Step 5.3.5: Capture role-based relationships of resources	Rs-Cn: Resources Connectivity: *Role-based Connectivity Table* [SV-2]
148	Step 5.3.6: Capture resource information elements for all resource activities to build the logical information model	Rs-If: Resources Information Model [DIV-2]
149	**Step 5.4: Define resource functional behavior** – including process flows, sequences, state machines, and their technical measures of performance which implement operational activities	Rs-Pr: Resources Processes: *Flow* [SV-4]
150	Step 5.4.1: Trace functions that implement operational activities	Rs-Tr: Resources Traceability: *Functions to Operational Activities Mapping* [SV-5a]
151	Step 5.4.2: Capture description of all resource functions	Rs-Pr: Resources Processes [SV-4]
152	Step 5.4.3: Capture resource action states	Rs-St: Resources States [SV-10b]
153	Step 5.4.4: Capture resource action timelines, and event-trace descriptions	Rs-Sq: Resources Sequences [SV-10c]
154	Step 5.4.5: Define performance characteristics of resources by type and category	Me-Pm: Measurements: *Resources Typical Measurements* [SV-7]
155	**Step 5.5: Define resource deployment plans** - to manage plans for resource deliveries and availabilities	Rs-Rm: Resources Roadmap Evolution [SV-8]
156	**Step 5.6: Perform analysis and capture resource requirements** – to document and capture resource and system requirements for resource components in preparation for acquisition or procurement actions	Rs-Tx: Resources Taxonomy: *Actual Resources* [N/A]
157	Step 5.6.1: Capture system specialization configurations	Rs-Sr: Resources Structure: *Performer Specialization Variant Map* [SV-2]
158	Step 5.6.2: Analyze function impacts based on resource Item	Rs-Sr: Resources Structure: *Function Impact Analysis Map* [SV-4]
159	Step 5.6.3: Analyze capability impacts based on resource role alternatives	Rs-Sr: Resources Structure: *Resource Role Impact Analysis Map*
160	Step 5.6.4: Analyze resource structure by decomposition	Rs-Sr: Resources Structure: *Decomposition* [SV-2]
161	Step 5.6.5: Define risk assessments by type and category	Rk-Pm: Risks: *Resource Risk Typical Assessments* [N/A]
162	Step 5.6.6: Capture actual quantitative and qualitative technical performance measures	Me-Pm: Measurements: *Resources Actual Measurements* [SV-7]
163	Step 5.6.7: Build parametric models for resource TPMs	Pm: Parameters: *Resource Parameters* [SV-7]
164	Step 5.6.8: Capture resource requirements	Rq-Mv: Requirements: *Resources* [N/A]
165	Step 5.6.9: Capture system requirements and represent requirements structure	Rq-Mv: Requirements: *Resource Requirements Diagram* [N/A]
166	Step 5.6.10: Analyze system requirements	Rq-Mv: Requirements: *Resource Requirement Containment Map* [N/A]
167	Step 5.6.11: Analyze system black box interfaces	Rq-Mv: Requirements: *Resource Black box Analysis* [N/A]
168	Step 5.6.12: Analyze subsystem internal interfaces	Rq-Mv: Requirements: *Resource White box Analysis* [N/A]

STEP 6 – Personnel Architectures

4.6 Step 6 – Personnel Architectures

Purpose. The purpose of this step is to clarify the role of Human Factors when creating architectures in order to facilitate both Human Factors Integration and Systems Engineering. Key stakeholders for this step are Personnel involved in operations of the enterprise, Solution Providers and Project Managers. Their concerns are mainly about what are the roles and responsibilities of humans in the enterprise operations and how are those human resources to be organized.

Conceptual Schema. The key concepts in the Personnel Viewpoint that can be used as model elements in the architecture views and the relationships between these concepts are illustrated in the conceptual schema shown in Figure 4:39. These ***key concepts*** are highlighted in italics within the Narrative and some of the less obvious concepts are listed with the associated UAF meaning of that concept. Detailed definition of the entities and relationships shown in the conceptual schema are provided in the UAFML specification document.



Note 1 – can also be a Resource Performer, Resource or Operational Signal or Geopolitical Extent

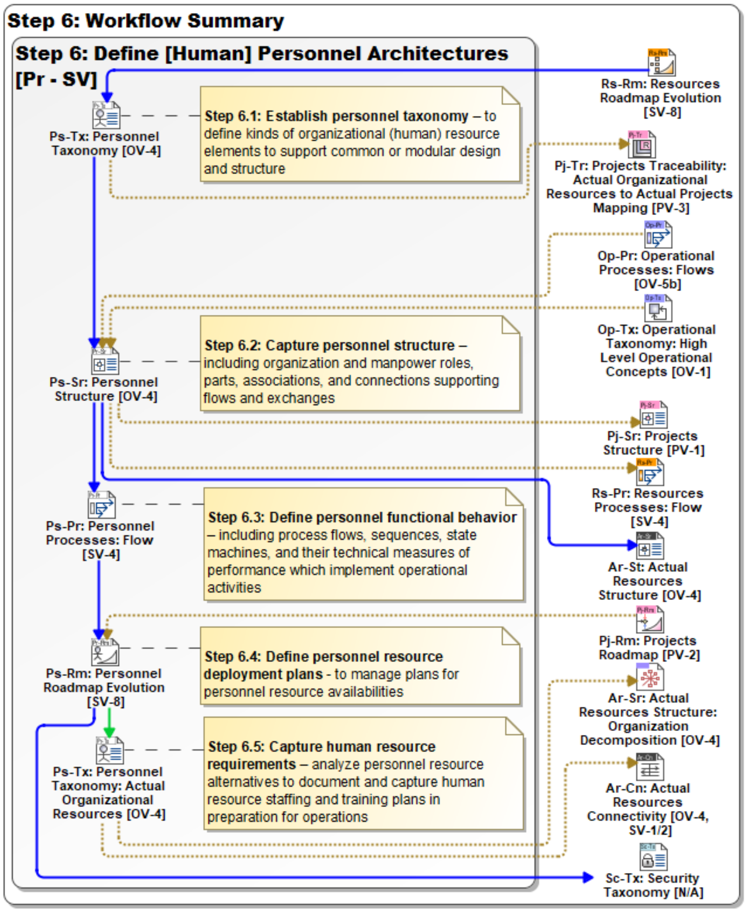
Figure 4:39 - Conceptual Schema for Personnel Architectures

The operational architecture concepts are shown on the left in blue and the resource and personnel architecture concepts are shown on the right in orange and white, respectively. ***Resource performers*** and the organizational resource performer ***functions*** performed by those performers will "implement" the ***operational agents*** and ***operational activities*** defined in the Operational Architecture.

Likewise, personnel (or organizational) ***resource roles, resource connectors, resource exchange items, resource exchanges,*** and ***resource interfaces*** will implement the analogous things in the ***operational architecture***. Decisions about which personnel resource elements (and what kinds of these elements) will implement operational elements are made in this Step 6. Resource mitigations can be devised to deal with personnel resource security risks, and these will be addressed in Step 7.

The personnel architectures are in contrast to the resource architectures in Step 5 because they involve human resources. This is where the tradeoffs dealing with how much automation of formerly manual work will be now performed by machines in the resource architecture. There will also be questions about when it is appropriate to have "man in the loop" versus "man on the loop" versus totally autonomous operations with no human involvement in the functions being performed by physical resources. The personnel architecture also deals with the assigned roles and responsibilities of human assets, how they are organized, and the necessary knowledge and skill competencies to properly fulfill their roles and responsibilities.

This Guide is intended to be used in conjunction with the UAF Sample Problem that defines architecture views for a Search and Rescue Mission. UAF Specification documents (including the Sample Problem) can be downloaded from the OMG webpage: www.omg.org/spec/UAF/About-UAF/.


Step 6: Workflow Summary
Step 6: Define [Human] Personnel Architectures [Pr - SV]
Ps-Tx: Personnel Taxonomy [OV-4]
Ps-Sr: Personnel Structure [OV-4]
Ps-Pr: Personnel Processes: Flow [SV-4]
Ps-Rm: Personnel Roadmap Evolution [SV-8]
Ps-Tx: Personnel Taxonomy: Actual Organizational Resources [OV-4]
Rs-Rm: Resources Roadmap Evolution [SV-8]
Pj-Tr: Projects Traceability: Actual Organizational Resources to Actual Projects Mapping [PV-3]
Op-Pr: Operational Processes: Flows [OV-5b]
Op-Tx: Operational Taxonomy: High Level Operational Concepts [OV-1]
Pj-Sr: Projects Structure [PV-1]
Rs-Pr: Resources Processes: Flow [SV-4]
Ar-St: Actual Resources Structure [OV-4]
Pj-Rm: Projects Roadmap [PV-2]
Ar-Sr: Actual Resources Structure: Organization Decomposition [OV-4]
Ar-Cn: Actual Resources Connectivity [OV-4, SV-1/2]
Sc-Tx: Security Taxonomy [N/A]

Step 6.1: Establish personnel taxonomy – to define kinds of organizational (human) resource elements to support common or modular design and structure

Step 6.2: Capture personnel structure – including organization and manpower roles, parts, associations, and connections supporting flows and exchanges

Step 6.3: Define personnel functional behavior – including process flows, sequences, state machines, and their technical measures of performance which implement operational activities

Step 6.4: Define personnel resource deployment plans - to manage plans for personnel resource availabilities

Step 6.5: Capture human resource requirements – analyze personnel resource alternatives to document and capture human resource staffing and training plans in preparation for operations

Figure 4:40 - Workflow Summary for Step 6: Define [Human] Personnel Architectures

Step 6.0 – Define [Human] Personnel Architectures – The main entry criterion for this Step is bringing forward the resource architectures from Step 5 with associated human ***resource interfaces*** and their operational ***implementation*** context. Both the operations and resources architectures are now examined for personnel implementations using various organizational ***resource performers*** including, for example, ***organizations, posts, persons***, and ***responsibilities***.

These organizational resource performers along with their associated exchanges and functions will form alternatives that clarify the role of Human Factors (HF) that are compared in trade-offs to support ***implementation*** decisions for Human Factors Integration (HFI). When examining the organizational resource performer options, the use of ***organizational resources*** is considered in this Step for specific trade-offs on how ***organizations***, ***posts***, and other human resources will be utilized, which could involve setting desired levels of automation, especially with regard to deciding when and where to have humans in-the-loop, on-the-loop or out-of-the-loop during operational activity execution.

- **Organizational Resource** – an abstract type for **Organization**, **Person**, **Post** and **Responsibility** (Note: these types also include **Project**, **Sub Organization**, and **Post Role** and are typically human resources that can interact with other personnel or resources and perform personnel functions)
- **Implements** – a tuple that defines how an element in the upper layer of abstraction is implemented by a semantically equivalent element (for example tracing the Functions to the Operational Activities) in the lower level of abstraction (Note: in this case a relationship between **Organizational Resource** elements that outfit or equip operational elements

When examining the organizational resource performer options, the use of ***organizational resources*** is considered in this Step for specific trade-offs on how ***organizations***, ***posts***, and other human resources will be utilized, which could involve setting desired levels of automation, especially with regarding deciding when and where to have humans in-the-loop, on-the-loop, or out-of-the-loop during operational activity execution.

The personnel architecture defines personnel *behavior* elements (e.g., processes, states, sequences) and allocates these to organizational resource *structure* elements (e.g., organizations and posts, to be used in roles, connectors, exchange items, exchanges, ports, and interfaces). The behavior elements define personnel ***functions*** that implement ***operational activities*** or amplify or clarify resource functions that will in turn help achieve enterprise goals.

Workflow Summary. A summary level view of the five steps involved in Step 6 workflow is shown above. The flowlines represent how one architecture view "influences" another architecture view. These lines do not represent a particular sequence of process activity execution or imply information exchanges. These should be thought of as influence diagrams rather than process diagrams. These architecture views (e.g., views, diagrams, tables) can be mapped to the architecture processes used in your organization and can be incorporated into an architecture modeling methodology.

The second-level steps in the Step 6 workflow are listed below along with the UAF views (and corresponding [DoDAF views] where applicable) associated with those steps. A complete list of the detailed steps at the third level with their corresponding views is provided at the end of this section.

169	**Step 6: Define [Human] Personnel Architectures [Ps - SV]**	**Views**
170	**Step 6.1: Establish personnel taxonomy** – to define kinds of organizational (human) resource elements to support common or modular design and structure	Ps-Tx: Personnel Taxonomy [OV-4]
177	**Step 6.2: Capture personnel structure** – including organization and manpower roles, parts, associations, and connections supporting flows and exchanges	Ps-Sr: Personnel Structure [OV-4]
183	**Step 6.3: Define personnel functional behavior** – including process flows, sequences, state machines, and their technical measures of performance which implement operational activities	Ps-Pr: Personnel Processes: *Flow* [SV-4]
189	**Step 6.4: Define personnel resource deployment plans** - to manage plans for personnel resource availabilities	Ps-Rm: Personnel Roadmap Evolution [SV-8]
191	**Step 6.5: Capture human resource requirements** – analyze personnel resource alternatives to document and capture human resource staffing and training plans in preparation for operations	Ps-Tx: Personnel Taxonomy: *Actual Organizational Resources* [OV-4]

4.6.1 Personnel Taxonomy

Step 6.1 – Establish personnel taxonomy – A set of organizational resources are described, including any that have been preliminarily identified from Step 3.2, where organizational resources are needed or anticipated that will ***implement operational agents***. These organizational resource performers are accounted for in a ***personnel taxonomy***, and may include ***organizations***, ***persons***, ***posts***, or ***responsibilities***. This captures the understanding of ***generalizations***, particularly when a reference personnel architecture is setting contextual guidance, organizational resource categories are established, or particular human resource plans are already known.

- **Organization** – a group of organizational resources (**Persons**, **Posts**, **Organizations** and **Responsibilities**) that are associated for a purpose
- **Person** – a type of human being used to define the characteristics that need to be described for **Actual Persons** (e.g., properties such as address, telephone number, nationality, etc.)
- **Post** – a type of job title or position that a **Person** can fill (e.g., Lawyer, Solution Architect, Machine Operator or Chief Executive Officer)
- **Responsibility** – a type of duty required of a **Post**, **Person,** or **Organization**

Organizations are elements intended to describe composite structures of human organizations that include other organizations, sub-organizations, and individual persons and posts along with their responsibilities. The ***resource architecture*** elements and their sub-type ***capability configuration***, and ***system*** elements should be checked and verified for their organizational resource components. When these ***resource architecture*** elements do not appear to comprise personnel, thought should be given to possibly refactoring them to more simple ***resource artifacts*** since systems can and usually do include human resources as components of the system.

The total environment developed in Step 2.1 expected for deploying capabilities in terms of ***conditions***, ***locations***, ***environments***, including kinds of environments and ***kinds*** of locations, as well as ***geo-political*** factors is brought forward to assess effects on organizational resources, and becomes an overarching conditional context for organizational resource parameters.

Competencies are listed and applied to all organizational resource performers who require a particular knowledge, skill or aptitude required to be competent to conduct actions in roles as well as functions. Additionally, organizational resource performers are subject to the same ***resource constraints*** and their ***rules*** brought forward from Step 5.1, which must be adjusted to account for human factors.

- **Competence** – a set of abilities defined by knowledge, skills and aptitude (Note: which typically describes human conduct provided to, or required by, their assigned role or expected participation in **Organizations** and **Functions**)
- **Resource Constraint** – a rule governing the structural or functional aspects of an implementation

An organizational resource technology ***forecast*** is assembled to show when key organizational resources are expected for periods applicable to human staffing needs for the operational architecture.

- **Forecast** – a dependency relationship that specifies a transition from one Resource Performer, **Standard**, or **Competence** to another future one, related to an Actual Enterprise Phase to give it a temporal context

A review is conducted to ensure all organizational resources have been covered or addressed within the ***personnel taxonomy table***.

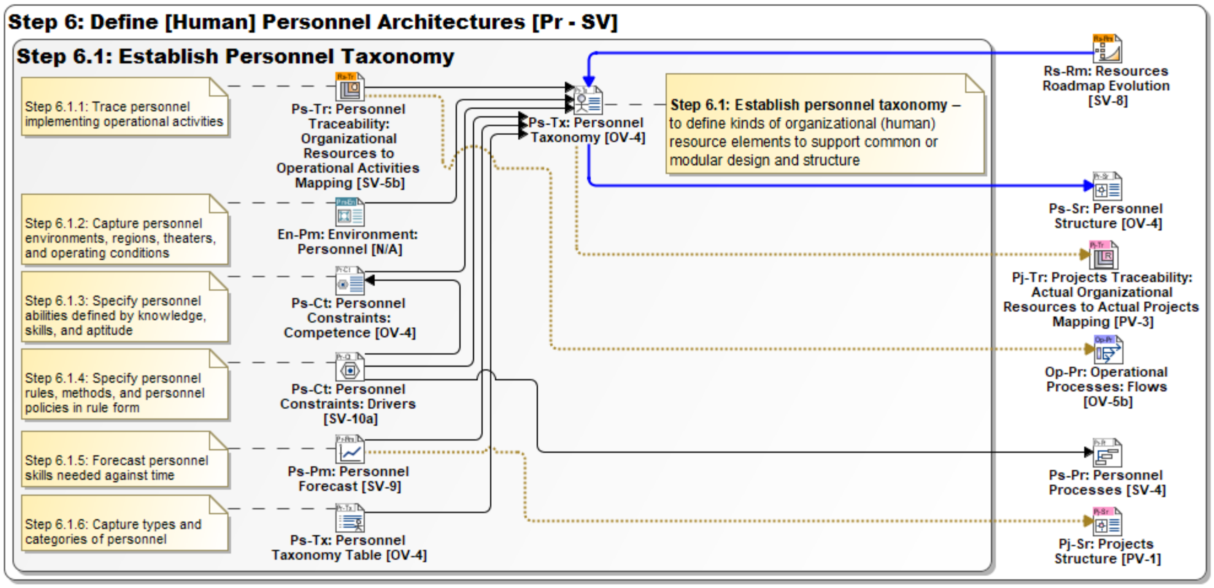

Step 6: Define [Human] Personnel Architectures [Pr - SV]
Step 6.1: Establish Personnel Taxonomy
Step 6.1.1: Trace personnel implementing operational activities
Ps-Tr: Personnel Traceability: Organizational Resources to Operational Activities Mapping [SV-5b]
Step 6.1.2: Capture personnel environments, regions, theaters, and operating conditions
En-Pm: Environment: Personnel [N/A]
Step 6.1.3: Specify personnel abilities defined by knowledge, skills, and aptitude
Ps-Ct: Personnel Constraints: Competence [OV-4]
Step 6.1.4: Specify personnel rules, methods, and personnel policies in rule form
Ps-Ct: Personnel Constraints: Drivers [SV-10a]
Step 6.1.5: Forecast personnel skills needed against time
Ps-Pm: Personnel Forecast [SV-9]
Step 6.1.6: Capture types and categories of personnel
Ps-Tx: Personnel Taxonomy Table [OV-4]
Ps-Tx: Personnel Taxonomy [OV-4]
Step 6.1: Establish personnel taxonomy – to define kinds of organizational (human) resource elements to support common or modular design and structure
Rs-Rm: Resources Roadmap Evolution [SV-8]
Ps-Sr: Personnel Structure [OV-4]
Pj-Tr: Projects Traceability: Actual Organizational Resources to Actual Projects Mapping [PV-3]
Op-Pr: Operational Processes: Flows [OV-5b]
Ps-Pr: Personnel Processes [SV-4]
Pj-Sr: Projects Structure [PV-1]

Figure 4:41 - Step 6.1: Establish Personnel Taxonomy

4.6.2 Personnel Structure

Step 6.2 – Capture Personnel Structure – Internal structure of organizational resources within the taxonomy are developed with internal features and characteristics, including ***resource roles*** they will have as part of a greater ***organization***. One organizational resource may perform different ***resource roles*** and one resource role may be performed by different ***organizational resources***. Examination of this is necessary to scope and perform assessment of the trade space and an analysis of alternatives, including differences between composition and aggregation in organizational resource structure. Formal personnel ***resource interfaces*** may be defined and declared for interface points of any organizational resource performer port, which implement service or operational interfaces.

- **Resource Role** – usage of a Resource Performer in the context of another Resource Performer creating a whole-part relationship (Note: typical enumerated kinds of personnel roles may include Human Resource, **Sub Organization**, **Post Role**, or Responsibility Role)
- **Resource Interface** – a declaration that specifies a contract between the Resource Performers it is related to and any other Resource Performers it can interact with. It is also intended to be an implementation of a specification of an Interface in the Business and/or Service layer.

Organizational resources are associated with each other to identify organizational resource relationships which will support organizational ***resource exchanges*** and ***resource connections*** to **control** physical resources or ***command*** organizational resources.

- **Control** – a type of **Resource Exchange** that asserts that one Physical Resource controls another Physical Resource (i.e., the driver of a vehicle controlling the vehicle speed or direction)
- **Command** – a type of **Resource Exchange** that asserts that one **Organizational Resource** commands another
- **Resource Connector** – a channel for exchange between two **Resource Roles**

An organizational resource ***logical information model*** is created to define the ***resource information*** elements which are exchanged. ***Commands*** and ***controls*** are placed on all associations with groupings of ***resource exchange items*** in accordance with actions or sequences of the organizational resources. Multiple associations may exist between organizational resources which represent different types of exchanges in terms of time, sequence, interface definition, or kinds of items that are exchanged (which for the personnel architecture consists of ***resource information*** elements).

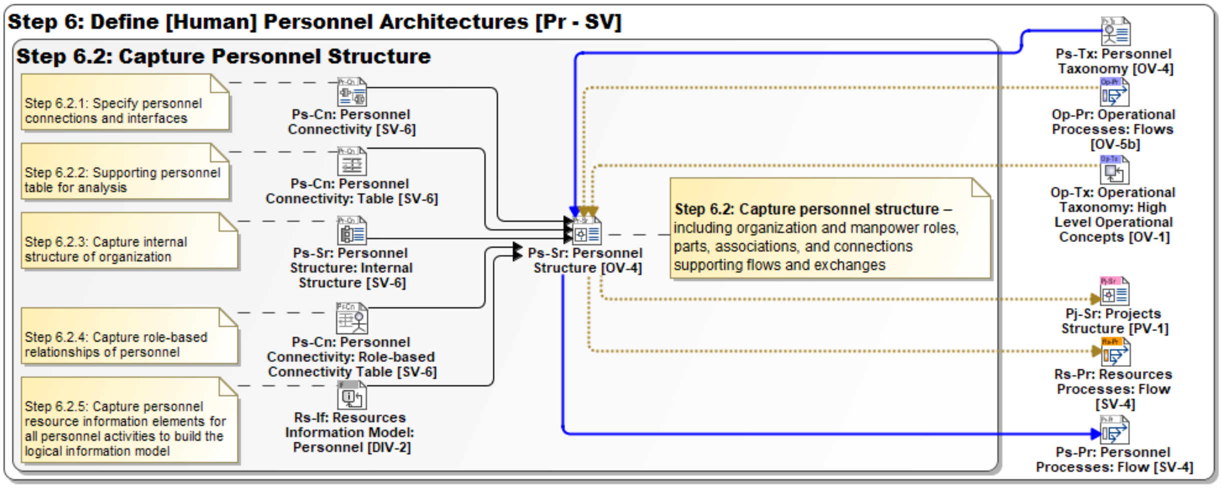

Step 6: Define [Human] Personnel Architectures [Pr - SV]
Step 6.2: Capture Personnel Structure
Step 6.2.1: Specify personnel connections and interfaces
Ps-Cn: Personnel Connectivity [SV-6]
Step 6.2.2: Supporting personnel table for analysis
Ps-Cn: Personnel Connectivity: Table [SV-6]
Step 6.2.3: Capture internal structure of organization
Ps-Sr: Personnel Structure: Internal Structure [SV-6]
Step 6.2.4: Capture role-based relationships of personnel
Ps-Cn: Personnel Connectivity: Role-based Connectivity Table [SV-6]
Step 6.2.5: Capture personnel resource information elements for all personnel activities to build the logical information model
Rs-If: Resources Information Model: Personnel [DIV-2]
Ps-Sr: Personnel Structure [OV-4]
Step 6.2: Capture personnel structure – including organization and manpower roles, parts, associations, and connections supporting flows and exchanges
Ps-Tx: Personnel Taxonomy [OV-4]
Op-Pr: Operational Processes: Flows [OV-5b]
Op-Tx: Operational Taxonomy: High Level Operational Concepts [OV-1]
Pj-Sr: Projects Structure [PV-1]
Rs-Pr: Resources Processes: Flow [SV-4]
Ps-Pr: Personnel Processes: Flow [SV-4]

Figure 4:42 - Step 6.2: Capture Personnel Structure

4.6.3 Personnel Functional Behavior

Step 6.3 – Define Personnel Functional Behavior – Process flow diagrams are constructed for all personnel ***functions*** including personnel ***function actions*** classified by personnel ***functions*** and process control-flow mechanisms such as decision nodes and forks. When ***organizational resources*** are already known or defined in Step 6.1, they are assigned to swim-lanes and the actions they are capable to perform. Otherwise, ***function actions*** are grouped into swim-lanes to create ***organizational resources*** which must be structured in Step 6.1 – 6.2.

The process flows are refined, when needed, with ***resource state descriptions*** for organizational resources and sequenced timelines of ***resource messages*** and ***resource methods***. ***Resource connections*** are added to the process flow to realize the ***controls*** or ***commands*** (which for the personnel architecture may consist of ***resource information*** elements). State descriptions and sequenced timelines may now be developed to supplement or refine the item exchanges.

- **Resource Message** – a message for use in a Resource Event-Trace which carries any of the subtypes of **Resource Exchange**
- **Resource Method** – a behavioral feature of a **Resource Performer** whose behavior is specified in a **Resource Function**

Human performance ***measures*** are defined for the resource performers and their ***functions***. Human performance ***measures*** (HPMs) may include parametric diagrams when needed. Measures of overall personnel functions and organizational resources should demonstrate satisfaction of operational implementations either directly, or indirectly through examination or correlation of resource HPMs with the operational MOPs. When personnel architecture is structured within any type of ***resource architecture***, the HPMs should factor in as part of the ***resource architecture*** TPMs and then correlated to the operational MOPs.

- **Measurement** – a property of an element representing something in the physical world, expressed in amounts of a unit of measure

When developing a business architecture, the personnel behavior functions and sequenced timelines, and their implementation of operational activities and service functions, are used to capture the business aspects of the enterprise. Business Process Model and Notation (BPMN)[7] is an alternative diagram for these views in the UAFML.

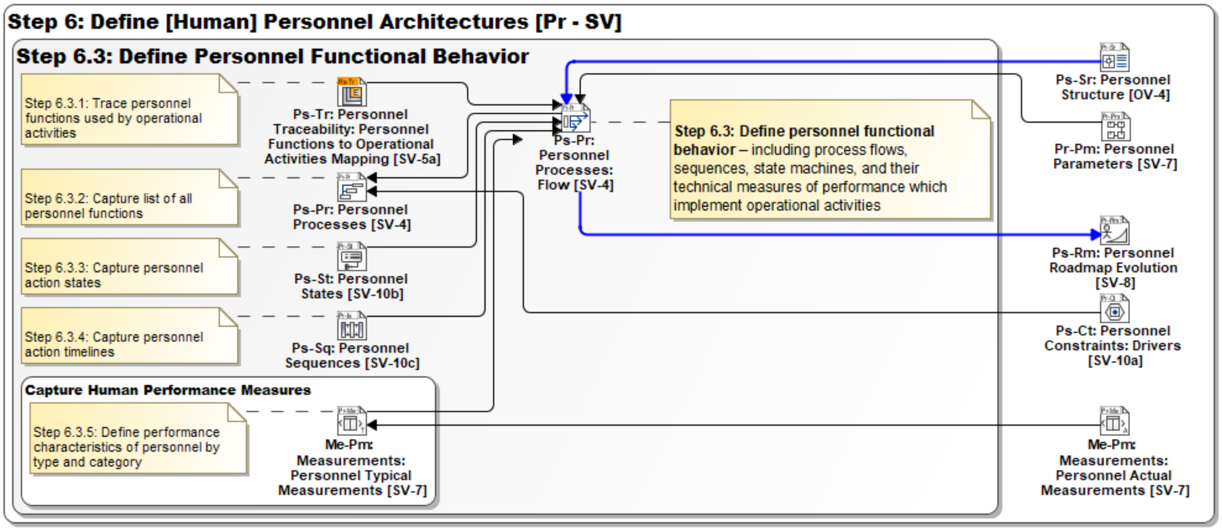

Step 6: Define [Human] Personnel Architectures [Pr - SV]
Step 6.3: Define Personnel Functional Behavior
Step 6.3.1: Trace personnel functions used by operational activities
Ps-Tr: Personnel Traceability: Personnel Functions to Operational Activities Mapping [SV-5a]
Step 6.3.2: Capture list of all personnel functions
Ps-Pr: Personnel Processes [SV-4]
Step 6.3.3: Capture personnel action states
Ps-St: Personnel States [SV-10b]
Step 6.3.4: Capture personnel action timelines
Ps-Sq: Personnel Sequences [SV-10c]
Capture Human Performance Measures
Step 6.3.5: Define performance characteristics of personnel by type and category
Me-Pm: Measurements: Personnel Typical Measurements [SV-7]
Ps-Pr: Personnel Processes: Flow [SV-4]
Step 6.3: Define personnel functional behavior – including process flows, sequences, state machines, and their technical measures of performance which implement operational activities
Ps-Sr: Personnel Structure [OV-4]
Pr-Pm: Personnel Parameters [SV-7]
Ps-Rm: Personnel Roadmap Evolution [SV-8]
Ps-Ct: Personnel Constraints: Drivers [SV-10a]
Me-Pm: Measurements: Personnel Actual Measurements [SV-7]

Figure 4:43 - Step 6.3: Define Personnel Functional Behavior

A review is conducted to ensure all relevant ***operational activities*** and their ***performers*** have been ***implemented*** by personnel ***functions***, and their performers, or by integrated personnel and resource ***functions***, and their composite or aggregated resource architectural performer. Additionally, personnel functional structural relationships are examined through intervening personnel ***function actions***, to check for redesign or simplification, for adjustments and changes in levels of automation or human control, as well as possible duplication in implementation coverage of operational activities. Personnel functions may be viewed either in complex meta-chain maps or in simple diagrams to ensure a complete library of personnel ***functions*** is established and understood. This library may be arranged by personnel function groupings, organizational resources capable to perform them, or some other useful organization.

4.6.4 Personnel Resource Deployment Plans

Step 6.4 – Define Personnel Resource Deployment Plans – When human resources are planned for future availability, a ***personnel roadmap*** is assembled to show when organizational resources will be ***released*** for employment and staffing, will be ***withdrawn***, or will be changed over time. This roadmap allows human resource planners to understand when resource performers for implementation are deployed, in service, out of service, or no longer used. Actual milestones are created for service specifications to denote their availability on a ***timeline***.

- **Version Released at Milestone** – an actual milestone category showing a version of an element to be released (Note: in this case an Organizational Resource)
- **Version Withdrawn at Milestone** – an actual milestone category showing a version of an element to be withdrawn (Note: in this case an Organizational Resource)

[7] ***ISO 19510:2013 – Information technology — Object Management Group Business Process Model and Notation*** establishes a set of notations and semantics for collaboration, process, and choreography diagrams to communicate process information to other business users, process implementers, customers and suppliers (see also OMG standard BPMN 2.0.2)

Organizational resource staffing is governed by ***version succession*** of ***versions of configurations*** within ***whole life configurations*** for organizational resources. Versions of configuration are structured within whole life configurations to organize the staffing of organizational resources. Various organizational resources ***succeed*** others as versions of the organizational resources are changed.

- **Whole Life Configuration** – a set of Versioned Elements (Note: in this case Personnel)
- **Version of Configuration** – a property of a **Whole Life Configuration**, used in version control of a Versioned Element. It asserts that a Versioned Element is a version of a **Whole Life Configuration**. (Note: in this case a particular version of an Organizational Resource)
- **Version Succession** – a tuple between two **Version Of Configurations** that denotes that one **Version Of Configuration** follows from another

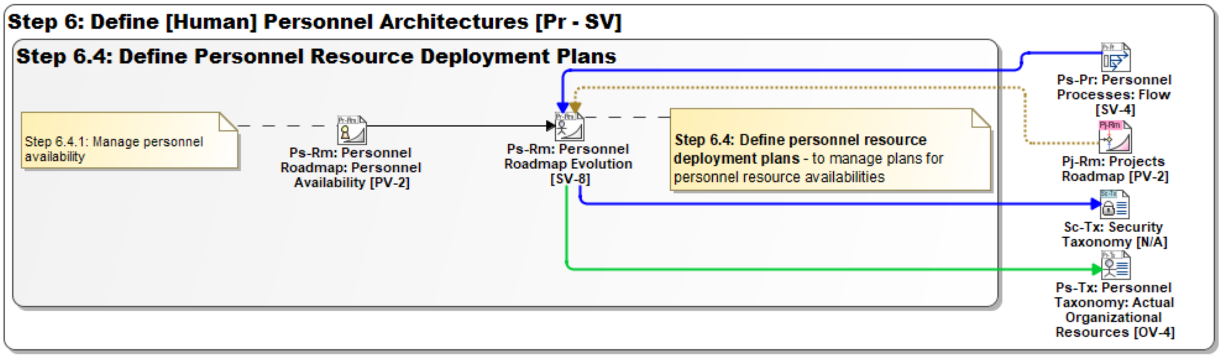

Step 6: Define [Human] Personnel Architectures [Pr - SV]
Step 6.4: Define Personnel Resource Deployment Plans
Step 6.4.1: Manage personnel availability
Ps-Rm: Personnel Roadmap: Personnel Availability [PV-2]
Ps-Rm: Personnel Roadmap Evolution [SV-8]
Step 6.4: Define personnel resource deployment plans - to manage plans for personnel resource availabilities
Ps-Pr: Personnel Processes: Flow [SV-4]
Pj-Rm: Projects Roadmap [PV-2]
Sc-Tx: Security Taxonomy [N/A]
Ps-Tx: Personnel Taxonomy: Actual Organizational Resources [OV-4]

Figure 4:44 - Step 6.4: Define Personnel Resource Deployment Plans

4.6.5 Human Resource Analysis and Requirements

Step 6.5 – Capture Human Resource Requirements – Once a personnel architecture has been captured and defined, organization of staffing possibilities can begin. Personnel structure is analyzed to bundle or organize staffing, or to examine variations and specializations in human resource utilization. Impact analysis is conducted on organizational resource and organizational resource role implementations. Typically, feasibility, cost and other factors may be tied to the organizational resource elements to drive other kinds of analysis.

- **Actual Organizational Resource** – an abstract element for an **Actual Organization**, **Actual Person** or **Actual Post** (Note: this is an instance of an **Organizational Resource** in the real world)
- **Actual Person** – an individual human being (Note: this is an instance of a **Person** in the real world)
- **Actual Post** – an actual, specific post, an instance of a **Post** "type" - e.g., "President of the United States of America" where the **Post** would be president (Note: this is an instance of a **Post** in the real world)
- **Fills Post** – a tuple that asserts that an **Actual Person** fills an **Actual Post**
- **Actual Responsibility** – an actual duty required of a **Person** or **Organization** (Note: this is an instance of a **Responsibility** in the real world)

An entire personnel architecture with all of its elements (agents, behaviors, etc.) may be set as one option in an analysis of alternatives between others. Each alternative to be examined can be represented by a separate personnel architecture. Typical human performance measures (HPMs) from Step 6.3 may be used to generate ***actual measurements*** to either capture analysis of these alternatives or records of analysis for human performers over time. Each actual measurement table is assigned start and end dates of their actual, estimated, or required points of existence, and are used to track validation, satisfaction, and realization of human performance changes over time.

A ***risk*** analysis assesses the impact of events that may ***affect*** organizational resource assets. Measures and ***actual measurements*** of these ***risks*** are included with the organizational resource measurement analysis.

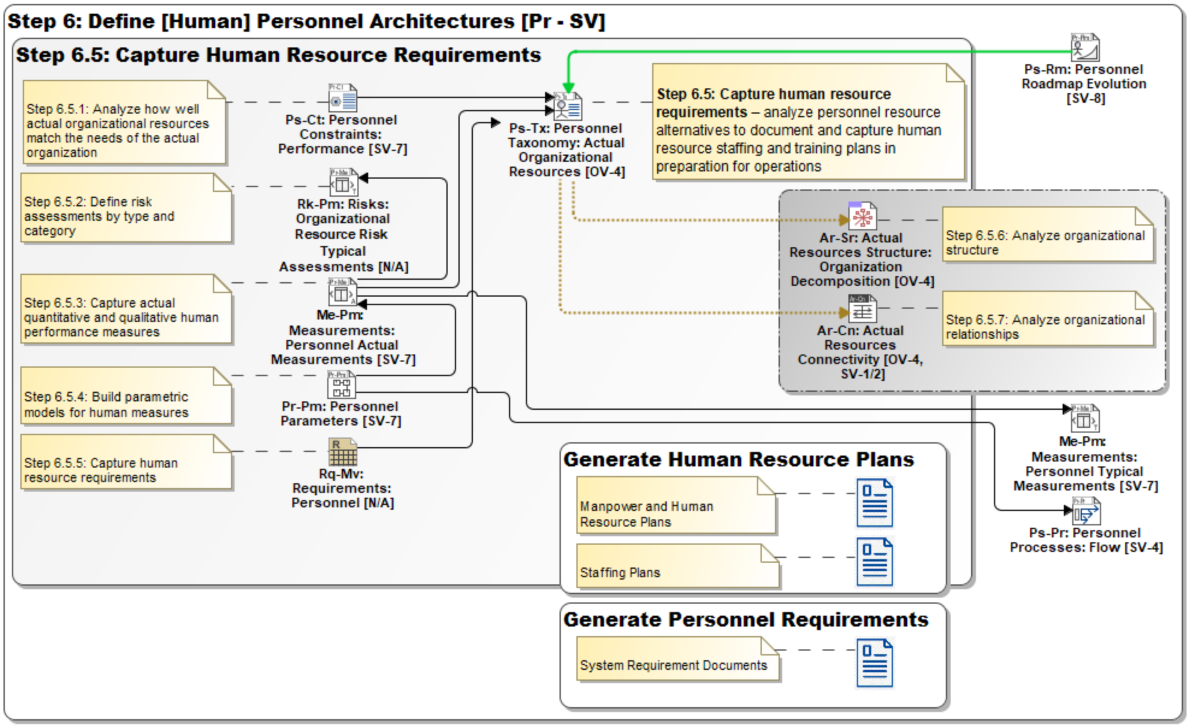

Step 6: Define [Human] Personnel Architectures [Pr - SV]
Step 6.5: Capture Human Resource Requirements
Step 6.5.1: Analyze how well actual organizational resources match the needs of the actual organization
Step 6.5.2: Define risk assessments by type and category
Step 6.5.3: Capture actual quantitative and qualitative human performance measures
Step 6.5.4: Build parametric models for human measures
Step 6.5.5: Capture human resource requirements
Ps-Ct: Personnel Constraints: Performance [SV-7]
Rk-Pm: Risks: Organizational Resource Risk Typical Assessments [N/A]
Me-Pm: Measurements: Personnel Actual Measurements [SV-7]
Pr-Pm: Personnel Parameters [SV-7]
Rq-Mv: Requirements: Personnel [N/A]
Ps-Tx: Personnel Taxonomy: Actual Organizational Resources [OV-4]
Step 6.5: Capture human resource requirements – analyze personnel resource alternatives to document and capture human resource staffing and training plans in preparation for operations
Ps-Rm: Personnel Roadmap Evolution [SV-8]
Ar-Sr: Actual Resources Structure: Organization Decomposition [OV-4]
Ar-Cn: Actual Resources Connectivity [OV-4, SV-1/2]
Step 6.5.6: Analyze organizational structure
Step 6.5.7: Analyze organizational relationships
Me-Pm: Measurements: Personnel Typical Measurements [SV-7]
Ps-Pr: Personnel Processes: Flow [SV-4]
Generate Human Resource Plans
Manpower and Human Resource Plans
Staffing Plans
Generate Personnel Requirements
System Requirement Documents

Figure 4:45 - Step 6.5: Capture Human Resource Requirements

- **Actual Measurement** – an actual value that is applied to a Measurement (Note: a measurement may be one of three kinds: actual, required, or estimate, and may have an associated start and/or end date)
- **Required** – an enumerated type of an actual measurement kind which is based on a required value

If the personnel architecture is used as a reference architecture to evaluate bids for contracts from human resource providers, ***actual measurements*** may form the basis of evaluation for bids received on parts or for all of the personnel architecture.

When needed, staffing plans, manpower plans, and human resources plans can be developed based on the models and views generated during this Step 6. Ideally these documents are automatically generated from the model itself using reporting scripts and model queries. Additionally, the architectural description itself may be published or shared with human resource planners or bidders to communicate staffing plans, requests for information, and requests for proposals.

Additionally, a set of service ***requirements*** can be generated based on the elements within a Personnel Architecture, which are intended to be personnel implementation agnostic. Requirements tracing from the Personnel Architecture can be satisfied by personnel solutions.

Requirements are developed and related to architectural elements relevant to the generation of the requirement. These are related using ***trace, verify, satisfy***, or ***refine*** relationships for linking to relevant elements in the architecture.

- **Requirement** – a statement that identifies a system, product or process characteristic or constraint, which is unambiguous, clear, unique, consistent, stand-alone (not grouped), and verifiable, and is deemed necessary for stakeholder acceptability (INCOSE 2010)
- **Refine** – a relationship from an architectural element which refines a text-based requirement

4.6.6 Architecture View Summary for Step 6

The view specifications in UAF for this viewpoint are outlined here:

	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Personnel Ps	Requirements Rq-Mv	Personnel Taxonomy Ps-Tx	Personnel Structure Ps-Sr	Personnel Connectivity Ps-Cn	Personnel Processes Ps-Pr	Personnel States Ps-St	Personnel Sequences Ps-Sq	Resources Information Model (Rs-If)	Envmt, Msmts, Risks (En-Pm, Me-Pm, Rk-Pm)	Competence, Drivers, Performance Ps-Ct-C,-D,-P	Availability, Evolution, Forecast PS-Rm-A,-E,-F	Personnel Traceability Ps-Tr

A summary of the lower-level steps and UAF views is shown below. The architecture views produced by each step are identified by the UAF grid designator Aa-Bb where Aa represents the Viewpoint row in the grid and Bb represents the Aspect column in the grid. DoDAF view designations are also shown in [brackets], when applicable. The name of the architecture view matches the name of the view specification in the UAFML. In some cases, a subtitle is provided to identify what kind of information is provided in an instance of that UAF view. There will be some architecture views that are not part of the UAF specification since you will sometimes need fit-for-purpose (i.e., custom) views that capture the necessary architectural information.

169	**Step 6: Define [Human] Personnel Architectures [Ps - SV]**	**Views**
170	**Step 6.1: Establish personnel taxonomy** – to define kinds of organizational (human) resource elements to support common or modular design and structure	Ps-Tx: Personnel Taxonomy [OV-4]
171	Step 6.1.1: Trace personnel implementing operational activities	Ps-Tr: Personnel Traceability: *Organizational Resources to Operational Activities Mapping* [SV-5b]
172	Step 6.1.2: Capture personnel environments, regions, theaters, and operating conditions	En-Pm: Environment: *Personnel* [N/A]
173	Step 6.1.3: Specify personnel abilities defined by knowledge, skills, and aptitude	Ps-Ct: Personnel Constraints: *Competence* [OV-4]
174	Step 6.1.4: Specify personnel rules, methods, and personnel policies in rule form	Ps-Ct: Personnel Constraints: *Drivers* [SV-10a]
175	Step 6.1.5: Forecast personnel skills needed against time	Ps-Pm: Personnel Forecast [SV-9]
176	Step 6.1.6: Capture types and categories of personnel	Ps-Tx: Personnel Taxonomy Table [OV-4]
177	**Step 6.2: Capture personnel structure** – including organization and manpower roles, parts, associations, and connections supporting flows and exchanges	Ps-Sr: Personnel Structure [OV-4]
178	Step 6.2.1: Specify personnel connections and interfaces	Ps-Cn: Personnel Connectivity [SV-6]
179	Step 6.2.2: Supporting personnel table for analysis	Ps-Cn: Personnel Connectivity Table [SV-6]
180	Step 6.2.3: Capture internal structure of organization	Ps-Sr: Personnel Structure: Internal Structure [SV-6]
181	Step 6.2.4: Capture role-based relationships of personnel	Ps-Cn: Personnel Role-based Connectivity Table [SV-6]

182	Step 6.2.5: Capture personnel resource information elements for all personnel activities to build the logical information model	Rs-If: Resources Information Model: *Personnel* [DIV-2]
183	**Step 6.3: Define personnel functional behavior** – including process flows, sequences, state machines, and their technical measures of performance which implement operational activities	Ps-Pr: Personnel Processes: *Flow* [SV-4]
184	Step 6.3.1: Trace personnel functions used by operational activities	Ps-Tr: Personnel Traceability: *Personnel Functions to Operational Activities Mapping* [SV-5a]
185	Step 6.3.2: Capture list of all personnel functions	Ps-Pr: Personnel Processes [SV-4]
186	Step 6.3.3: Capture personnel action states	Ps-St: Personnel States [SV-10b]
187	Step 6.3.4: Capture personnel action timelines	Ps-Sq: Personnel Sequences [SV-10c]
188	Step 6.3.5: Define performance characteristics of personnel by type and category	Me-Pm: Measurements: *Personnel Typical Measurements* [SV-7]
189	**Step 6.4: Define personnel resource deployment plans** - to manage plans for personnel resource availabilities	Ps-Rm: Personnel Roadmap Evolution [SV-8]
190	Step 6.4.1: Manage personnel availability	Ps-Rm: Personnel Roadmap: *Personnel Availability* [PV-2]
191	**Step 6.5: Capture human resource requirements** – analyze personnel resource alternatives to document and capture human resource staffing and training plans in preparation for operations	Ps-Tx: Personnel Taxonomy: *Actual Organizational Resources* [OV-4]
192	Step 6.5.1: Analyze how well actual organizational resources match the needs of the actual organization	Ps-Ct: Personnel Constraints: *Performance* [SV-7]
193	Step 6.5.2: Define risk assessments by type and category	Rk-Pm: Risks: *Organizational Resource Risk Typical Assessments* [N/A]
194	Step 6.5.3: Capture actual quantitative and qualitative human performance measures	Me-Pm: Measurements: *Personnel Actual Measurements* [SV-7]
195	Step 6.5.4: Build parametric models for human measures	Pm: Parameters: *Personnel Parameters* [SV-7]
196	Step 6.5.5: Capture human resource requirements	Rq-Mv: Requirements: *Personnel* [N/A]
197	Step 6.5.6: Analyze organizational structure	Ar-Sr: Actual Resources Structure: *Organization Decomposition* [OV-4]
198	Step 6.5.7: Analyze organizational relationships	Ar-Cn: Actual Resources Connectivity [OV-4]

STEP 7 – Security Architectures

4.7 Step 7 – Security Architectures

Purpose. The purpose of this step is to illustrate security assets, security constraints, security controls, security control families and the measures required to address specific security concerns. Key stakeholders for this step are Security Architects, Security Engineers, Systems Engineers and Operational Architects. Their concerns are mainly about how to address the security constraints and information assurance attributes that exist on exchanges between resources and operational performers.

Conceptual Schema. The key concepts in the Security Viewpoint that can be used as model elements in the architecture views and the relationships between these concepts are illustrated in the conceptual schema shown in Figure 4:46. These ***key concepts*** are highlighted in italics within the Narrative and some of the less obvious concepts are listed with the associated UAF meaning of that concept. Detailed definition of the entities and relationships shown in the conceptual schema are provided in the UAFML specification document.

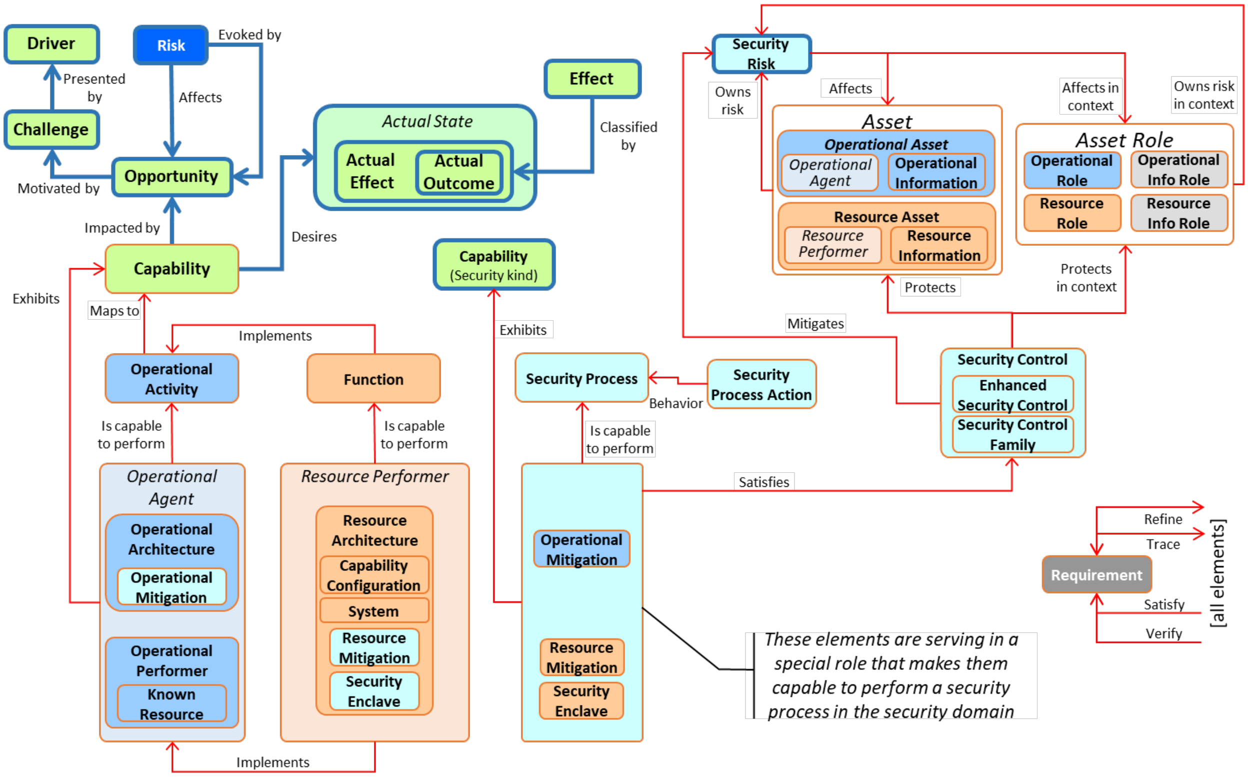


Figure 4:46 - Conceptual Schema for Security Architectures

The operational architecture concepts are shown on the left in blue and the resource and personnel architecture concepts are shown on the right in orange and white, respectively. Security architecture concepts are shown on the right in light blue. Security resource performers (operational and resource mitigations, and security enclaves) and the corresponding operational and resource performer ***functions*** performed by those security resource performers will perform a ***security process*** that implements ***security controls*** for protecting enterprise ***assets*** to ***mitigate*** expected ***risks***.

Decisions about which security resource elements (and what kinds of these elements) will ***implement*** security process actions are made in this Step 7. ***Resource mitigations*** will "implement" the ***operational mitigations*** and ***security processes*** may exhibit ***capabilities***.

Security can often be addressed by things other than physical and human resources. For example, it is worth considering how other dimensions of the solution space can be put in place, such as doctrine, organization, training, materiel (other than regular resources, systems, or other artifacts), leadership and education, personnel, and facilities (DOTMLPF). Alternatively, you can consider other categories associated with the "defense lines of development" used in the UK: training, equipment, personnel, information, concepts and doctrine, organization, infrastructure, and logistics (TEPIDOIL).

It is common to consider the security architecture to be about "protection" in the sense that the architecture needs to protect various assets, whether these be operational assets, resource assets, information assets, and the like. Operational and resource mitigations can be defined that perform security processes that will satisfy security control tactics, techniques, and procedures, and that will comply with relevant security policies, directives and constraints.

This Guide is intended to be used in conjunction with the UAF Sample Problem that defines architecture views for a Search and Rescue Mission. UAF Specification documents (including the Sample Problem) can be downloaded from the OMG webpage: www.omg.org/spec/UAF/About-UAF/.

Step 7.0 – Define [Protection] Security Architectures – The main entry criterion for this Step is bringing forward the ***personnel architectures*** from Step 6 with associated human ***resource interfaces*** and their operational ***implementation*** context. The ***operational architecture*** is now examined for security ***operational mitigation*** inclusions using various ***operational agents*** from Step 3. Following this, the resource and personnel architectures are examined for security ***resource mitigation*** implementations using various ***resource performers*** from Steps 5 and 6.

These agents and performers along with their associated exchanges and functions will form alternatives that clarify the role of ***security controls*** to ***protect*** operational and resource assets by ***mitigating*** various ***risks*** that ***affect*** them.

- **Operational** – a set of **Operational Performers** intended to address against specific operational **Risks**
- **Resource Mitigation** – a set of Resource Performers intended to address against specific **Risk**
- **Asset** – an abstract element that indicates the types of elements that can be affected by **Risk**. Asset as applied to Security views is an abstract element that indicates the types of elements that can be considered as a subject for security analysis. (Note: types assets can include Operational Agents**, Operational Information,** Resource Performers, and **Resource Information**)
- **Security Control** – the management, operational, and technical control (i.e., safeguard or countermeasure) to **Protect** the confidentiality, integrity, and availability of the system and its information [NIST SP 800-53]
- **Risk** – a type that represents a situation involving exposure to danger of Affectable Elements (e.g., Assets, Processes, **Capabilities**, **Opportunities**, or **Enterprise Goals**) where the effects of such exposure can be characterized in terms of the likelihood of occurrence of a given threat and the potential adverse consequences of that threat's occurrence (Note: this is typically expressed as a statement of the impact of an event that **Affects** an asset representing a constraint in terms of adverse effects with an associated measure)
- **Affectable Element** – an abstract grouping of elements that can be affected by a **Risk** (Note: these include **Enterprise Goals**, Processes, Assets, **Opportunities**, and **Capabilities**)
- **Security Risk** – the level of impact on enterprise operations, assets, or individuals resulting from the operation of an information system given the potential impact of a threat and the likelihood of that threat occurring [NIST SP 800-65]
- **Mitigates** – a tuple relating a **Security Control** to a **Risk**. Mitigation is established to manage risk and could be represented as an overall strategy or through techniques (mitigation configurations) and procedures (**Security Processes**).

A ***security control*** is a prescribed safeguard or countermeasure for an information system to protect the confidentiality, integrity, and availability of the system and its information [NIST SP 800-53]. In various businesses, industries, and organizations, these systems and resource architectures are composed of a variety of resource artifacts. ***Security control*** features may apply broadly to systems in general and security controls may be intended or extended to cover broader areas of physical security and physical risks in general.

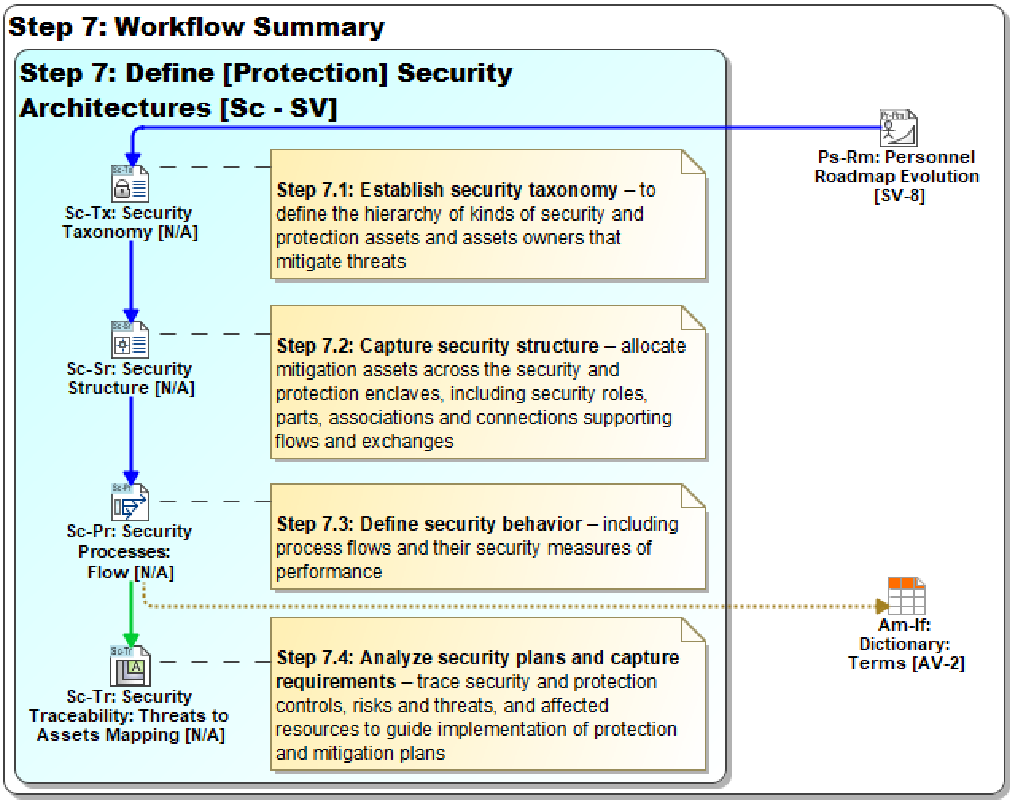

Step 7: Workflow Summary
Step 7: Define [Protection] Security Architectures [Sc - SV]
Sc-Tx: Security Taxonomy [N/A]
Step 7.1: Establish security taxonomy – to define the hierarchy of kinds of security and protection assets and assets owners that mitigate threats
Sc-Sr: Security Structure [N/A]
Step 7.2: Capture security structure – allocate mitigation assets across the security and protection enclaves, including security roles, parts, associations and connections supporting flows and exchanges
Sc-Pr: Security Processes: Flow [N/A]
Step 7.3: Define security behavior – including process flows and their security measures of performance
Sc-Tr: Security Traceability: Threats to Assets Mapping [N/A]
Step 7.4: Analyze security plans and capture requirements – trace security and protection controls, risks and threats, and affected resources to guide implementation of protection and mitigation plans
Ps-Rm: Personnel Roadmap Evolution [SV-8]
Am-If: Dictionary: Terms [AV-2]

Figure 4:47 - Workflow Summary for Step 7: Define [Protection] Security Architectures

When examining the security resource performer options, the use of security ***resources*** is considered in this Step 7 for specific trade-offs on how ***these*** resources will be utilized to address expected security risks.

The security architecture defines security process *behavior* elements and allocates these to security *structure* elements (e.g., operational or resource mitigations, to be used in roles, connectors, exchange items, exchanges, ports, and interfaces). The behavior elements define ***security processes*** that can exist independently of or jointly across ***operational activities*** that will in turn help achieve enterprise goals. The behavior elements also define ***security processes*** that can exist independently of or jointly across ***resource functions*** that will in turn implement operational activities.

Workflow Summary. A summary level view of the four steps involved in Step 7 workflow is shown above. The flowlines represent how one architecture view "influences" another architecture view. These lines do not represent a particular sequence of process activity execution or imply information exchanges. These should be thought of as influence diagrams rather than process diagrams. These architecture views (e.g., views, diagrams, tables) can be mapped to the architecture processes used in your organization and can be incorporated into an architecture modeling methodology.

The second-level steps in the Step 7 workflow are listed below along with the UAF views (and corresponding [DoDAF views] where applicable) associated with those steps. A complete list of the detailed steps at the third level with their corresponding views is provided at the end of this section.

199	**Step 7: Define [Protection] Security Architectures [Sc - SV]**	**Views**
200	**Step 7.1: Establish security taxonomy** – to define the hierarchy of kinds of security and protection assets and asset owners that mitigate threats	Sc-Tx: Security Taxonomy [N/A]
205	**Step 7.2: Capture security structure** – allocate mitigation assets across the security and protection enclaves, including security roles, parts, associations, and connections supporting flows and exchanges	Sc-Sr: Security Structure [N/A]
213	**Step 7.3: Define security behavior** – including process flows, and their security measures of performance	Sc-Pr: Security Processes: *Flow* [N/A]
216	**Step 7.4: Capture security deployment plans** – trace security and protection controls, risks, and threats, and affected resources to guide implementation of protection and mitigation plans	Sc-Tr: Security Traceability: *Threats to Assets Mapping* [N/A]

4.7.1 Security Taxonomy

Step 7.1 – Establish security taxonomy – A ***security taxonomy*** of security operational assets and assets roles are described or brought forward from Step 3.2 which could be susceptible to adverse events, along with ***risks*** that affect those ***operational performers***. ***Security controls***, which are types of ***requirements***, are then described which define to the intended result or outcome of ***mitigating*** the ***risks*** affecting the operational performers, and to ***protect*** the resources that implement those operational performers. New ***operational mitigations*** are described, along with new operational performers, which will satisfy the security controls.

This taxonomy includes understanding of ***generalizations***, particularly when a reference security architecture is setting contextual guidance, security asset categories are predefined, particular security strategies and plans are already known, or when ***enhanced*** or ***families*** of controls are deemed necessary. Step 7.1 and Step 3.2 may be conducted iteratively to adjust both operational performers and security taxonomies and to trade-off designs of the operational and security architectures.

- **Operational Mitigation** - a set of **Operational Performers** intended to address against specific operational **Risks** (Note: these performers are established to manage operational **Risks** which can represented as an overall strategy or through techniques (mitigation configurations) and procedures (**Security Processes**) They are a type of **Operational Architecture** which is used to mitigate a risk through satisfaction of a security control.))
- **Enhanced Security Control** – a statement of security capability to: (i) build in additional but related, functionality to a basic control; and/or (ii)increase the strength of a basic control
- **Security Control Family** – an element that organizes **Security Controls** into a family. Each **Security Control Family** contains **Security Controls** related to the general security topic of the family.

The total ***environment*** developed in Step 2.1 expected for deploying capabilities in terms of ***conditions***, ***locations***, ***environments***, including kinds of environments and ***kinds*** of locations, as well as ***geo-political*** factors is brought forward to describe the adversity or threat environment. This provides overall conditions for all security elements and becomes an overarching ***environmental*** context for ***risk*** measures.

The operational assets dealing with information are then categorized with associated security ***measurement*** to measure security impacts, along with features such as ***security availability***, ***security classification***, and ***security integrity***.

- **Security Measurement** – an abstract type grouping all types of security measurements (e.g., **Security Integrity**, **Security Availability**)
- **Security Availability** – details the potential impact on organizations or individuals if the information is not available to those who need to access it
- **Security Classification** – details a classification for the exchange
- **Security Integrity** – details the potential impact on organization or individuals due to modification or destruction of information, and includes ensuring information non-repudiation and authenticity

- **Security Classification Kind** - a type that defines acceptable values for the security category (SC) of an information system, where the acceptable values for potential impact are low, moderate, or high

Once resource architecture solutions are developed, a set of security resource ***assets*** and ***asset roles*** are now described or brought forward from Step 5.3, which are susceptible to adverse events that could affect the operational elements implemented by these resource elements. The ***security controls***, previously identified, are identified that can ***protect*** those resources. Next, ***resource mitigations*** or ***security enclaves*** are described, along with new ***resource performers***, which will satisfy the ***security controls***, and which ***implement*** the ***operational performers*** doing the same. Step 7.1 and Step 5.3 may be conducted iteratively to adjust both resource performers and security taxonomies and to trade-off designs of the operational and security architectures.

- **Asset** – an abstract element that indicates the types of elements that can be affected by **Risk**. Asset as applied to Security views is an abstract element that indicates the types of elements that can be considered as a subject for security analysis.
- **Asset Role** – an abstract element that indicates the types of elements that can be affected by **Risk** in the particular context. **Asset Role** as applied to Security views, is an abstract element that indicates the type of elements that can be considered as a subject for security analysis in the particular context.
- **Resource Mitigation** – a set of Resource Performers intended to address against specific **Risk** (Note: these performers are established to manage resource **Risks** which can be represented as an overall strategy or through techniques (mitigation configurations) and procedures (**Security Processes**). They are a type of **Resource Architecture** which is a resource or structured resources that are used to mitigate a risk through satisfaction of a security control.)
- **Security Enclave** – a collection of information systems connected by one or more internal networks under the control of a single authority and security policy. The systems may be structured by physical proximity or by function, independent of location. (Note: a **Security Enclave** is a type of **Resource Mitigation**).

An organizational resource, such as an ***organization*** or ***post***, ***owns a risk***, and in the resource architecture's design implementation must take responsibility for implementing the solutions provided by the resource mitigations for the risk.

- **Owns Risk** – a tuple relating a **Risk** to an organizational resource that is responsible for executing the risk mitigation

Operational and resource elements affected by risks or protected by or providing security controls that are subject to the **security constraints** and their **rules**.

- **Security Constraint** – a type of rule that captures a formal statement to define access control policy language

A review is conducted to ensure all security elements have been covered or addressed within the ***security taxonomy*** table.

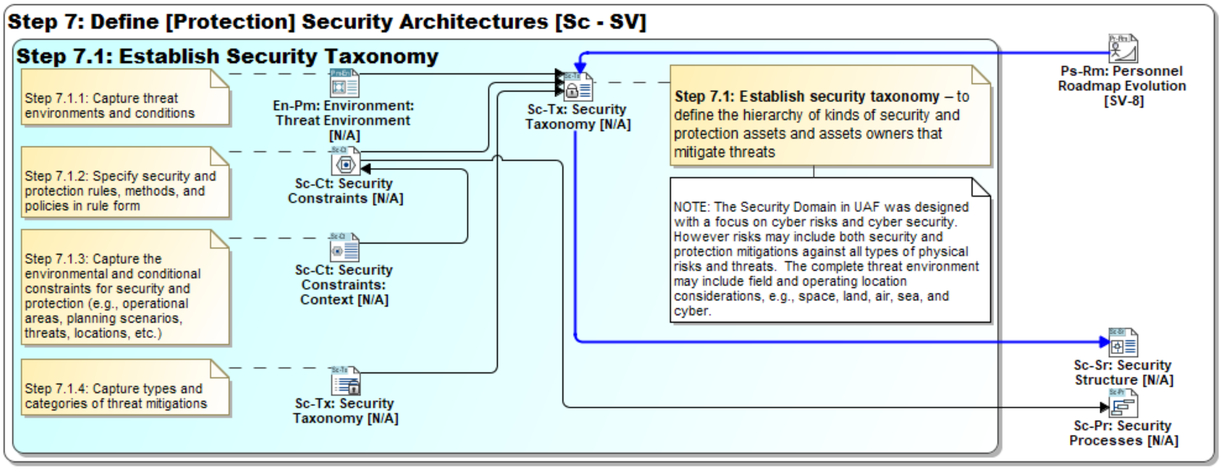

Step 7: Define [Protection] Security Architectures [Sc - SV]
Step 7.1: Establish Security Taxonomy
Step 7.1.1: Capture threat environments and conditions
Step 7.1.2: Specify security and protection rules, methods, and policies in rule form
Step 7.1.3: Capture the environmental and conditional constraints for security and protection (e.g., operational areas, planning scenarios, threats, locations, etc.)
Step 7.1.4: Capture types and categories of threat mitigations
En-Pm: Environment: Threat Environment [N/A]
Sc-Ct: Security Constraints [N/A]
Sc-Ct: Security Constraints: Context [N/A]
Sc-Tx: Security Taxonomy [N/A]
Sc-Tx: Security Taxonomy [N/A]
Step 7.1: Establish security taxonomy – to define the hierarchy of kinds of security and protection assets and assets owners that mitigate threats
NOTE: The Security Domain in UAF was designed with a focus on cyber risks and cyber security. However risks may include both security and protection mitigations against all types of physical risks and threats. The complete threat environment may include field and operating location considerations, e.g., space, land, air, sea, and cyber.
Ps-Rm: Personnel Roadmap Evolution [SV-8]
Sc-Sr: Security Structure [N/A]
Sc-Pr: Security Processes [N/A]

Figure 4:48 - Step 7.1: Establish Security Taxonomy

4.7.2 Security Structure

Step 7.2 – Capture Security Structure – Internal structure of operational assets within the taxonomy are developed with internal features and characteristics, including ***operational roles*** they have, in accordance with Step 3.2. Similarly, internal structure of resource assets within the taxonomy are also developed with internal features and characteristics, including the ***resource roles*** they have, in accordance with Step 5.3.

As in Step 3.2 and 5.3, the operational and resource structure includes establishing ***role kinds, exchange items, connectors***, and ***messages***. Once again performers may perform multiple and different roles, and one role may be performed by multiple and different performers. Examination of this is necessary to scope and perform assessment of the trade space and an analysis of alternatives, including differences between composition and aggregation in performer structure. Additionally, an operations ***conceptual information model*** and a resources ***logical information model*** are created to define the ***operational information*** elements and ***resource information*** elements which are exchanged.

The focus of ***security structure*** is to capture the allocation of assets (operational and resource, information, and data) across the security enclaves, show applicable security controls necessary to protect organizations, systems and information during processing, while in storage, and during transmission. This view also captures asset aggregation and allocates the usage of the aggregated information to a location using the security property.

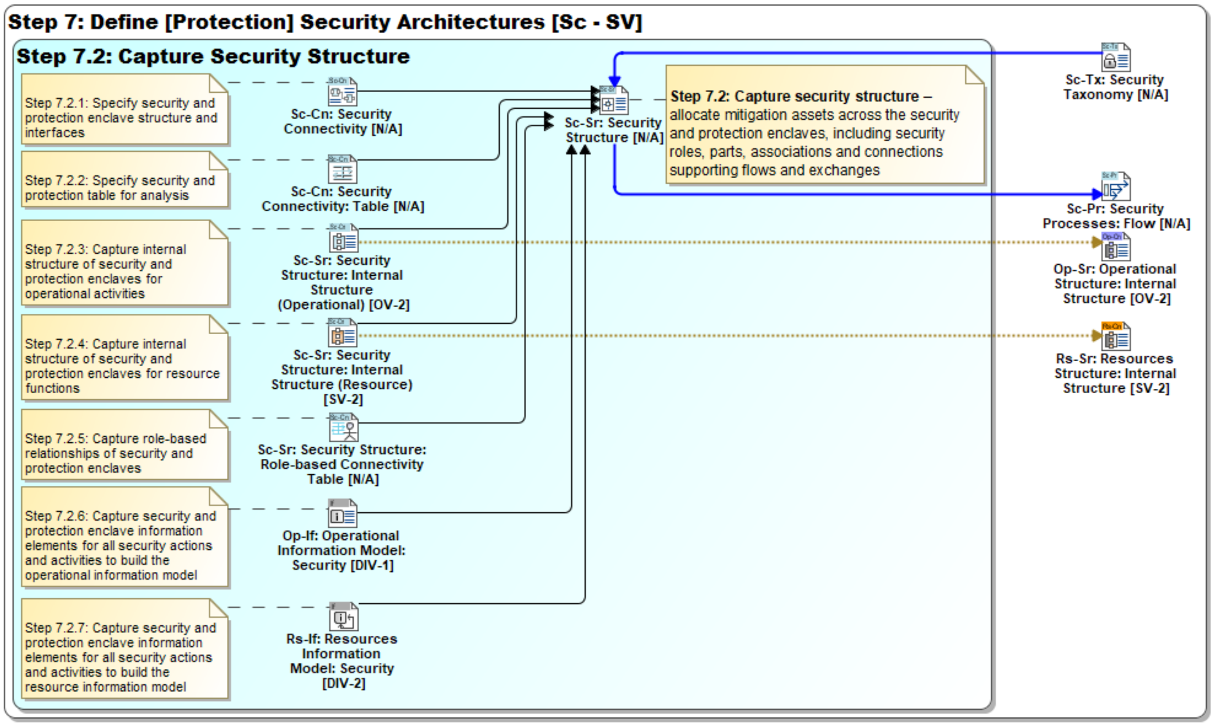

Step 7: Define [Protection] Security Architectures [Sc - SV]
Step 7.2: Capture Security Structure
Step 7.2.1: Specify security and protection enclave structure and interfaces
Sc-Cn: Security Connectivity [N/A]
Step 7.2.2: Specify security and protection table for analysis
Sc-Cn: Security Connectivity: Table [N/A]
Step 7.2.3: Capture internal structure of security and protection enclaves for operational activities
Sc-Sr: Security Structure: Internal Structure (Operational) [OV-2]
Step 7.2.4: Capture internal structure of security and protection enclaves for resource functions
Sc-Sr: Security Structure: Internal Structure (Resource) [SV-2]
Step 7.2.5: Capture role-based relationships of security and protection enclaves
Sc-Sr: Security Structure: Role-based Connectivity Table [N/A]
Step 7.2.6: Capture security and protection enclave information elements for all security actions and activities to build the operational information model
Op-If: Operational Information Model: Security [DIV-1]
Step 7.2.7: Capture security and protection enclave information elements for all security actions and activities to build the resource information model
Rs-If: Resources Information Model: Security [DIV-2]
Sc-Sr: Security Structure [N/A]
Step 7.2: Capture security structure – allocate mitigation assets across the security and protection enclaves, including security roles, parts, associations and connections supporting flows and exchanges
Sc-Tx: Security Taxonomy [N/A]
Sc-Pr: Security Processes: Flow [N/A]
Op-Sr: Operational Structure: Internal Structure [OV-2]
Rs-Sr: Resources Structure: Internal Structure [SV-2]

Figure 4:49 - Step 7.2: Capture Security Structure

4.7.3 Security Behavior

Step 7.3 – Define Security Behavior – Process flow diagrams are constructed for all ***security processes*** including ***security process actions*** classified by ***security processes*** and process control-flow mechanisms such as decision nodes and forks. When ***operational*** or ***resource mitigations*** are already known or defined in Step 7.1, they are assigned to swim-lanes and the actions they are capable to perform. Otherwise, ***security process actions*** are grouped into swim-lanes to create ***operational*** or ***resource mitigations*** which must be structured in Step 7.1 – 7.2.

The process flows are refined, when needed, with ***operational*** or ***resource state descriptions*** for operational or resource performers and sequenced timelines of ***operational*** or ***resource messages***, in accordance with Step 3.2, 5.4 and 6.3 development of operational, resource, and personnel process flows[8]. ***Operational*** or ***resource connections*** are added to the process flow to realize the ***interaction messages***. State descriptions and sequenced timelines may now be further described with these item exchanges.

Security processes and ***security process actions*** are shown inter-woven with ***operational activities*** and ***operational activities actions***, and resource ***functions*** and ***function actions***. Performers in both the Operational and Resource Viewpoints will be capable to perform both their normal Viewpoint actions, as well as security process actions.

Performance ***measures*** are defined for the performers and their ***security process actions***, or ***operational activities*** or ***functions***. Performance ***measures*** may include parametric diagrams when needed. Measures of security elements may demonstrate satisfaction of capability MOE's when operational activities are mapped to security types of

[8] NOTE: the security process flows can either be created as stand-alone flow diagrams or integrated with operational or resource process flow diagrams.

capabilities. This may be done either directly or indirectly through examination or correlation of security measures with the operational MOPs or their associated resource TPMs.

- **Measurement** – a property of an element representing something in the physical world, expressed in amounts of a unit of measure

A review is conducted to ensure all relevant ***operational mitigations*** and their performers have been ***implemented*** by ***resource mitigations***, and their performers. Additionally, ***security process*** structural relationships are examined through intervening ***security process actions*** to check for redesign or simplification, for adjustments, as well as possible duplication in implementation coverage of operational activities. ***Security process actions*** may be viewed either in complex meta-chain maps or in simple diagrams to ensure a complete library of ***security processes*** is established and understood. This library may be arranged by security enclave or process groupings, resources capable to perform them, or some other useful organization.

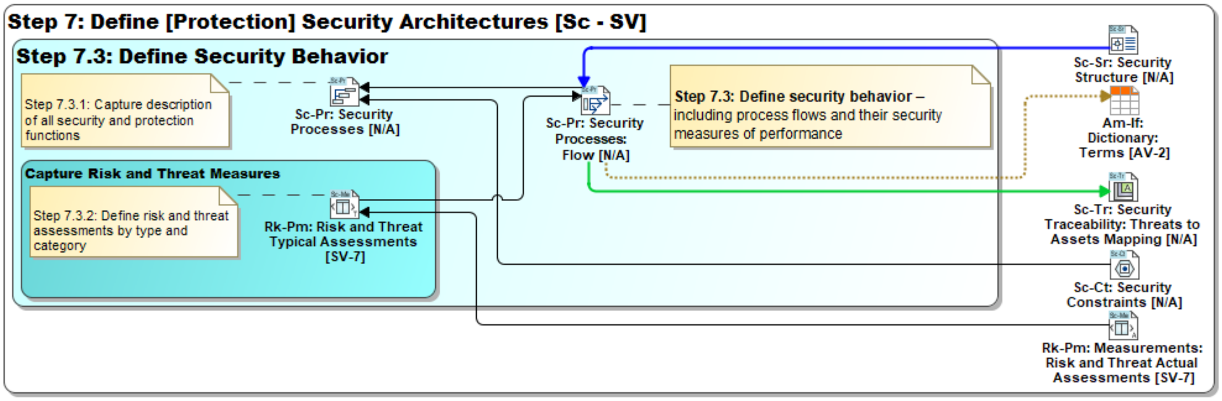

Step 7: Define [Protection] Security Architectures [Sc - SV]
Step 7.3: Define Security Behavior
Step 7.3.1: Capture description of all security and protection functions
Sc-Pr: Security Processes [N/A]
Sc-Pr: Security Processes: Flow [N/A]
Step 7.3: Define security behavior – including process flows and their security measures of performance
Sc-Sr: Security Structure [N/A]
Am-If: Dictionary: Terms [AV-2]
Capture Risk and Threat Measures
Step 7.3.2: Define risk and threat assessments by type and category
Rk-Pm: Risk and Threat Typical Assessments [SV-7]
Sc-Tr: Security Traceability: Threats to Assets Mapping [N/A]
Sc-Ct: Security Constraints [N/A]
Rk-Pm: Measurements: Risk and Threat Actual Assessments [SV-7]

Figure 4:50 - Step 7.3: Define Security Behavior

4.7.4 Security Deployment Plans

Step 7.4 – Analyze Security Plans and Capture Requirements – Once a security architecture has been captured and defined, organization of security plans can begin. Security traces of affected and protected resources are analyzed to review impacts on resource designs, or to examine variations and specializations in security controls. Impact analysis is conducted on resource and resource role implementations. Security alternatives may be examined, and risk, adversity or threat evaluations may be re-examined, as well as expanded to other kinds and types of risks, such as physical risks, threat risks, project risks, and safety risks. Typically, feasibility, cost and other factors may be tied to the security elements to help drive other kinds of analysis.

An entire security architecture, with all its elements (agents, behaviors, etc.) may be set as one option in an analysis of alternatives between others. Each alternative to be examined can be represented by a separate security architecture. Typical performance measures from Step 7.3 may be used to generate ***actual measurements*** to either capture analysis of these alternatives or records of analysis for performers over time. Each actual measurement table is assigned start and end dates of their actual, estimated, or required points of existence, and are used to track validation, satisfaction, and realization of performance changes over time.

- **Actual Measurement** – an actual value that is applied to a **Measurement** (Note: a measurement may be one of three kinds: actual, required, or estimate, and may have an associated start and/or end date)
- **Required** – an enumerated type of an **Actual Measurement Kind** which is based on a required value
- **Actual Measurement Kind** – an enumerated type of an actual measurement kind which is based on a required value (Note: a measurement may be one of three kinds: **Actual**, **Required**, or **Estimate**, and may have an associated start and/or end date)

If the security architecture is used as a reference architecture to evaluate bids for contracts from security resource mitigation developers, ***actual measurements*** may form a basis of evaluation for bids received on parts or for all of the security resource mitigation architecture.

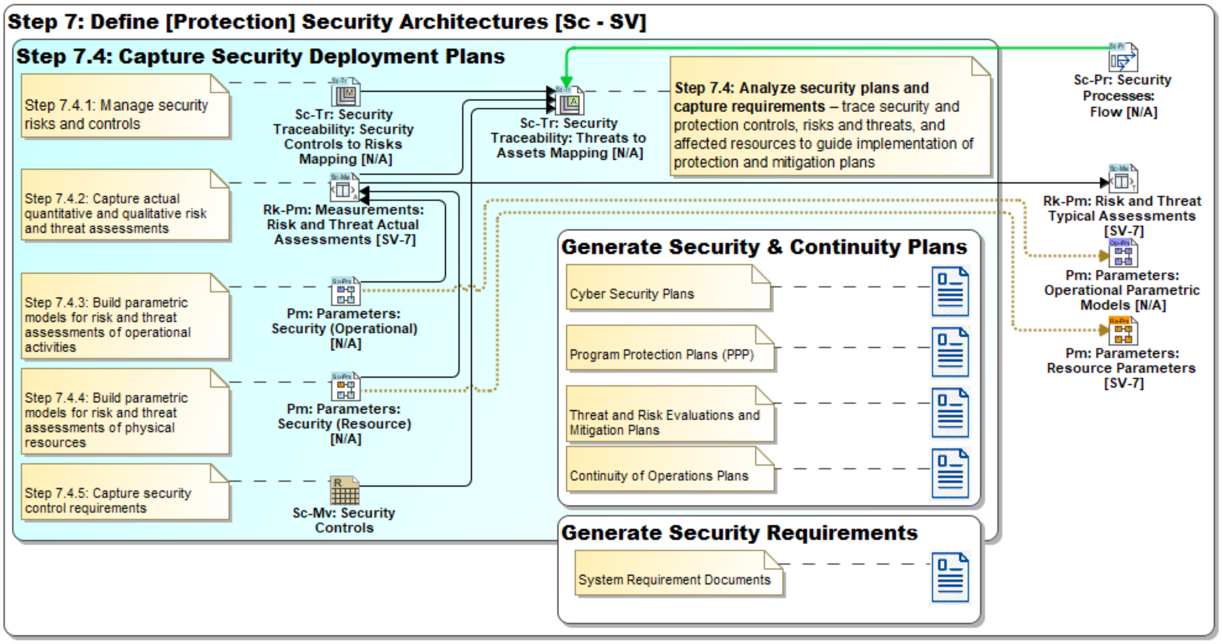

Step 7: Define [Protection] Security Architectures [Sc - SV]
Step 7.4: Capture Security Deployment Plans
Step 7.4.1: Manage security risks and controls
Sc-Tr: Security Traceability: Security Controls to Risks Mapping [N/A]
Sc-Tr: Security Traceability: Threats to Assets Mapping [N/A]
Step 7.4: Analyze security plans and capture requirements – trace security and protection controls, risks and threats, and affected resources to guide implementation of protection and mitigation plans
Sc-Pr: Security Processes: Flow [N/A]
Step 7.4.2: Capture actual quantitative and qualitative risk and threat assessments
Rk-Pm: Measurements: Risk and Threat Actual Assessments [SV-7]
Rk-Pm: Risk and Threat Typical Assessments [SV-7]
Step 7.4.3: Build parametric models for risk and threat assessments of operational activities
Pm: Parameters: Security (Operational) [N/A]
Pm: Parameters: Operational Parametric Models [N/A]
Step 7.4.4: Build parametric models for risk and threat assessments of physical resources
Pm: Parameters: Security (Resource) [N/A]
Pm: Parameters: Resource Parameters [SV-7]
Step 7.4.5: Capture security control requirements
Sc-Mv: Security Controls
Generate Security & Continuity Plans
Cyber Security Plans
Program Protection Plans (PPP)
Threat and Risk Evaluations and Mitigation Plans
Continuity of Operations Plans
Generate Security Requirements
System Requirement Documents

Figure 4:51 - Step 7.4: Capture Security Deployment Plans

When needed, security plans, such as cyber security plans, physical security plans, threat response plans, can be developed based on the models and views generated during this Step 6. Ideally these documents are automatically generated from the model itself using reporting scripts and model queries. Additionally, the architectural description itself may be published or shared with security planners or bidders to communicate implementation plans, requests for information, and requests for proposals.

Additionally, a set of security ***requirements*** can be generated based on the elements within a Security Architecture, which are intended to be service, resource and personnel implementation agnostic. Requirements tracing from the Security Architecture can be satisfied by service, resource, personnel, and security solutions.

Requirements are developed and related to architectural elements relevant to the generation of the requirement. These are related using ***trace, verify, satisfy***, or ***refine*** relationships for linking to relevant elements in the architecture.

- **Requirement** – a statement that identifies a system, product or process characteristic or constraint, which is unambiguous, clear, unique, consistent, stand-alone (not grouped), and verifiable, and is deemed necessary for stakeholder acceptability (INCOSE 2010)
- **Refine** – a relationship from an architectural element which refines a text-based requirement

4.7.5 Architecture View Summary for Step 7

The view specifications in UAF for this viewpoint are outlined here:

	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Security Sc	Security Controls Sc-Mv	Security Taxonomy Sc-Tx	Security Structure Sc-Sr	Security Connectivity Sc-Cn	Security Processes Sc-Pr	-	-	Resources Information Model (Rs-If)	**Envmt, Msmts, Risks (En-Pm, Me-Pm, Rk-Pm)**	Security Constraints Sc-Ct	-	Security Traceability Sc-Tr

A summary of the lower-level steps and UAF views is shown below. The architecture views produced by each step are identified by the UAF grid designator Aa-Bb where Aa represents the Viewpoint row in the grid and Bb represents the Aspect column in the grid. DoDAF view designations are also shown in [brackets], when applicable. The name of the architecture view matches the name of the view specification in the UAFML. In some cases, a subtitle is provided to identify what kind of information is provided in an instance of that UAF view. There will be some architecture views that are not part of the UAF specification since you will sometimes need fit-for-purpose (i.e., custom) views that capture the necessary architectural information.

199	**Step 7: Define [Protection] Security Architectures [Sc - SV]**	**Views**
200	**Step 7.1: Establish security taxonomy** – to define the hierarchy of kinds of security and protection assets and assets owners that mitigate threats	Sc-Tx: Security Taxonomy [N/A]
201	Step 7.1.1: Capture threat environments and conditions	En-Pm: Environment: *Threat Environment* [N/A]
202	Step 7.1.2: Specify security and protection rules, methods, and policies in rule form	Sc-Ct: Security Constraints [N/A]
203	Step 7.1.3: Capture the environmental and conditional constraints for security and protection (e.g., operational areas, planning scenarios, threats, locations, etc.)	Sc-Ct: Security Constraints: *Context* [N/A]
204	Step 7.1.4: Capture types and categories of threat mitigations	Sc-Tx: Security Taxonomy [N/A]
205	**Step 7.2: Capture security structure** – allocate mitigation assets across the security and protection enclaves, including security roles, parts, associations and connections supporting flows and exchanges	Sc-Sr: Security Structure [N/A]
206	Step 7.2.1: Specify security and protection enclave structure and interfaces	Sc-Cn: Security Connectivity [N/A]
207	Step 7.2.2: Specify security and protection table for analysis	Sc-Cn: Security Connectivity: *Table* [N/A]
208	Step 7.2.3: Capture internal structure of security and protection enclaves for operational activities	Sc-Sr: Security Structure: *Internal Connectivity (Operational)* [OV-2]
209	Step 7.2.4: Capture internal structure of security and protection enclaves for resource functions	Sc-Sr: Security Structure: *Internal Connectivity (Resource)* [SV-2]
210	Step 7.2.5: Capture role-based relationships of security and protection enclaves	Sc-Sr: Security Structure: *Role-based Connectivity Table* [N/A]
211	Step 7.2.6: Capture security and protection enclave information elements for all security actions and activities to build the operational information model	Op-If: Operational Information Model: *Security* [DIV-1]
212	Step 7.2.7: Capture security and protection enclave information elements for all security actions and activities to build the resource information model	Rs-If: Resources Information Model: *Security* [DIV-2]

213	**Step 7.3: Define security behavior** – including process flows, and their security measures of performance	Sc-Pr: Security Processes: *Flow* [N/A]
214	Step 7.3.1: Capture description of all security and protection functions	Sc-Pr: Security Processes [N/A]
215	Step 7.3.2: Define risk and threat assessments by type and category	Rk-Pm: Risks: *Risk and Threat Typical Assessments* [SV-7]
216	**Step 7.4: Analyze security plans and capture requirements** – trace security and protection controls, risks and threats, and affected resources to guide implementation of protection and mitigation plans	Sc-Tr: Security Traceability: *Threats to Assets Mapping* [N/A]
217	Step 7.4.1: Manage security risks and controls	Sc-Tr: Security Traceability: *Security Controls to Risks Mapping* [N/A]
218	Step 7.4.2: Capture actual quantitative and qualitative risk and threat assessments	Rk-Pm: Risk: *Risk and Threat Actual Assessments* [SV-7]
219	Step 7.4.3: Build parametric models for risk and threat assessments of operational activities	Pm: Parameters*: Security (Operational)* [N/A]
220	Step 7.4.4: Build parametric models for risk and threat assessments of physical resources	Pm: Parameters: *Security (Resource)* [N/A]
221	Step 7.4.5: Capture security control requirements	Sc-Mv: Security Controls [N/A]

STEP 8 – Projects Portfolio Management

4.8 Step 8 – Projects Portfolio Management

Purpose. The purpose of this step is to describe projects and project milestones, how those projects deliver resources that lead to capabilities, the organizations contributing to the projects and dependencies between projects. Key stakeholders for this step are Project Managers, Project Portfolio Managers and Enterprise Architects. Their concerns are mainly about what are the projects in the project portfolio, what are their project milestones and how are these associated with resources that make up the capability configuration in the capability roadmaps.

Conceptual Schema. The key concepts used in the Projects Viewpoint that can be used as model elements in the architecture views and the relationships between these concepts are illustrated in the conceptual schema shown in Figure 4:52. These ***key concepts*** are highlighted in italics within the Narrative and some of the less obvious concepts are listed with the associated UAF meaning of that concept. Detailed definition of the entities and relationships shown in the conceptual schema are provided in the UAFML specification document.

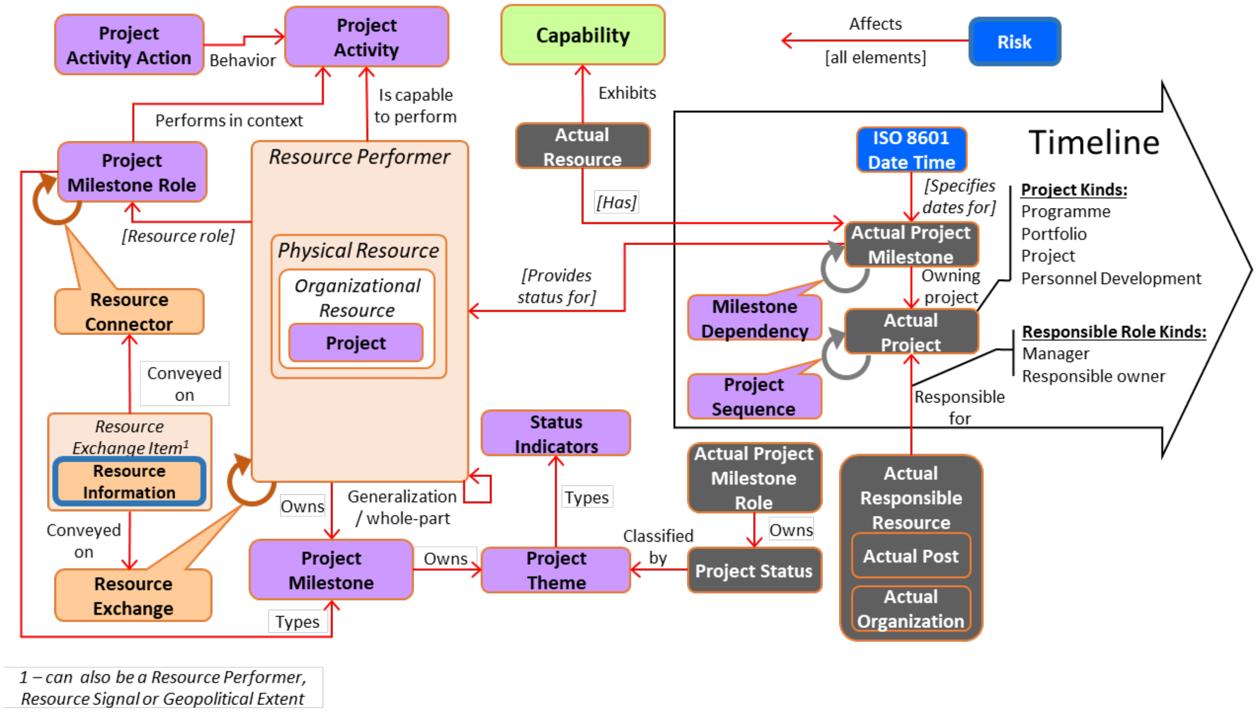


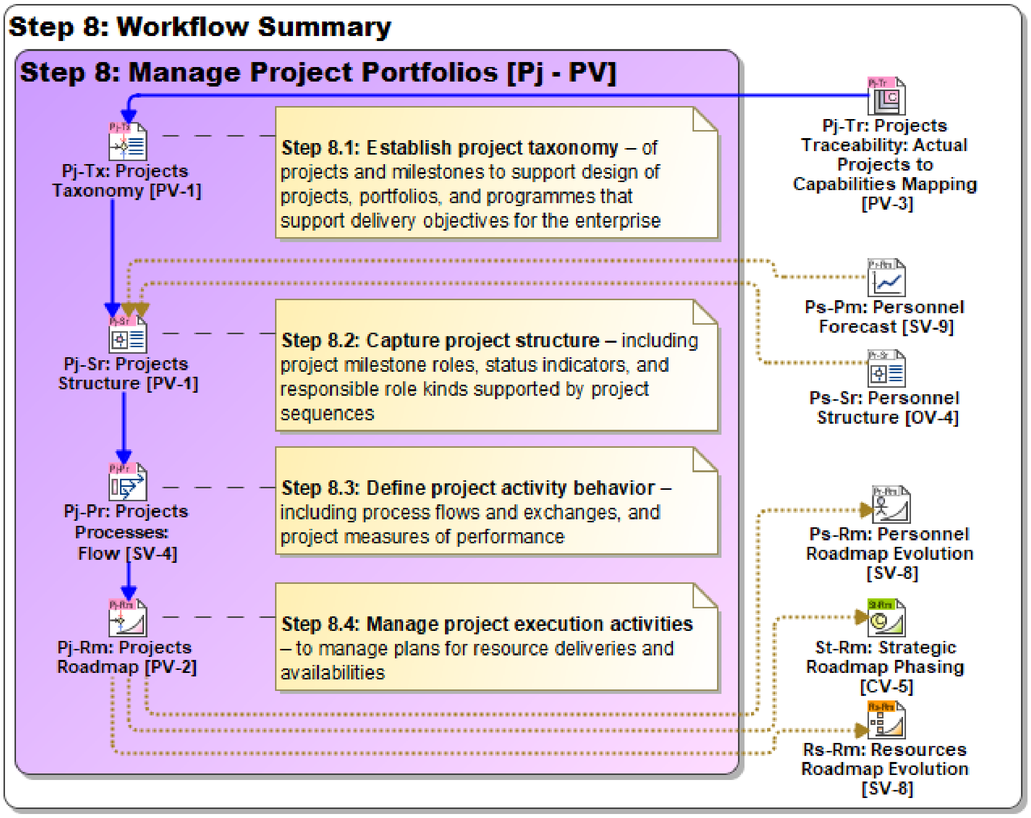
Figure 4:52 - Conceptual Schema for Projects Portfolio Management

The project portfolio concepts are shown in pink and the affected architecture concepts are shown on the right in legend-based colors. ***Projects*** will conduct ***project activities*** that "support" capabilities and "realize" ***resources***, using ***project milestones*** that provide ***project themes*** with ***status indicators***. ***Managers*** and ***responsible owners*** are responsible for overall ***project***, ***portfolio***, or ***programme*** achievements and progress.

Projects will have ***resource roles, project sequences, resource exchange items, resource exchanges,*** and resource performers that are associated with the ***functions*** implementing the projects in the project portfolio.

This Guide is intended to be used in conjunction with the UAF Sample Problem that defines architecture views for a Search and Rescue Mission. UAF Specification documents (including the Sample Problem) can be downloaded from the OMG webpage: www.omg.org/spec/UAF/About-UAF/.

Step 8.0 – Manage Project Portfolios – The main entry criterion for this Step is bringing forward the planned ***capabilities*** from Step 2 with associated mappings to existing or planned projects. The projects portfolio is developed in parallel with Steps 3 through 7. As ***forecasts***, ***roadmaps***, and ***actual enterprise phases*** are developed, their various use of start and end ***dates***, ***timelines***, and ***actual project milestones*** will affect or be driven by project planning.



Step 8: Workflow Summary
Step 8: Manage Project Portfolios [Pj - PV]
Pj-Tx: Projects Taxonomy [PV-1]
Step 8.1: Establish project taxonomy – of projects and milestones to support design of projects, portfolios, and programmes that support delivery objectives for the enterprise
Pj-Sr: Projects Structure [PV-1]
Step 8.2: Capture project structure – including project milestone roles, status indicators, and responsible role kinds supported by project sequences
Pj-Pr: Projects Processes: Flow [SV-4]
Step 8.3: Define project activity behavior – including process flows and exchanges, and project measures of performance
Pj-Rm: Projects Roadmap [PV-2]
Step 8.4: Manage project execution activities – to manage plans for resource deliveries and availabilities
Pj-Tr: Projects Traceability: Actual Projects to Capabilities Mapping [PV-3]
Ps-Pm: Personnel Forecast [SV-9]
Ps-Sr: Personnel Structure [OV-4]
Ps-Rm: Personnel Roadmap Evolution [SV-8]
St-Rm: Strategic Roadmap Phasing [CV-5]
Rs-Rm: Resources Roadmap Evolution [SV-8]

Figure 4:53 - Workflow Summary for Step 8: Management Project Portfolios

Projects are organized, arranged and grouped into ***portfolios*** and ***programmes*** that will obtain the resources necessary to implement operational activities and provide capabilities. These ***projects*** along with their associated exchanges and functions will form alternatives that are compared in trade-offs to support procurement and acquisition decisions.

- **Project** – a type that describes types of time-limited endeavors that are required to meet one or more **Capability** needs (Note: these typically represent a planned endeavor executed by an **Actual Organization** responsible for actions and progress in accordance with **Actual Project Milestones**)
- **ISO 8601 Date Time** – a date and time specified in the ISO8601 date-time format including time zone designator (TZD): YYYY-MM-DDThh:mm:ssTZD (Note: it is useful to display elements with date times in a timeline format, as well as organize date times by **Projects**)
- **Actual Project Milestone** – an event with a start date in an **Actual Project** from which progress is measured

A project model defines project *behavior* elements (e.g., processes) and allocates these to project *structure* elements to be used in roles, exchange items, and exchanges. The behavior elements define ***project activities*** that will in turn help achieve enterprise goals.

Workflow Summary. A summary level view of the six steps involved in Step 8 workflow is shown below. The flowlines represent how one architecture view “influences” another architecture view. These lines do not represent a particular sequence of process activity execution or imply information exchanges. These should be thought of as influence diagrams rather than process diagrams. These architecture views (e.g., views, diagrams, tables) can be mapped to the architecture processes used in your organization and can be incorporated into an architecture modeling methodology.

The second-level steps in the Step 8 workflow are listed below along with the UAF views (and corresponding [DoDAF views] where applicable) associated with those steps. A complete list of the detailed steps at the third level with their corresponding views is provided at the end of this section.

222	**Step 8: Manage Project Portfolios [Pj - PV]**	**Views**
223	**Step 8.1: Establish project taxonomy** – of projects and milestones to support design of projects, portfolios, and programmes that support delivery objectives for the enterprise	Pj-Tx: Projects Taxonomy [PV-1]
226	**Step 8.2: Capture project structure** – including project milestone roles, status indicators, and responsible role kinds supported by project sequences	Pj-Sr: Projects Structure [PV-1]
229	**Step 8.3: Define project activity behavior** – including process flows and exchanges, and project measures of performance	Pj-Pr: Projects Processes: *Flow* [N/A]
233	**Step 8.4: Manage project execution activities** – to manage plans for resource deliveries and availabilities	Pj-Rm: Projects Roadmap [PV-2]

4.8.1 Project Taxonomy

Step 8.1 – Establish project taxonomy – A set of ***projects*** are described, including any that have been preliminarily identified from Step 2.2, where capabilities are needed or anticipated that will require the use of an existing ***actual project*** or design of a new project. These ***projects*** are accounted for in a ***project taxonomy*** and may include ***portfolios***, ***programmes***, and ***personnel development*** as well as ***project milestones***. This view captures understanding of ***generalizations***, particularly when a reference project model is setting contextual guidance, project and project milestone categories are known, or particular project plans are already known.

- **Actual Project** – a time-limited endeavor to provide a specific set of **Actual Resources** that meet specific **Capability** needs (Note: this is an instance of a **Project** in the real world)
- **Project Milestone** – a type of event in a **Project** by which progress is measured (Note: e.g., design reviews, readiness events, deployment and launch events, retirements, disposals)
- **Project Kind** – a possible enumeration kind of an **Actual Project** to include a **Project**, **Programme**, or **Portfolio**

Actual Projects, along with their sub-type ***projects***, ***programmes***, and ***portfolios*** are elements intended to describe composite structures of a complex nature that are organized for different purposes in meeting enterprise strategic objectives. Project designs and types drive enterprise organizational structures for managers to change or transform business practices.

- **Project** – an enumerated option of **Project Kind** of an **Actual Project** (Note: this typically describes types of time-limited endeavors that are required to meet one or more **Capability** needs and may represent a planned endeavor executed by an **Actual Organization** responsible for actions and progress in accordance with **Actual Project Milestones**)
- **Programme** – an enumerated option of **Project Kind** of an **Actual Project** (Note: this typically describes an undertaking that is a temporary, flexible organization created to co-ordinate, direct and oversee the implementation of a set of related projects and tasks in order to deliver outcomes and benefits related to the organization’s strategic objectives. A programme is likely to have a lifespan of several years or decades. During a programme lifecycle, projects are initiated, executed, and closed. Programmes provide an umbrella under which these projects can be coordinated. The programme integrates the projects so that it can deliver an outcome greater than the sum of its parts.)

- **Portfolio** – an enumerated option of **Project Kind** of an **Actual Project** (Note: this typically describes an undertaking comprised of the projects and programmes that are the totality of an organization's investment (or segment thereof) in the changes required to achieve its strategic objectives)

A review is done to ensure all resources have been covered or addressed within the projects taxonomy table.

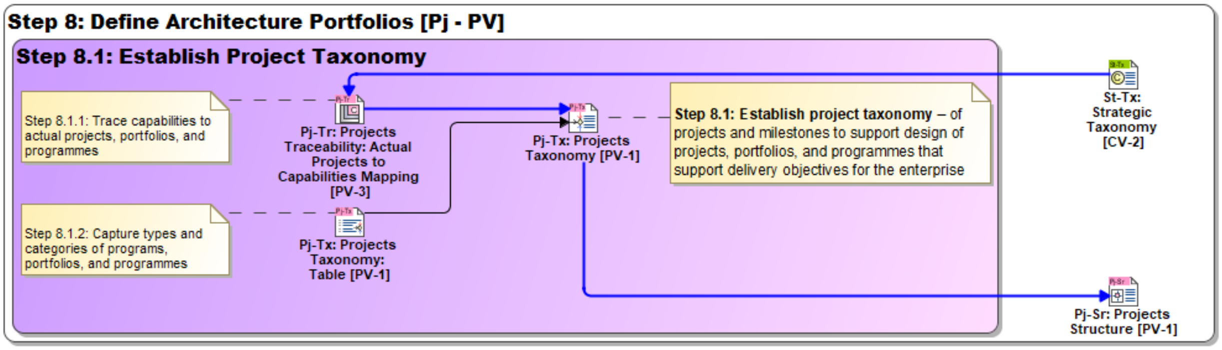

Step 8: Define Architecture Portfolios [Pj - PV]
Step 8.1: Establish Project Taxonomy
Step 8.1.1: Trace capabilities to actual projects, portfolios, and programmes
Step 8.1.2: Capture types and categories of programs, portfolios, and programmes
Pj-Tr: Projects Traceability: Actual Projects to Capabilities Mapping [PV-3]
Pj-Tx: Projects Taxonomy: Table [PV-1]
Pj-Tx: Projects Taxonomy [PV-1]
Step 8.1: Establish project taxonomy – of projects and milestones to support design of projects, portfolios, and programmes that support delivery objectives for the enterprise
St-Tx: Strategic Taxonomy [CV-2]
Pj-Sr: Projects Structure [PV-1]

Figure 4:54 - Step 8.1: Establish Project Taxonomy

4.8.2 Project Structure

Step 8.2 – Capture Project Structure – When capturing project structure, it is sometimes necessary to understand existing enterprise organizations that are executing existing projects and types of projects. A ***project traceability*** is brought in or created which maps ***actual organizational resources*** to ***actual projects***. This serves as a basis from which new project structure can be integrated or designed to transform actual project structures. New project structures are sometimes created based on evolution of project processes or frameworks, especially if the enterprise or program requires a dynamic and agile approach to change.

Internal structure of projects within the taxonomy are developed with internal features and characteristics, including the ***resource roles*** they have as part of a greater kind of project. One project kind may perform different resource roles and one resource role may be performed by different project kinds. Examination of this is necessary to scope and perform assessment of the trade space and an analysis of alternatives, including differences between composition and aggregation in projects, personnel development, programmes, and portfolio structure. ***Project milestones*** should be developed and described which characterize the different kinds suitable and necessary for an enterprise organization, such as milestones for design reviews, readiness events, deployment and launch events, retirements, disposals, etc.

- **Project Resource Role** – the role played by a **Project Milestone** in the context of a **Project**

Project Milestone structure is organized by ***Project Themes*** which give ***Status Indicators***.

- **Project Theme** – a property of a **Project Milestone** that captures an aspect by which the progress of **Actual Projects** may be measured (Note: e.g., doctrine, organization, training, materiel, logistics, personnel, facilities)
- **Status Indicators** – an enumerated type that specifies a status for a **Project Theme** (Note: which indicates selectable progress options)

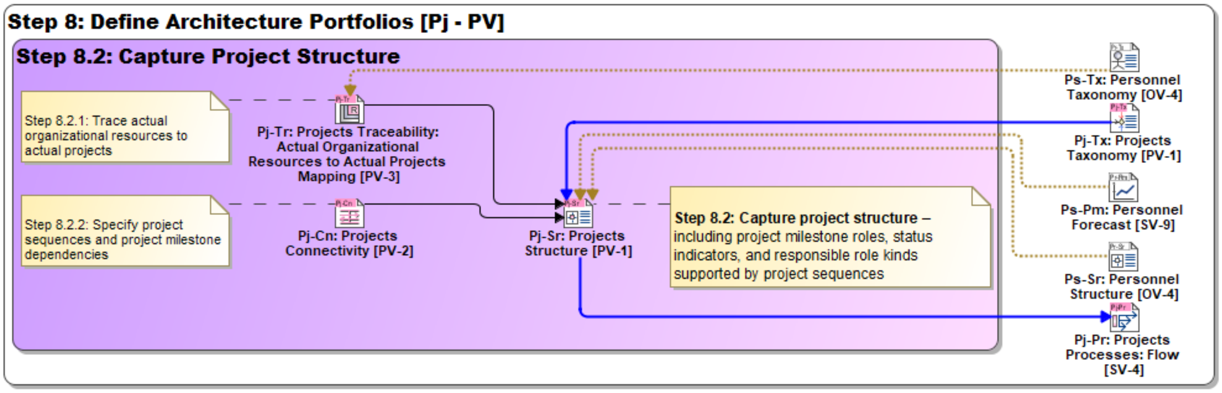
Actual Organizations or ***Actual Posts*** are designated which act as the ***Managers*** and ***Responsible Owners*** who are ***responsible for*** the various kinds of projects or specific project milestones.

- **Responsible Role Kind** – an enumerated type that specifies a Manager or Responsible Owner designation for an Actual Organization or Actual Post
- **Responsible For** – a tuple between an **Actual Responsible Resource** and an **Actual Responsibility** or **Actual Project**. It defines the duties that the **Actual Responsible Resource** is **Responsible For**. (Note: in

this case a relationship between an **Actual Organization** or **Actual Post** and the **Actual Project** or **Actual Project Milestone** that they are responsible for)

Now that projects, portfolios, project milestones, etc. are structured by their kinds and themes, with defined status indicators and responsible organizations, the projects and their milestones must be understood on the basis of timelines. ***Milestone dependencies*** are created for milestones of certain kinds and themes are dependent upon other milestones to precede them. Similarly, ***project sequences*** are created to show when certain ***projects*** must finish before others can begin to ensure that critical paths are understood.

- **Milestone Dependency** – a tuple between two **Actual Project Milestones** that denotes one **Actual Project Milestone** follows from another
- **Project Sequence** – a tuple between two **Actual Projects** that denotes one **Actual Project** cannot start before the previous **Actual Project** is finished


Step 8: Define Architecture Portfolios [Pj - PV]
Step 8.2: Capture Project Structure
Pj-Tr: Projects Traceability: Actual Organizational Resources to Actual Projects Mapping [PV-3]
Pj-Cn: Projects Connectivity [PV-2]
Pj-Sr: Projects Structure [PV-1]
Ps-Tx: Personnel Taxonomy [OV-4]
Pj-Tx: Projects Taxonomy [PV-1]
Ps-Pm: Personnel Forecast [SV-9]
Ps-Sr: Personnel Structure [OV-4]
Pj-Pr: Projects Processes: Flow [SV-4]

Step 8.2.1: Trace actual organizational resources to actual projects

Step 8.2.2: Specify project sequences and project milestone dependencies

Step 8.2: Capture project structure – including project milestone roles, status indicators, and responsible role kinds supported by project sequences

Figure 4:55 - Step 8.2: Capture Project Structure

4.8.3 Project Activity Behavior

Step 8.3 – Define Project Activity Behavior – Process flow diagrams are constructed for ***project activities***. These process flows include ***project activity actions*** classified by other ***project activities***, and also include process-flow mechanisms such as forks, signals, controls, flows, pins, connectors, etc.

- **Project Activity** – an activity carried out during a project
- **Resource Exchange** – asserts that a flow can exist between Resource Performers (i.e., flows of data, people, material, or energy)
- **Resource Exchange item** – an abstract grouping for elements that defines the types of elements that can be exchanged between Resource Performers and conveyed by a **Resource Exchange**

Project success ***measures*** are defined for the projects and their ***project activities***. Measures of overall project activities and projects should demonstrate satisfaction of ***project themes*** either directly through measures correlated to ***status indicators*** or indirectly through examination or correlation of ***project themes*** with the measures.

- **Measurement** – a property of an element representing something in the physical world, expressed in amounts of a unit of measure

A review is done using a ***project traceability*** which maps ***project activities*** to ***capabilities***. This serves as a basis from which completeness of support to capabilities can be understood.

Additionally, project structural relationships may be examined through intervening ***project activity actions*** to check for redesign or simplification as well as duplication. ***Project activities*** may be viewed either in complex meta-chain maps or in simple diagrams to ensure a complete understanding of ***project activities*** is known and understood. Projects may be arranged by activity groupings, the projects capable to perform them, actual organizations responsible for them, or some other useful organization.

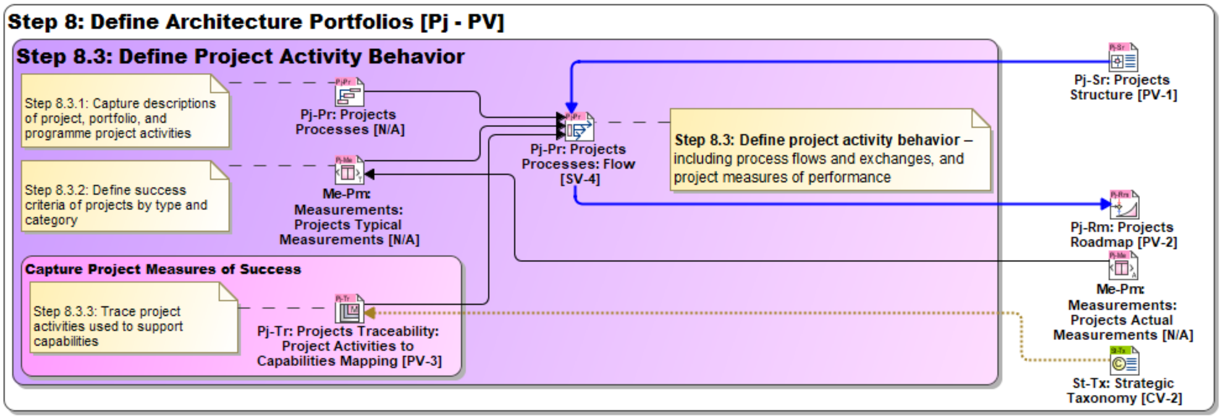

Step 8: Define Architecture Portfolios [Pj - PV]
Step 8.3: Define Project Activity Behavior
Step 8.3.1: Capture descriptions of project, portfolio, and programme project activities
Pj-Pr: Projects Processes [N/A]
Step 8.3.2: Define success criteria of projects by type and category
Me-Pm: Measurements: Projects Typical Measurements [N/A]
Capture Project Measures of Success
Step 8.3.3: Trace project activities used to support capabilities
Pj-Tr: Projects Traceability: Project Activities to Capabilities Mapping [PV-3]
Pj-Pr: Projects Processes: Flow [SV-4]
Step 8.3: Define project activity behavior – including process flows and exchanges, and project measures of performance
Pj-Sr: Projects Structure [PV-1]
Pj-Rm: Projects Roadmap [PV-2]
Me-Pm: Measurements: Projects Actual Measurements [N/A]
St-Tx: Strategic Taxonomy [CV-2]

Figure 4:56 - Step 8.3: Define Project Activity Behavior

4.8.4 Project Execution Activities

Step 8.4 – Manage Project Execution Activities – A project roadmap is assembled to show when ***actual project milestones*** occur on a timeline, and their associated ***project status*** in terms of the status indicators of the ***project themes***. This roadmap allows project managers and responsible owners understanding of when overall progress is made for deploying doctrine, training, leadership, materiel, and facilities, as well as organizations, personnel, or other items necessary to support capability deployment and evolution.

- **Project Status** – the status (i.e., level of progress) of a **Project Theme** for an **Actual Project** at the time of the **Actual Project Milestone**
- **Actual Project** – a time-limited endeavor to provide a specific set of Actual Resources that meet specific **Capability** needs (Note: i.e., an instance of a **Project** in the real world)
- **Actual Project Milestone** – an event with a start date in an **Actual Project** from which progress is measured

An entire project portfolio with all its elements (agents, behaviors, etc.) may be set as one option in an analysis of alternatives between others. Each alternative to be examined can be represented by a separate project portfolio. Typical project success measures from Step 8.3 may be used to generate ***actual measurements*** to either capture analysis of these alternatives or records of analysis for projects over time. Each actual measurement table is assigned start and end dates of their actual, estimated, or required points of existence and are used to track satisfaction of project success over time.

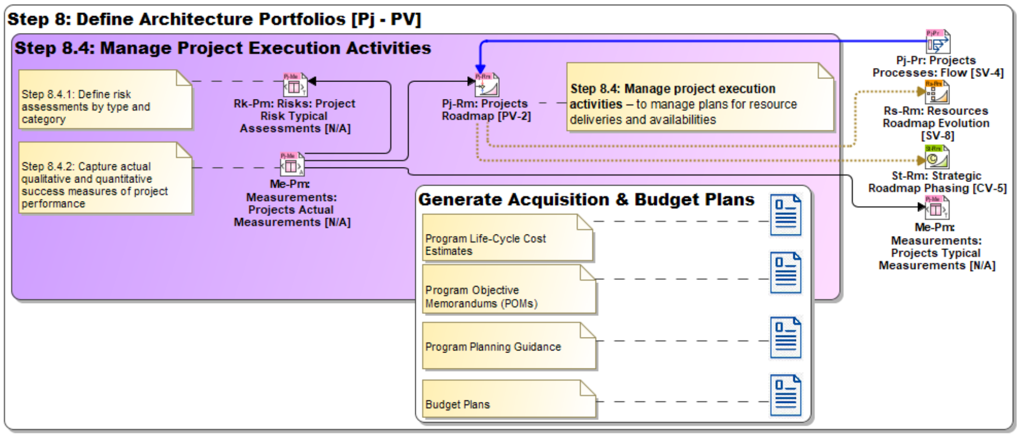
A ***risk*** analysis assesses the impact of events that may ***affect*** projects. Measures and ***actual measurements*** of these ***risks*** are included with the project measurement analysis.

- **Actual Measurement** – an actual value that is applied to a **Measurement** (Note: a measurement may be one of three kinds: actual, required, or estimate, and may have an associated start and/or end date)
- **Required** – an enumerated type of an **Actual Measurement Kind** which is based on a required value

If the project portfolio is used as a reference model to evaluate performance of managers responsible for the projects, ***actual measurements*** may form a basis of evaluation of them and of project effectiveness.

When needed, project, portfolio and program acquisition plans and budgeting plans can be developed based on the models and views generated during this Step 8. Ideally these documents are automatically generated from the model itself using reporting scripts and model queries. Additionally, the architectural description itself may be published or shared with project managers to communicate architectural planning guidance, budget guidance, program objectives.

When ***projects*** are shown embedded in architectural context, bidders and vendors have a better contextual understanding for the project context behind resource and other requirements and may use approved architectural descriptions to stem off their own project designs using SysML, UML and other languages compatible with a UAF based architecture.



Step 8: Define Architecture Portfolios [Pj - PV]
Step 8.4: Manage Project Execution Activities
Step 8.4.1: Define risk assessments by type and category
Rk-Pm: Risks: Project Risk Typical Assessments [N/A]
Step 8.4.2: Capture actual qualitative and quantitative success measures of project performance
Me-Pm: Measurements: Projects Actual Measurements [N/A]
Pj-Rm: Projects Roadmap [PV-2]
Step 8.4: Manage project execution activities – to manage plans for resource deliveries and availabilities
Pj-Pr: Projects Processes: Flow [SV-4]
Rs-Rm: Resources Roadmap Evolution [SV-8]
St-Rm: Strategic Roadmap Phasing [CV-5]
Me-Pm: Measurements: Projects Typical Measurements [N/A]
Generate Acquisition & Budget Plans
Program Life-Cycle Cost Estimates
Program Objective Memorandums (POMs)
Program Planning Guidance
Budget Plans

Figure 4:57 - Step 8.4: Manage Project Execution Activities

4.8.5 Architecture View Summary for Step 8

The view specifications in UAF for this viewpoint are outlined here:

	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Projects Pj	-	Projects Taxonomy Pj-Tx	Projects Structure Pj-Sr	Projects Connectivity Pj-Cn	Projects Processes Pj-Pr	-	-	-	**Envmt, Msmts, Risks (En-Pm, Me-Pm, Rk-Pm)**	-	Projects Roadmap Pj-Rm	Projects Traceability Pj-Tr

A summary of the lower-level steps and UAF views is shown below. The architecture views produced by each step are identified by the UAF grid designator Aa-Bb where Aa represents the Viewpoint row in the grid and Bb represents the Aspect column in the grid. DoDAF view designations are also shown in [brackets], when applicable. The name of the architecture view matches the name of the view specification in the UAFML. In some cases, a subtitle is provided to identify what kind of information is provided in an instance of that UAF view. There will be some architecture views that are not part of the UAF specification since you will sometimes need fit-for-purpose (i.e., custom) views that capture the necessary architectural information.

222	Step 8: Manage Project Portfolios [Pj - PV]	Views
223	**Step 8.1: Establish arrange project taxonomy** – of projects and milestones to support design of projects, portfolios, and programmes that support delivery objectives for the enterprise	Pj-Tx: Projects Taxonomy [PV-1]
224	Step 8.1.1: Trace capabilities to actual projects, portfolios, and programmes	Pj-Tr: Projects Traceability: *Actual Projects to Capabilities Mapping* [PV-3]
225	Step 8.1.2: Capture types and categories of programs, portfolios, and programmes	Pj-Tx: Projects Taxonomy: *Table* [SV-1]
226	**Step 8.2: Capture project structure** – including project milestone roles, status indicators, and responsible role kinds supported by project sequences	Pj-Sr: Projects Structure [PV-1]
227	Step 8.2.1: Trace actual organizational resources to actual projects	Pj-Tr: Projects Traceability: *Actual Organizational Resources to Actual Projects Mapping* [PV-3]

228	Step 8.2.2: Specify project sequences and project milestone dependencies	Pj-Cn: Projects Connectivity [PV-2]
229	**Step 8.3: Define project activity behavior** – including process flows and exchanges, and project measures of performance	Pj-Pr: Projects Processes: *Flow* [N/A]
230	Step 8.3.1: Capture descriptions of project, portfolio, and programme project activities	Pj-Pr: Projects Processes [N/A]
231	Step 8.3.2: Define success criteria of projects by type and category	Me-Pm: Measurements: *Projects Typical Measurements* [N/A]
231	Step 8.3.3: Trace project activities used to support capabilities	Pj-Tr: Projects Traceability: *Project Activities to Capabilities Mapping* [PV-3]
233	**Step 8.4: Manage project execution activities** – to manage plans for projects affecting capabilities and resource deliveries and availabilities	Pj-Rm: Projects Roadmap [PV-2]
234	Step 8.4.1: Define risk assessments by type and category	Rk-Pm: Risks: *Project Risk Typical Assessments* [N/A]
235	Step 8.4.2: Capture actual qualitative and quantitative success measures of project performance	Me-Pm: Measurements: *Projects Actual Measurements* [N/A]

STEP 9 – Actual Resources Realization

4.9 Step 9 – Actual Resources Realization

Purpose. The purpose of this step is to illustrate the expected or achieved actual resource configurations and actual relationships between them. This step also entails the identification of technical, operational, and business standards applicable to the architecture and defining the underlying current and expected standards. Key stakeholders for this step are Solution Providers, Systems Engineers, Business Architects and Human Resources. Their concerns are mainly about the analysis of different alternatives, what-if scenarios, architectural tradeoffs, and the verification and validation of the actual resource configurations. They also have concerns regarding the technical and non-technical standards applicable to the architecture.

Conceptual Schema. The key concepts used in the Actual Resources Viewpoint that can be used as model elements in the architecture views and the relationships between these concepts are illustrated in the conceptual schema shown in Figure 4:58. These ***key concepts*** are highlighted in italics within the Narrative and some of the less obvious concepts are listed with the associated UAF meaning of that concept. Detailed definition of the entities and relationships shown in the conceptual schema are provided in the UAFML specification document.

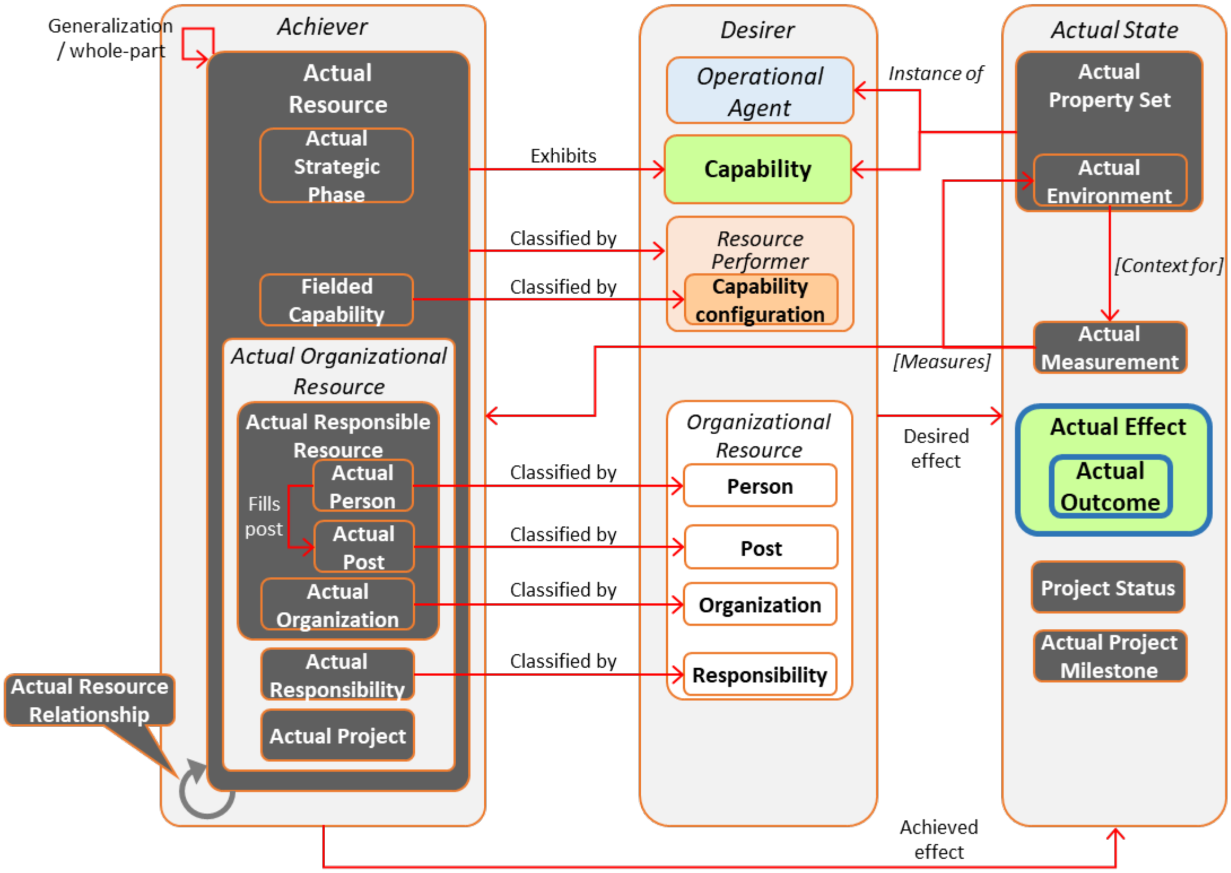


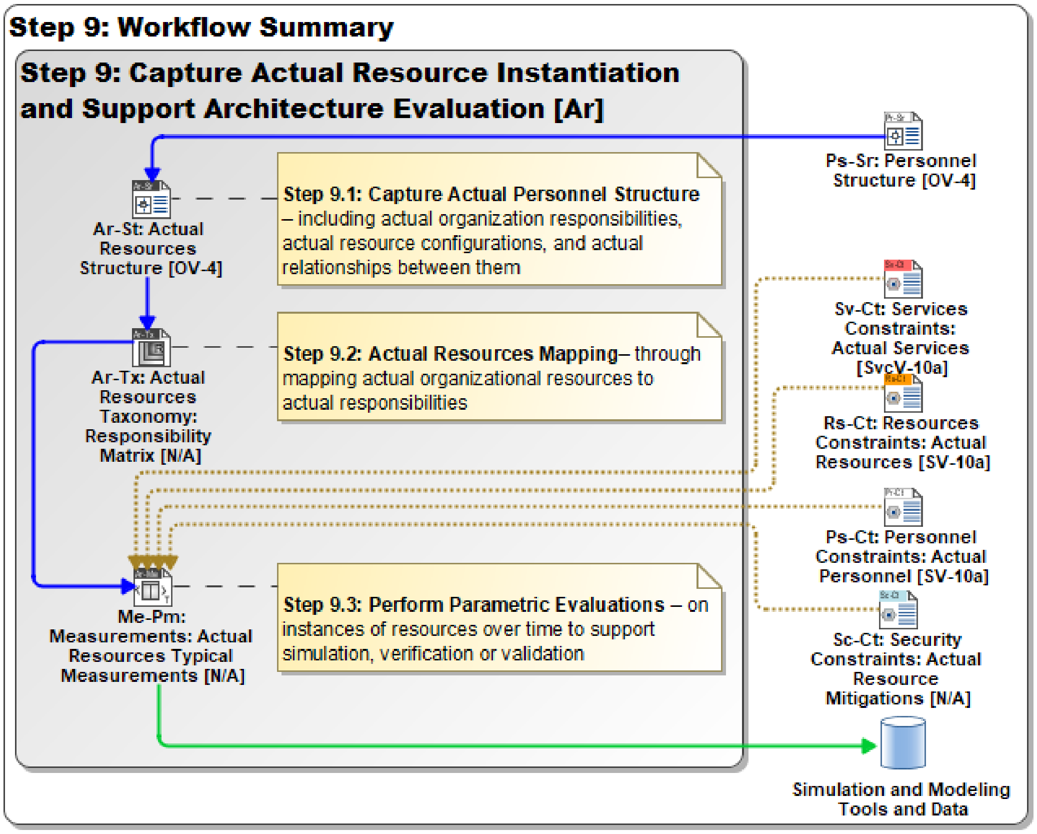
Figure 4:58 - Conceptual Schema for Actual Resources Realization

The actual resource concepts are shown in gray and the affected architecture concepts are shown by the color legend. ***Actual resources***, ***actual organizational resources*** and ***fielded capabilities*** will exhibit ***capabilities***.

This Guide is intended to be used in conjunction with the UAF Sample Problem that defines architecture views for a Search and Rescue Mission. UAF Specification documents (including the Sample Problem) can be downloaded from the OMG webpage: www.omg.org/spec/UAF/About-UAF/.

Step 9.0 – Capture Actual Resource Instantiation and Support Architecture Evaluation – The main entry criterion for this Step is bringing forward the resource architectures from Step 5 and personnel architectures from Step 6 with associated operational ***implementation*** context.

Workflow Summary. A summary level view of the six steps involved in Step 9 workflow is shown below. The flowlines represent how one architecture view "influences" another architecture view. These lines do not represent a particular sequence of process activity execution or imply information exchanges. These should be thought of as influence diagrams rather than process diagrams. These architecture views (e.g., views, diagrams, tables) can be mapped to the architecture processes used in your organization and can be incorporated into an architecture modeling methodology.



Step 9: Workflow Summary
Step 9: Capture Actual Resource Instantiation and Support Architecture Evaluation [Ar]
Ar-St: Actual Resources Structure [OV-4]
Step 9.1: Capture Actual Personnel Structure – including actual organization responsibilities, actual resource configurations, and actual relationships between them
Ar-Tx: Actual Resources Taxonomy: Responsibility Matrix [N/A]
Step 9.2: Actual Resources Mapping– through mapping actual organizational resources to actual responsibilities
Me-Pm: Measurements: Actual Resources Typical Measurements [N/A]
Step 9.3: Perform Parametric Evaluations – on instances of resources over time to support simulation, verification or validation
Ps-Sr: Personnel Structure [OV-4]
Sv-Ct: Services Constraints: Actual Services [SvcV-10a]
Rs-Ct: Resources Constraints: Actual Resources [SV-10a]
Ps-Ct: Personnel Constraints: Actual Personnel [SV-10a]
Sc-Ct: Security Constraints: Actual Resource Mitigations [N/A]
Simulation and Modeling Tools and Data

Figure 4:59 - Workflow Summary for Step 9: Capture Actual Resource Instantiation, etc.

The second-level steps in the Step 9 workflow are listed below along with the UAF views (and corresponding [DoDAF views] where applicable) associated with those steps. A complete list of the detailed steps at the third level with their corresponding views is provided at the end of this section.

236	**Step 9: Capture Actual Resource Instantiation and Support Architecture Evaluation [Ar]**	**Views**
237	**Step 9.1: Capture Actual Personnel Structure** – including actual organization responsibilities, actual resource configurations, and actual relationships between them	Ar-Sr: Actual Resources Structure [OV-4, SV-1/2]

240	**Step 9.2: Map Actual Organizational Resources to Actual Responsibilities** – through mapping actual resources to capabilities they exhibit	Ar-Tx: Actual Resources Taxonomy: *Responsibility Matrix* [N/A]
241	**Step 9.3: Perform Parametric Evaluations** – on instances of resources over time to support simulation, verification or validation	Me-Pm: Measurements: *Actual Resources Typical Measurements* [N/A]

4.9.1 Actual Personnel Structure

Step 9.1 – Capture Actual Personnel Structure – Existing actual organizational resources that have been preliminarily identified from Step 6.1 through 6.2 are brought forward, and finalization of all ***actual resource*** instances from Steps 4 through 7 are made. Actual persons that fill actual posts are captured and actual responsibilities that are assigned to actual resources are captured.

- **Actual Resource** – an individual, fully-realized Resource Performer (Note: this is an instance of a Resource in the real world, including a **Fielded Capability** or an Actual Organizational Resource)
- **Fielded Capability** – an individual, fully-realized **Capability** (Note: this is an instance of a **Capability Configuration** in the real world)
- **Actual Organization** – an actual formal or informal organizational unit, e.g., "Driving and Vehicle Licensing Agency", "UAF team Alpha" (Note: this is an instance of an **Organization** in the real world)
- **Actual Person** – an individual human being (Note: this is an instance of a **Person** in the real world)
- **Actual Post** – an actual, specific post, an instance of a Post "type" - e.g., "President of the United States of America" where the Post would be president (Note: this is an instance of a **Post** in the real world)
- **Fills Post** – a tuple that asserts that an **Actual Person** fills an **Actual Post**
- **Actual Responsibility** – an actual duty required of a **Person** or Organization (Note: this is an instance of a **Responsibility** in the real world)
- **Responsible For** – a tuple between an **Actual Responsible Resource** and an **Actual Responsibility** or **Actual Project**. It defines the duties that the **Actual Responsible Resource** is **Responsible For**.

Next, ***actual resource relationships*** are created between ***actual organizational resources*** and ***actual resources***, including ***fielded capabilities***. These relationships capture the realization of ***resource exchanges*** that are conveying the flow of information, data, people, materiel or energy, to assist in verification of those exchanges.

- **Actual Resource Relationship** – an abstract element that details the Actual Organizational Resources that are able to carry out an **Actual Responsibility**

The complexity of verifying or validating ***actual responsibility*** roles of the actual organizations and actual posts is sometimes better understood through the use of a trace matrix for the actual resource structure.

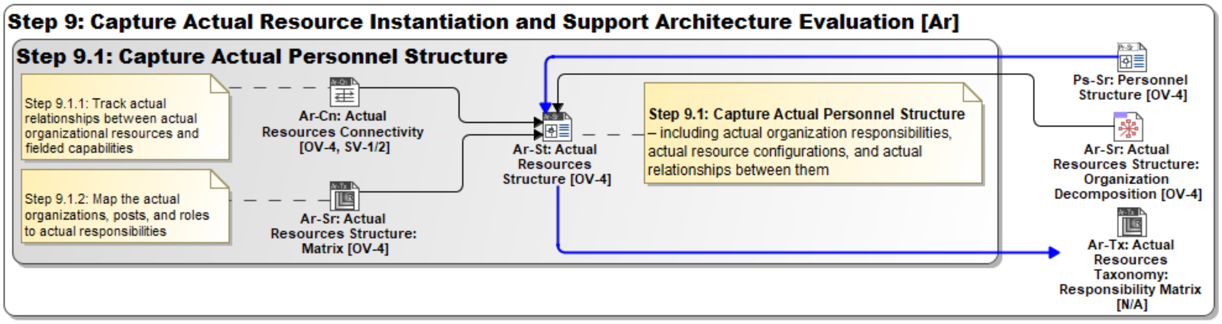

Step 9: Capture Actual Resource Instantiation and Support Architecture Evaluation [Ar]
Step 9.1: Capture Actual Personnel Structure
Step 9.1.1: Track actual relationships between actual organizational resources and fielded capabilities
Step 9.1.2: Map the actual organizations, posts, and roles to actual responsibilities
Ar-Cn: Actual Resources Connectivity [OV-4, SV-1/2]
Ar-Sr: Actual Resources Structure: Matrix [OV-4]
Ar-St: Actual Resources Structure [OV-4]
Step 9.1: Capture Actual Personnel Structure – including actual organization responsibilities, actual resource configurations, and actual relationships between them
Ps-Sr: Personnel Structure [OV-4]
Ar-Sr: Actual Resources Structure: Organization Decomposition [OV-4]
Ar-Tx: Actual Resources Taxonomy: Responsibility Matrix [N/A]

Figure 4:60 - Step 9.1: Capture Actual Personnel Structure

4.9.2 Actual Resources Mapping

Step 9.2 – Map Actual Organizational Resources to Actual Responsibilities – A resource mapping matrix is generated to depict the mapping of ***actual organizational resources*** to ***actual responsibilities*** which will identify the transformation of operational needs into purposeful responsibilities performed by organizational resource solutions.

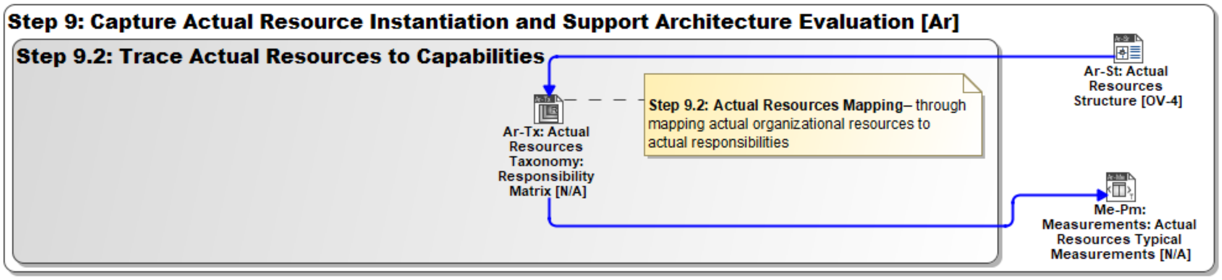

Step 9: Capture Actual Resource Instantiation and Support Architecture Evaluation [Ar]
Step 9.2: Trace Actual Resources to Capabilities
Ar-Tx: Actual Resources Taxonomy: Responsibility Matrix [N/A]
Step 9.2: Actual Resources Mapping– through mapping actual organizational resources to actual responsibilities
Ar-St: Actual Resources Structure [OV-4]
Me-Pm: Measurements: Actual Resources Typical Measurements [N/A]

Figure 4:61 - Step 9.2: Map Actual Organizational Resources to Actual Responsibilities

4.9.3 Actual Resource Analysis

Step 9.3 – Perform Parametric Evaluations – Initial actual resources captured in Steps 4 through 7 are brought forward and instances are made of resources that need evaluation, verification or validation. This includes associated ***actual measurements*** and parametric models. ***Typical measures*** from resources are used to generate ***actual measurements*** to either capture points in time that the actual resource exists or to evaluate it against resource designs. Actual resources may be captured with actual measurements in tables with assigned start and end dates of their ***actual*** or ***estimated*** measurement values and are used to track verification and validation of those actual resources over time. As necessary, the parametric models in the various resource constraints views may aid in understanding of computed measures.

- **Actual Measurement** – an actual value that is applied to a **Measurement** (Note: a measurement may be one of three kinds: actual, required, or estimate, and may have an associated start and/or end date)
- **Required** – an enumerated type of an **Actual Measurement Kind** which is based on a required value
- **Actual** – an enumerated type of an **Actual Measurement Kind** which is based on an actual value
- **Estimate** – an enumerated type of an **Actual Measurement Kind** which is based on an estimated value

Analysis using other simulation and evaluation tools and mechanisms can be conducted to aid in providing actual measurement numbers when needed. These evaluation tools may include the capture of live or reported operations metrics, the results of analysis on mission performance of the resources, as well as what-if scenarios analysis, including wargaming, role-playing, and business or process simulations.

When needed, verification, validation and mission or business simulation plans can be developed based on the models and views generated during this Step 9. Ideally these documents are automatically generated from the model itself using reporting scripts and model queries. Additionally, the architectural description itself may be published or shared as a permanent record to communicate architecture performance to decision makers, designers, stakeholders, and inform emphasis areas for future architectural changes or requests for proposals.

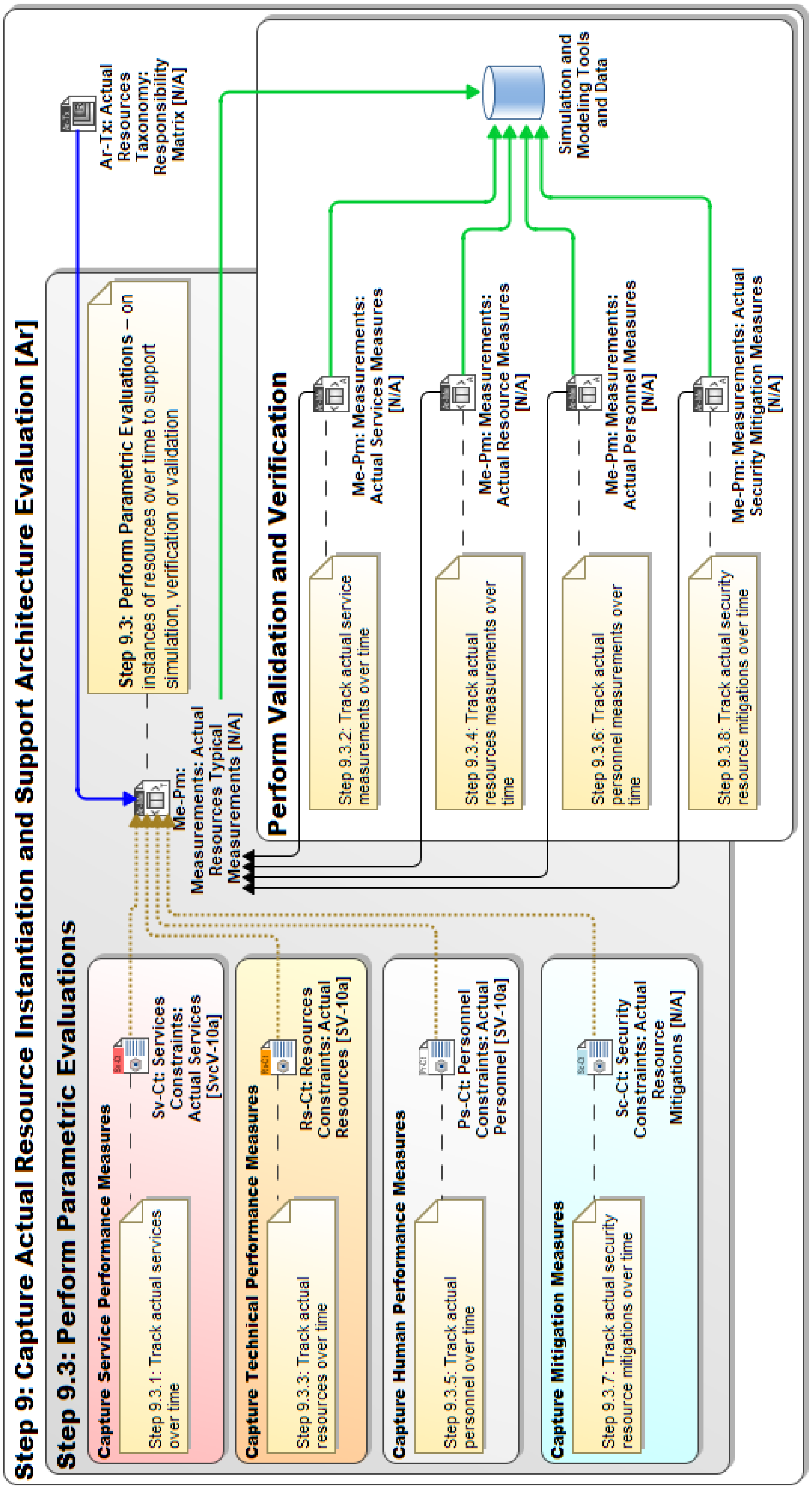

Step 9: Capture Actual Resource Instantiation and Support Architecture Evaluation [Ar]
Step 9.3: Perform Parametric Evaluations
Step 9.3: Perform Parametric Evaluations – on instances of resources over time to support simulation, verification or validation
Ar-Tx: Actual Resource Taxonomy: Responsibility Matrix [N/A]
Me-Pm: Measurements: Actual Resources Typical Measurements [N/A]
Capture Service Performance Measures
Step 9.3.1: Track actual services over time
Sv-Ct: Services Constraints: Actual Services [SvcV-10a]
Capture Technical Performance Measures
Step 9.3.3: Track actual resources over time
Rs-Ct: Resources Constraints: Actual Resources [SV-10a]
Capture Human Performance Measures
Step 9.3.5: Track actual personnel over time
Ps-Ct: Personnel Constraints: Actual Personnel [SV-10a]
Capture Mitigation Measures
Step 9.3.7: Track actual security resource mitigations over time
Sc-Ct: Security Constraints: Actual Resource Mitigations [N/A]
Perform Validation and Verification
Step 9.3.2: Track actual service measurements over time
Me-Pm: Measurements: Actual Services Measures [N/A]
Step 9.3.4: Track actual resources measurements over time
Me-Pm: Measurements: Actual Resource Measures [N/A]
Step 9.3.6: Track actual personnel measurements over time
Me-Pm: Measurements: Actual Personnel Measures [N/A]
Step 9.3.8: Track actual security resource mitigations over time
Me-Pm: Measurements: Actual Security Mitigation Measures [N/A]
Simulation and Modeling Tools and Data

Figure 4:62 - Step 9.3: Perform Parametric Evaluations

4.9.4 Architecture View Summary for Step 9

The view specifications in UAF for this viewpoint are outlined here:

	Moti-vation Mv	Taxo-nomy Tx	Struc-ture Sr	Connec-tivity Cn	Pro-cesses Pr	States St	Sequ-ences Sq	Informa-tion If	Para-meters Pm	Con-straints Ct	Road-map Rm	Trace-ability Tr
Actual Resources Ar	-	-	Actual Resources Structure Ar-Sr	Actual Resources Connectivity Ar-Cn	Simulation			-	-	Parametric Execution/ Evaluation	-	-

A summary of the lower-level steps and UAF views is shown below. The architecture views produced by each step are identified by the UAF grid designator Aa-Bb where Aa represents the Viewpoint row in the grid and Bb represents the Aspect column in the grid. DoDAF view designations are also shown in [brackets], when applicable. The name of the architecture view matches the name of the view specification in the UAFML. In some cases, a subtitle is provided to identify what kind of information is provided in an instance of that UAF view. There will be some architecture views that are not part of the UAF specification since you will sometimes need fit-for-purpose (i.e., custom) views that capture the necessary architectural information.

236	**Step 9: Capture Actual Resource Instantiation and Support Architecture Evaluation [Ar]**	**Views**
237	**Step 9.1: Capture Actual Personnel Structure** – including actual organization responsibilities, actual resource configurations, and actual relationships between them	Ar-Sr: Actual Resources Structure [OV-4]
238	Step 9.1.1: Track actual relationships between actual organizational resources and fielded capabilities	Ar-Cn: Actual Resources Connectivity [OV-4, SV-1/2]
239	Step 9.1.2: Map the actual organizations, posts, and roles to actual responsibilities	Ar-Sr: Actual Resources Structure: *Matrix* [OV-4]
240	**Step 9.2: Map Actual Organizational Resources to Actual Responsibilities** – through mapping actual resources to capabilities they exhibit	Ar-Tx: Actual Resources Taxonomy: *Responsibility Matri*x [N/A]
241	**Step 9.3: Perform Parametric Evaluations** – on instances of resources over time to support simulation, verification or validation	Me-Pm: Measurements: *Actual Resources Typical Measurements* [N/A]
242	Step 9.3.1: Track actual services over time	Sv-Ct: Services Constraints: *Actual Services* [SvcV-10a]
243	Step 9.3.2: Track actual service measurements over time	Me-Pm: Measurements: *Actual Services Measures* [N/A]
244	Step 9.3.3: Track actual resources over time	Rs-Ct: Resources Constraints: *Actual Resources* [SV-10a]
245	Step 9.3.4: Track actual resources measurements over time	Me-Pm: Measurements: *Actual Resource Measures* [N/A]
246	Step 9.3.5: Track actual personnel over time	Ps-Ct: Personnel Constraints: *Actual Personnel* [SV-10a]
247	Step 9.3.6: Track actual personnel measurements over time	Me-Pm: Measurements: *Actual Personnel Measures* [N/A]
248	Step 9.3.7: Track actual security resource mitigations over time	Sc-Ct: Security Constraints: *Actual Resource Mitigations* [N/A]
249	Step 9.3.8: Track actual security resource mitigations over time	Me-Pm: Measurements: *Actual Security Mitigation Measures* [N/A]

This page intentionally left blank.

A Appendix A – Architecture Management and Description

A.1 Architecture Management Concepts

Conceptual Schema. The key concepts used in the Architecture Management Viewpoint that can be used as model elements in the architecture views and the relationships between these concepts are illustrated in the conceptual schema shown in Figure A:1. These *key concepts* are highlighted in italics within the Narrative and some of the less obvious concepts are listed with the associated ISO-42010[9] *meaning* or the UAF ***meaning*** of that concept. Detailed definition of the entities and relationships shown in the conceptual schema are provided in the UAFML specification document.

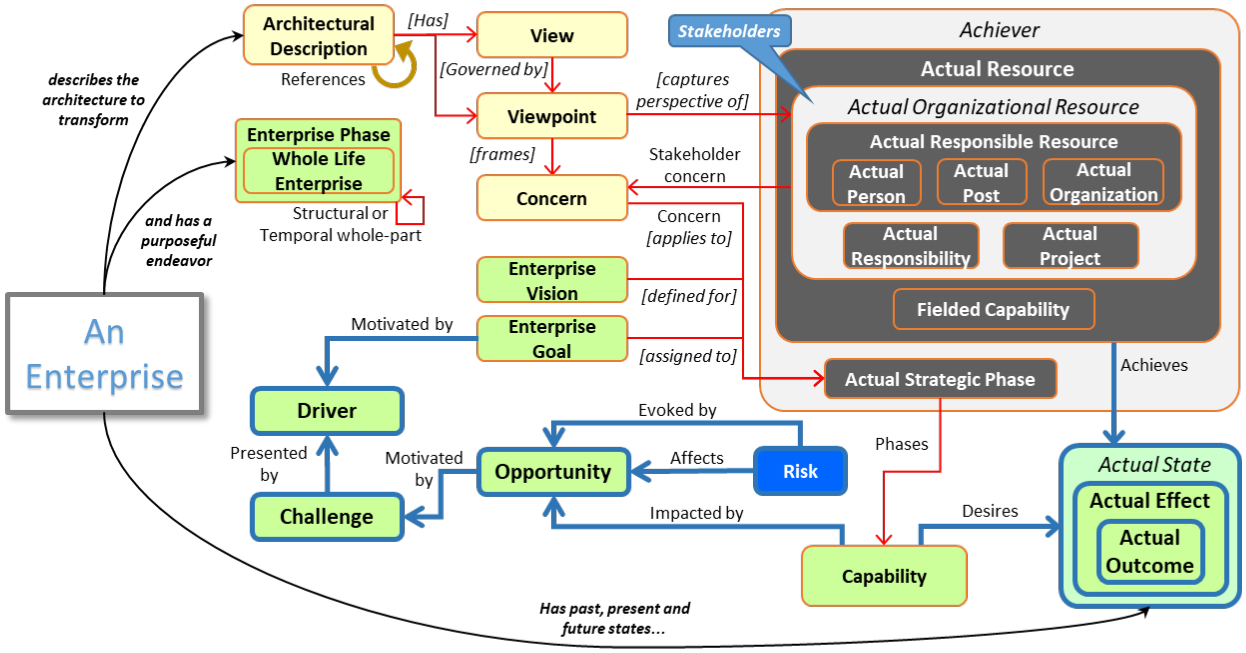


Figure A:1 - Conceptual Schema for Architecture Management

An enterprise is a purposeful endeavor with an established ***vision*** to achieve its stated ***enterprise goals***. The enterprise will encounter strategic ***drivers*** that present ***challenges*** to the organizations that participate in the enterprise, which in turn will motivate the enterprise to pursue ***opportunities*** that address these challenges. The ***capabilities*** of the enterprise (or capabilities deployed to others for their own use) will be impacted by the opportunities to be pursued. The current or future capabilities will help achieve a series of ***effects*** that in the end will achieve some desired ***outcomes***. The enterprise will typically structure its transformation efforts into ***strategic phases*** that will endeavor to exhibit the desired ***capabilities***. Also, notice that the enterprise has had previous states and has current states that will affect how it currently responds to changes and to internal and external perturbations. This “history” of states must be taken into account when planning to transform the enterprise.

[9] ***ISO/IEC/IEEE 42010:2021 Software, systems and enterprise – Architecture description*** establishes a standard approach to describing an architecture using views and viewpoints, architecture description languages and architecture descriptions frameworks.

The enterprise can use an architecture framework (such as UAF) as the basis for developing a set of ***architectural descriptions*** to help transform the enterprise by setting new or modified ***enterprise goals***.

A.2 Architecture Description Concepts

Each architectural description will be composed of architecture ***views*** which are governed by defined ***viewpoints*** (typically captured as standard viewpoint specifications in an architecture description framework). These ***viewpoints*** address particular stakeholder ***concerns*** of ***organizational resources*** (either inside or external to the enterprise). These organizational resources will have a desire to achieve certain states in terms of capability-driven ***effects*** or desired ultimate ***outcomes***.

Figure A:2 shows the key concepts defined in the ISO 42010 standard on Architecture Description.

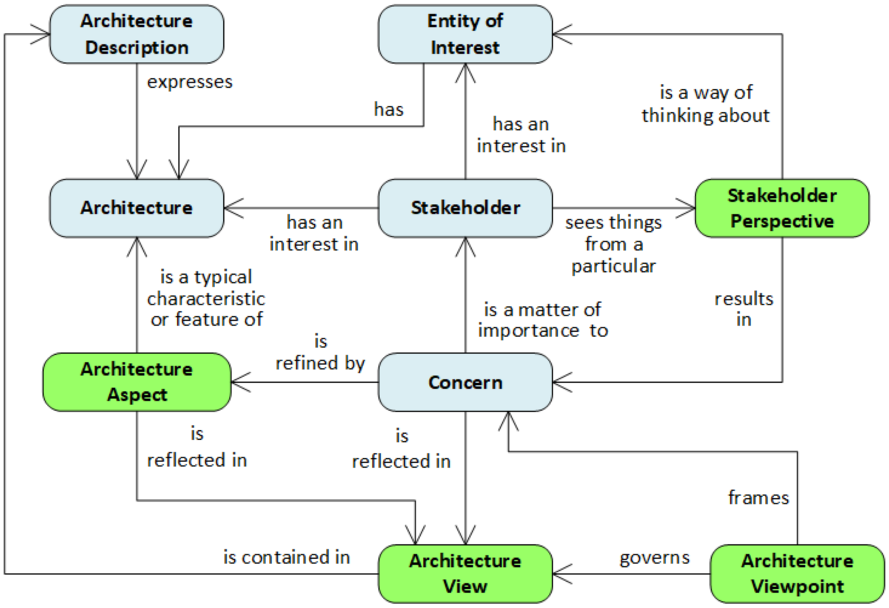


Figure A:2 - Architecture Description Concepts

The *architecture* is a set of fundamental concepts and properties of an *entity of interest*. In our case, the entity of interest is typically an enterprise, although it could also be a system, product or service depicted from an enterprise perspective. Architectures are expressed in *architecture descriptions* to inform *stakeholders* who have an interest in the architecture. These descriptions facilitate decision making concerning the enterprise and how best to meet its missions, goals, and objectives. An architecture description is composed of *architecture views* which are governed by *architecture viewpoints* that frame *concerns* reflected in those *views*.

Stakeholders see things from a particular *perspective* which results in the concerns (i.e., interests) they have regarding the architecture. An enterprise *aspect* is a “mode of characterization” of those concerns that is used to explore different dimensions of the problem and solution spaces. Examples of aspects to address are motivational, structural, behavioral, informational, evolutionary timelines, taxonomic hierarchies, and measurements. Stakeholder *perspectives* correspond to viewpoints in the UAF grid while enterprise *aspects* correspond to Aspects in the UAF grid.

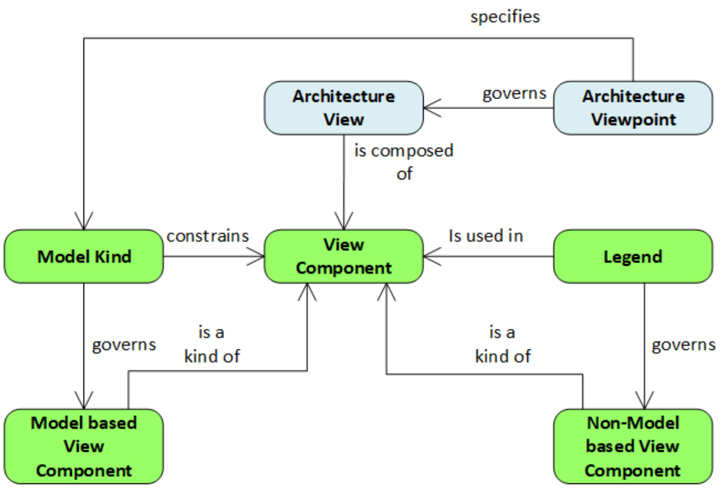


Figure A:3 - Architecture Views and View Components

As illustrated in Figure A:3, *architecture views* are composed of *view components* that can include a variety of *Aspects* to capture the information necessary within the architecture description to address the *concerns* of *stakeholders*. The view components can be either based on models built using a modeling language (such as the UAFML or SysML) or based on other sources of non-model-based data that is used to populate the views, such as project schedules, accounting tools, risk lists, analytical tools, mapping databases, material resource planning tools, etc.

A.3 Architecture Enablers – Overview of Step 0

Step 0.0 – Define Reference Architecture, Framework and Architecture Enablers – The main entry criterion for this Step is reaching a decision regarding the purpose of the architecture description, partly based on the concerns of the primary *stakeholders* who have an interest in the *architecture*. These *stakeholders* are often business or government leaders who have an interest in transformation of the enterprise to become more effective and efficient in achieving its goals and objectives.

- *Architecture* – fundamental concepts or properties related to an entity in its environment and governing principles for the realization and evolution of this entity and its related life cycle processes
- *Architecture Description* – work product used to express an architecture
- *Stakeholder* – an individual, team, organization, or classes thereof, having an interest in a **Strategic Phase** [ISO/IEC/IEEE 42010:2011]
- *Enterprise* – human undertaking or venture that has a mission, goals, and objectives to offer products or services or to achieve a desired project or business outcome

The enterprise architecture description is used by stakeholders to improve communication and cooperation among affected parties and enable them to work together in a more integrated, coherent fashion. This will, in turn, help the enterprise more effectively achieve its goals. This can be facilitated by creating a “reference architecture” that guides development of the rest of the enterprise architecture in Steps 3-7, as well as using an architecture framework that defines the views to be used.

This organizational framework can be tailored from the UAF by choosing the relevant views, modifying them where appropriate, and defining new views that are needed to express the key concepts and properties of the enterprise architecture. In addition, modeling templates, patterns and methods will be needed to help conduct and manage the architecture development efforts; these items are called “architecture enablers” in this Guide and are associated with the Architecture Enablement process in ISO 42020.

Workflow Summary. A summary level view of the six steps involved in Step 0 workflow is shown in Figure A:4. The flowlines represent how one architecture view “influences” another architecture view. These lines do not represent a particular sequence of process activity execution or imply information exchanges. These should be thought of as influence diagrams rather than process diagrams. These architecture views (e.g., views, diagrams, tables) can be mapped to the architecture processes used in your organization and can be incorporated into an architecture modeling methodology.

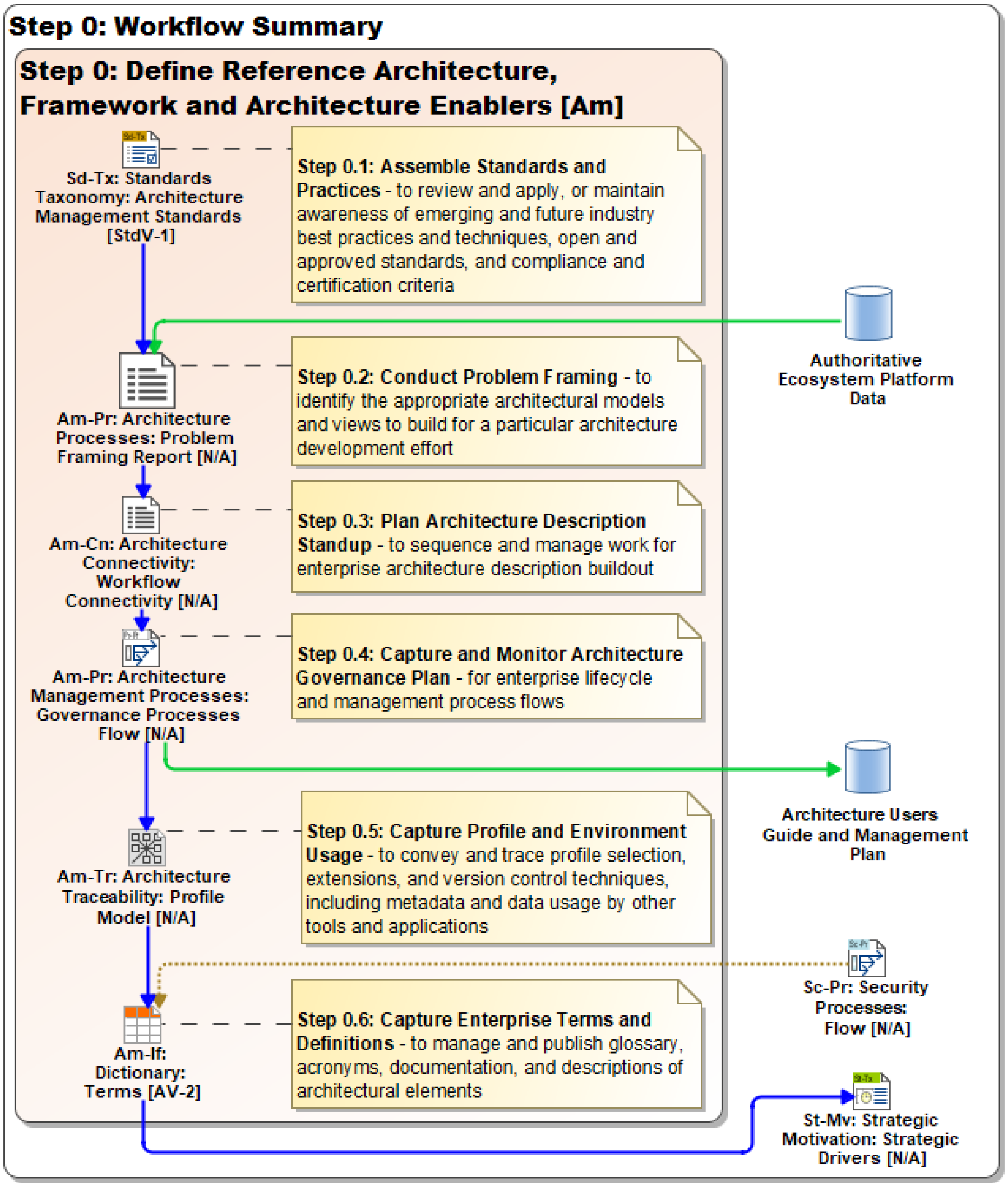

Step 0: Workflow Summary
Step 0: Define Reference Architecture, Framework and Architecture Enablers [Am]
Sd-Tx: Standards Taxonomy: Architecture Management Standards [StdV-1]
Step 0.1: Assemble Standards and Practices - to review and apply, or maintain awareness of emerging and future industry best practices and techniques, open and approved standards, and compliance and certification criteria
Authoritative Ecosystem Platform Data
Am-Pr: Architecture Processes: Problem Framing Report [N/A]
Step 0.2: Conduct Problem Framing - to identify the appropriate architectural models and views to build for a particular architecture development effort
Am-Cn: Architecture Connectivity: Workflow Connectivity [N/A]
Step 0.3: Plan Architecture Description Standup - to sequence and manage work for enterprise architecture description buildout
Am-Pr: Architecture Management Processes: Governance Processes Flow [N/A]
Step 0.4: Capture and Monitor Architecture Governance Plan - for enterprise lifecycle and management process flows
Architecture Users Guide and Management Plan
Am-Tr: Architecture Traceability: Profile Model [N/A]
Step 0.5: Capture Profile and Environment Usage - to convey and trace profile selection, extensions, and version control techniques, including metadata and data usage by other tools and applications
Sc-Pr: Security Processes: Flow [N/A]
Am-If: Dictionary: Terms [AV-2]
Step 0.6: Capture Enterprise Terms and Definitions - to manage and publish glossary, acronyms, documentation, and descriptions of architectural elements
St-Mv: Strategic Motivation: Strategic Drivers [N/A]

Figure A:4 - Workflow Summary for Define Reference Architecture, Framework, etc.

The second-level steps in Step 0 are listed below along with the UAF views (and corresponding [DoDAF views] where applicable) associated with those steps. A complete list of the detailed steps at the third level with their corresponding views is provided at the end of the next section.

1	**Step 0: Define Reference Architecture, Framework and Architecture Enablers [Am]**	**Views**
2	**Step 0.1: Assemble Standards and Practices** – to review and apply, or maintain awareness of emerging and future best industry practices and techniques, open and approved standards, and compliance and certification criteria	Sd-Tx: Standards Taxonomy: *Architecture Management Standards* [StdV-1]
5	**Step 0.2: Conduct Problem Framing** – to identify the appropriate architectural models and views to build for a particular architecture development effort	Am-Pr: Architecture Development Method: *Problem Framing* Report [N/A]
9	**Step 0.3: Plan Architecture Description Standup** – to sequence and manage work for enterprise architecture description buildout	Am-Cn: Architecture Views: *Workflow Connectivity* [N/A]
15	**Step 0.4: Capture and Monitor Architecture Governance** – for enterprise lifecycle and management process flows	Am-Pr: Architecture Management Processes: *Governance Processes Flow* [N/A]
22	**Step 0.5: Capture Profile and Environment Usage** – to convey and trace profile selection, extensions, and version control techniques, including metadata and data usage by other tools and applications	Am-Tr: Architecture Traceability: *Profile Model* [N/A]
26	**Step 0.6: Capture Enterprise Terms and Definitions** – to manage and publish glossary, definitions, acronyms, documentation, and descriptions of architectural elements	Am-If: Dictionary: *Terms* [AV-2]

A.4 Architecture Enablers Development Workflow Details

A.4.1 Standards and Practices

Step 0.1 – Assemble Standards and Practices – Existing organizational standards are brought forward, and decisions are made regarding which industry standards for architecture to adopt and apply. These standards are used in order to guide an enterprise organization's body of knowledge, practices, and cultural framework to enable the organization's decision making for enterprise and system life cycle architecting activities, including strategic planning, budgeting, and evaluation of performance, risk, schedule and cost. Key overarching architecture standards include the following:

- **OMG Unified Architecture Framework** – defines ways of representing an enterprise architecture that enables stakeholders to focus on specific areas of interest in the enterprise
- ISO/IEC/IEEE 42010:2021 – *Software, systems and enterprise – Architecture description*: provides core terms, definitions and relationships for architecture descriptions
- **ISO/IEC/ IEEE 42020:2019 – *Software, systems and enterprise – Architecture processes*:** establishes a set of process descriptions for the governance and management of a collection of architectures and the architecting of the entities
- **ISO/IEC/ IEEE 42030:2019 – *Software, systems and enterprise – Architecture evaluation framework***: specifies the means to organize and record architecture evaluations for enterprise, systems and software fields of application
- ISO/IEC 15704:2019 – *Enterprise modelling and architecture – Requirements for enterprise-referencing architectures and methodologies*: specifies a reference base of concepts and principles for enterprise architectures that enable enterprise development, enterprise integration, enterprise interoperability, human understanding, and computer processing
- **ISO/IEC 19510** – ***Information technology – Object Management Group Business Process Model and Notation***: provides a notation that is readily understandable by all business users to bridge business process design and implementation

An enterprise organization's complete standards profile may include many other standards applicable to enterprise governance, systems and software engineering life cycle management, project management, and their supporting technical processes. As architecting and other life cycle processes are tailored and used, these standards are traced to those processes, as well as supporting tools, software, and procurement or acquisition contracts.

Standardized *architecture description frameworks* are used to help codify the conventions and common practices of architecting and the description of architectures within different communities and domains of application in the enterprise. An enterprise organization may blend concepts from multiple architecture frameworks to develop a tailored framework which facilitates the enterprise stakeholders decision-making processes.

- *Architecture Description Framework* – conventions, principles and practices for the description of architectures established within a specific domain of application or community of stakeholders

Standard architecture frameworks such as DOD Architecture Framework (DoDAF), NATO Architecture Framework (NAF) and UK's MOD Architecture Framework (MODAF) are examples of frameworks supported through the Unified Architecture Framework (UAF), which enables efficient integration or translation of architecture information, and *correspondence* rules between different enterprises which use different frameworks or their elements in their own internal architecting practices.

- *Correspondence* – expression of relationship among architecture description elements or among architecture descriptions

As the organization uses these standards, and as best practices are learned, adopted, and modified, feedback is captured internally and shared within the enterprise culture, as well as with the standards development organizations that are responsible for developing and publishing these standards.

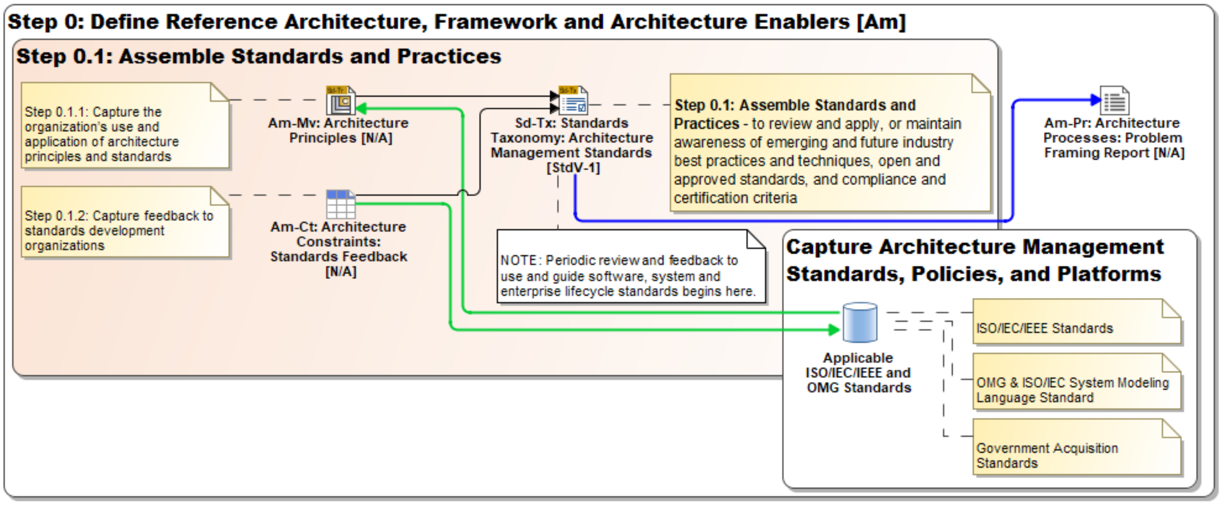

Step 0: Define Reference Architecture, Framework and Architecture Enablers [Am]
Step 0.1: Assemble Standards and Practices
Step 0.1.1: Capture the organization's use and application of architecture principles and standards
Step 0.1.2: Capture feedback to standards development organizations
Am-Mv: Architecture Principles [N/A]
Am-Ct: Architecture Constraints: Standards Feedback [N/A]
Sd-Tx: Standards Taxonomy: Architecture Management Standards [StdV-1]
Step 0.1: Assemble Standards and Practices - to review and apply, or maintain awareness of emerging and future industry best practices and techniques, open and approved standards, and compliance and certification criteria
Am-Pr: Architecture Processes: Problem Framing Report [N/A]
NOTE: Periodic review and feedback to use and guide software, system and enterprise lifecycle standards begins here.
Capture Architecture Management Standards, Policies, and Platforms
Applicable ISO/IEC/IEEE and OMG Standards
ISO/IEC/IEEE Standards
OMG & ISO/IEC System Modeling Language Standard
Government Acquisition Standards

Figure A:5 - Step 0.1: Assemble Standards and Practices

A.4.2 Problem Framing

Step 0.2 – Conduct Problem Framing – Problem framing is conducted to help identify the appropriate architecture *models* and *views* to build. Intended uses and users of the architecture are captured to determine issues to be explored, questions to be answered, the types of analysis that need to be performed using architecture models and views, and what are the interests and *perspectives* of the intended audience and expected users of the architecture description. This activity will also identify the activities and decisions to be supported using the architecture models and views.

- *Model* – abstract representation of an entity or collection of entities that provides the ability to portray, understand or predict the properties or characteristics of the entity or collection under conditions or situations of interest [ISO 42020]

- *View* – an “information item, governed by an architecture viewpoint, comprising part of an architecture description” [ISO 42010] that communicates some aspect of an architecture and expressing the architecture from the perspective of specific stakeholders regarding specific aspects of the architecture entity and its environment [ISO 42020]
- *Stakeholder Perspective* – way of thinking about an entity, especially as it relates to concerns

The intended scope and context of the architecture is captured to help understand the dependencies, relevant points of view, environmental factors, operational scenarios, major constraints and key assumptions. The depth and breadth of the architecture should be defined. This bounds the purpose and use of the architecture description and helps ensure that architecture descriptions best serve the stakeholder needs. The necessary information and data needs are identified, to include what information will be needed to help populate and generate views and products, the expected precision and granularity of information in the views, and the desired presentation forms and methods. Sources of the information and data must be determined that will be used to populate the models and views.

Next, the types of views and products needed are identified that address the intended uses and users identified above. Sample views and products are captured in conceptual schemas and storyboards to help understand the nature of these items as well as having a way to do early validation with key stakeholders to ensure they will properly express how their concerns are being addressed in the architecture.

- *Architecture Viewpoint* - conventions for the creation, interpretation and use of an architecture view to frame one or more concerns
- *Architecture View* - information item, governed by an architecture viewpoint, comprising part of an architecture description which expresses the architecture of the enterprise or system-of-interest
- *Aspect* – category of model distinguished by its key characteristics and modeling conventions. UAF Aspects include taxonomy, structure, connectivity, processes, states, sequences, information, parameters, motivation, constraints, roadmaps, and traceability.

To support the ability to produce certain views for the conceptual schemas and viewpoints, the architecture description elements should be prototyped to validate and verify metamodel application and model profile usages. Particular organizational and cultural language in the enterprise may differ from standard metamodels and profiles, so a trace is made between enterprise cultural terms and standard architecting terms. This trace is further supported by capturing enterprise terms in Step 0.6 and formalizing modeling profile extensions in Step 0.5.

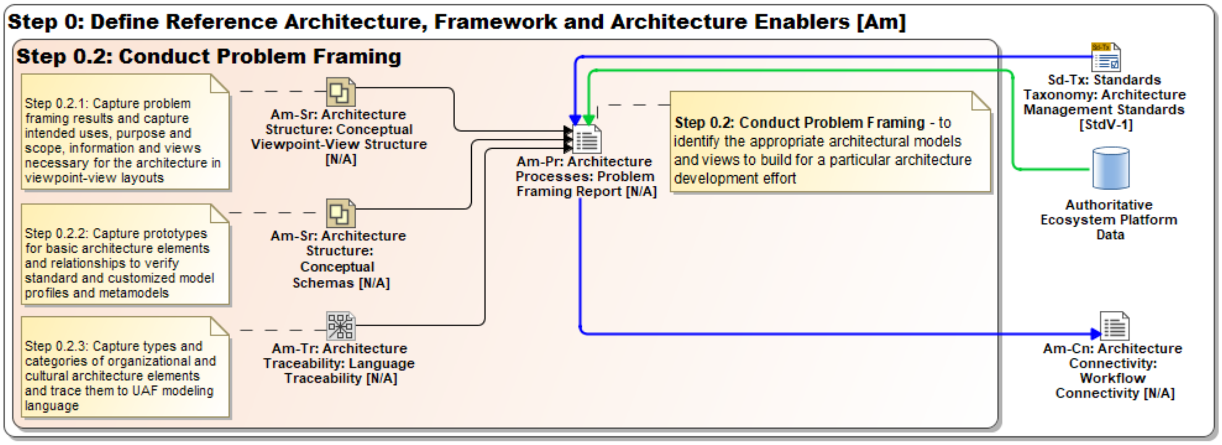

Step 0: Define Reference Architecture, Framework and Architecture Enablers [Am]
Step 0.2: Conduct Problem Framing
Step 0.2.1: Capture problem framing results and capture intended uses, purpose and scope, information and views necessary for the architecture in viewpoint-view layouts
Step 0.2.2: Capture prototypes for basic architecture elements and relationships to verify standard and customized model profiles and metamodels
Step 0.2.3: Capture types and categories of organizational and cultural architecture elements and trace them to UAF modeling language
Am-Sr: Architecture Structure: Conceptual Viewpoint-View Structure [N/A]
Am-Sr: Architecture Structure: Conceptual Schemas [N/A]
Am-Tr: Architecture Traceability: Language Traceability [N/A]
Am-Pr: Architecture Processes: Problem Framing Report [N/A]
Step 0.2: Conduct Problem Framing - to identify the appropriate architectural models and views to build for a particular architecture development effort
Sd-Tx: Standards Taxonomy: Architecture Management Standards [StdV-1]
Authoritative Ecosystem Platform Data
Am-Cn: Architecture Connectivity: Workflow Connectivity [N/A]

Figure A:6 - Step 0.2: Conduct Problem Framing

A.4.3 Architecture Description Organization and Relationships

Step 0.3 – Plan Architecture Description Standup – The enterprise architecture description must be designed and stood up through model preparation and data collection in order to enable building the requisite models, views and products to be used in the construction of the architecture description, which will be produced in Steps 1 through 9.

As the enterprise architecture description is built in those steps, it will be used to analyze, evaluate, compare, and iterate architecture information, which is then used in accordance with architecture governance as described in Step 0.4. This enables the architecture to be used for its intended purposes, including enterprise transformation efforts with regard to making acquisition decisions, changing operational concepts, designing systems, outsourcing services, establishing joint ventures, migrating systems, etc. The workflow for a particular enterprise architecture may be specifically tailored and adapted from this *Enterprise Architecture Guide* which itself captures and describes a generic *Workflow Connectivity* for *Architecture Management*.

A variety of views may be needed to communicate usage of other ***referenced*** architectural descriptions by the architecture description under development. Some ***architectural descriptions*** may represent generalizations or elements intended to convey guidance and patterns for reuse or high-level implementation guidance, and thus serve as foundations of objective architectures which drive commonality, modularity, integration, efficiency, or other factors in specific ***architectural descriptions*** which are intended for specific and real implementation.

- **Architectural Description** – a work product used to express the Architecture of some System Of Interest. It provides executive-level summary information about the architecture description in a consistent form to allow quick reference and comparison between architecture descriptions -- It includes assumptions, constraints, and limitations that may affect high-level decisions relating to an architecture-based work program. (Note: ***architectural description*** is a UAF model element and "*architecture description*" is simultaneously a concept used in ISO 42010 that is defined as a collection of architecture views)

An enterprise ***architectural description*** may, then, reference two kinds of architectural descriptions. One kind is an architecture of a system that exists within the enterprise that is created as a part of the enterprise. If this system is heavily integrated within the enterprise, it needs a description that is embedded within the enterprise to eliminate duplication of elements shared with the enterprise. A second kind is an enterprise-reference architecture, as outlined in ISO/IEC 15704. An enterprise-reference architecture is a guide to help tailor new and emerging development of components or systems to the enterprise of interest. These concepts may include designs for future or targeted operations, resources, services, personnel and security viewpoints for the enterprise, and may come in the form of generic element, partial models, or particular models. The 15704 standard establishes a standardized approach for reuse of explicit enterprise designs and models to achieve enterprise engineering on an ongoing basis to realize further improvements in enterprise operation.

Although a specific ***architectural description*** may ***reference*** another within its description context, there may be a need to federate multiple architectural descriptions in a structured manner or as a set of associations or usages within an organization's architecture description repository or between repositories. A larger *architecture description* may thus be formed, comprised of associations or levels which may support governance controls, levels of detail, interface organization, and even proposal libraries used by an organization to communicate with vendors and contractors. This federation may also include Systems Engineering design models based in languages that are compatible with the overarching architecture framework, so that *architecture description elements* may be directly translated into model profiles using the same underlying modeling language, such as Systems Modeling Language (SysML) or Unified Modeling Language (UML).

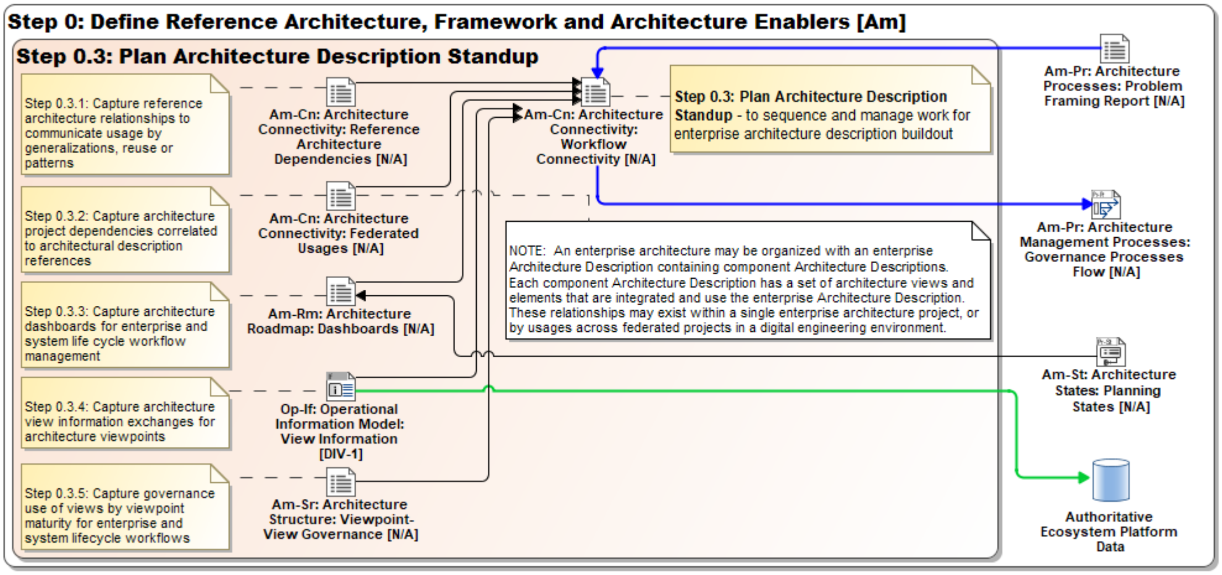
- **Architectural Reference** – a tuple that specifies that one **Architectural Description** refers to another (Note: this relationship may specify that one **Architectural Description** refers to, or depends upon, another one)

Dashboards can be used to help users navigate the model and are created to plan and lay out work or workflows for construction of the *architecture description*. Dashboards may include a special arrangement which specifically supports the *viewpoints* or other schemas that are addressing certain kinds of questions that are commonly asked of the architecture by key *stakeholders*. These dashboards may show view dependencies that correlate to information or *architecture view components* used in one *view* that are re-utilized in another *view* to allow expression of answers or framing of *concerns* which need to be addressed in *viewpoints*. Additionally, *legends* are designed for common use which may further prepare views for particular audiences or *stakeholder perspectives*.

- *View* – an "information item, governed by an architecture viewpoint, comprising part of an architecture description" [ISO 42010] that communicates some aspect of an architecture and expressing the architecture from the perspective of specific stakeholders regarding specific aspects of the architecture entity and its environment [ISO 42020]

- *Viewpoint* – “conventions for the creation, interpretation and use of an architecture view to frame one or more concerns” [ISO 42010] that governs the creation of views
- *Architecture View Component* – a separable portion of one or more architecture views that is governed by the applicable *Aspect* or legend
- *Legend* – offers readers the conventions used in preparing a view, such as its scale, color scheme, and other symbology, thereby aiding readers in interpreting the view as intended

The information used by views for the viewpoints is captured in a special ***logical information model*** view to support standup of the ***views***, ***viewpoints***, and their associated information usage.


Step 0: Define Reference Architecture, Framework and Architecture Enablers [Am]
Step 0.3: Plan Architecture Description Standup
Am-Cn: Architecture Connectivity: Reference Architecture Dependencies [N/A]
Am-Cn: Architecture Connectivity: Federated Usages [N/A]
Am-Rm: Architecture Roadmap: Dashboards [N/A]
Op-If: Operational Information Model: View Information [DIV-1]
Am-Sr: Architecture Structure: Viewpoint-View Governance [N/A]
Am-Cn: Architecture Connectivity: Workflow Connectivity [N/A]
Am-Pr: Architecture Processes: Problem Framing Report [N/A]
Am-Pr: Architecture Management Processes: Governance Processes Flow [N/A]
Am-St: Architecture States: Planning States [N/A]
Authoritative Ecosystem Platform Data

Step 0.3.1: Capture reference architecture relationships to communicate usage by generalizations, reuse or patterns

Step 0.3.2: Capture architecture project dependencies correlated to architectural description references

Step 0.3.3: Capture architecture dashboards for enterprise and system life cycle workflow management

Step 0.3.4: Capture architecture view information exchanges for architecture viewpoints

Step 0.3.5: Capture governance use of views by viewpoint maturity for enterprise and system lifecycle workflows

Step 0.3: Plan Architecture Description Standup - to sequence and manage work for enterprise architecture description buildout

NOTE: An enterprise architecture may be organized with an enterprise Architecture Description containing component Architecture Descriptions. Each component Architecture Description has a set of architecture views and elements that are integrated and use the enterprise Architecture Description. These relationships may exist within a single enterprise architecture project, or by usages across federated projects in a digital engineering environment.

Figure A:7 - Step 0.3: Plan Architecture Description Standup

A.4.4 Architecture Governance

An important consequence of performing workflow steps in parallel with different groups and people is the challenge to keep things in concordance and synchronized across the full set of architecture views. This will usually require establishment of architecture governance procedures and forums, sometimes with something like an architecture governance board to help orchestrate changes and serve as a decision body with authority for making architectural changes. The UAF architecture views can be used to inform the governance process as illustrated in Figure 1:4.

This tiering structure described in section 2.3 has implications for Architecture Governance as discussed in ISO 42020 and illustrated in Figure 1:4. This can be a complex subject which needs to be addressed in an architecture modeling methodology document and in organizational governance processes and procedures.

Step 0.4 – Capture and Monitor Architecture Governance – Establish governance and supporting management processes, guided by ISO 42010 and ISO 15288[10], to be used by an enterprise’s organizations and define how these

[10] ***ISO/IEC/IEEE 15288:2015 – Systems and software engineering – System life cycle processes***, establishes a common framework of process descriptions for describing the life cycle of systems created by humans.

processes will be supported by the use of architectural descriptions and their associated views and viewpoints. To integrate these the governance and management processes are captured in terms of ***personnel functions*** to understand the ***resource information*** inputs necessary for governance and management decisions and the associated information outputs resulting from the decisions. ***Personnel*** involved in architecture processes are associated with each other where ***controls*** exist between the personnel involved in the processes.

- **Control** – a type of **Resource Exchange** that asserts that an organizational resource controls another physical resource (Note: e.g., the chair of an architecture concept decision forum makes a decision that controls actions for a budget planning process)

If necessary, a complete personnel architecture may be created, as outlined in Step 6, to implement architecture processes in an organization or among organizations that participate in an enterprise. However, this Step 0.4 serves only to capture ***resource information*** exchanges which relate to real resource information elements, that are supported by a governance ***logical information model*** which serves to directly tie views to governance and management decision making for enterprise life cycle activities.

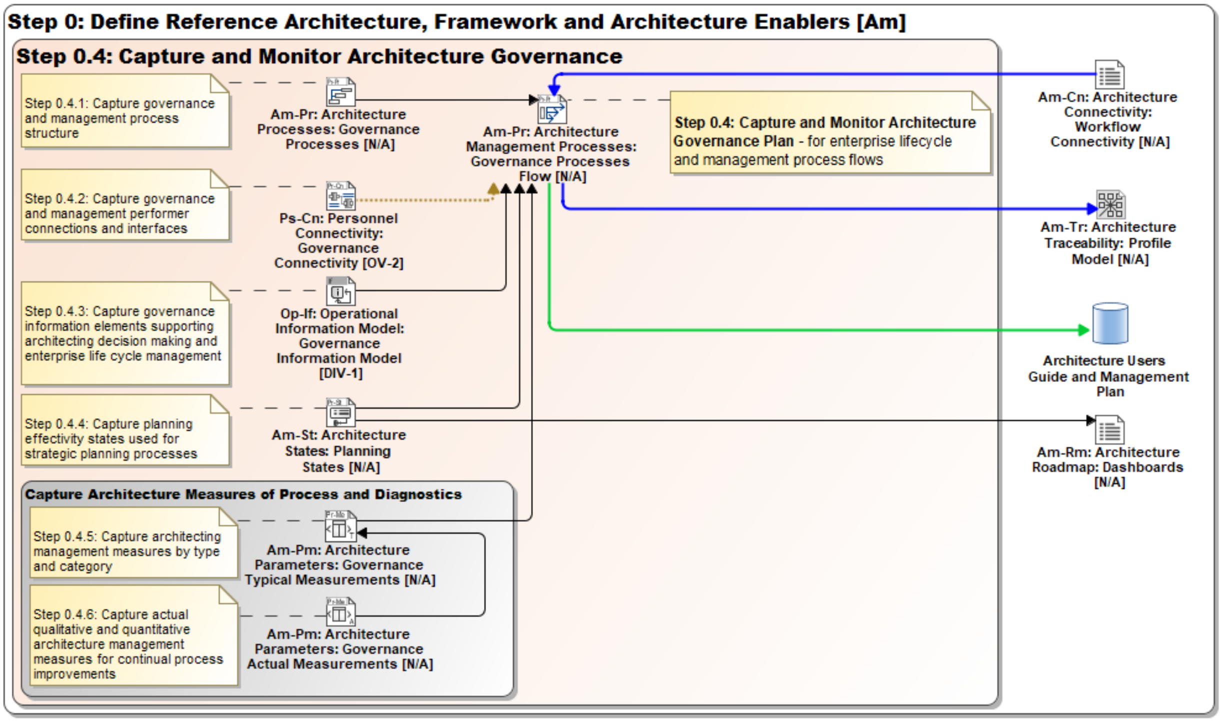

Step 0: Define Reference Architecture, Framework and Architecture Enablers [Am]
Step 0.4: Capture and Monitor Architecture Governance
Step 0.4.1: Capture governance and management process structure
Am-Pr: Architecture Processes: Governance Processes [N/A]
Am-Pr: Architecture Management Processes: Governance Processes Flow [N/A]
Step 0.4: Capture and Monitor Architecture Governance Plan - for enterprise lifecycle and management process flows
Am-Cn: Architecture Connectivity: Workflow Connectivity [N/A]
Step 0.4.2: Capture governance and management performer connections and interfaces
Ps-Cn: Personnel Connectivity: Governance Connectivity [OV-2]
Am-Tr: Architecture Traceability: Profile Model [N/A]
Step 0.4.3: Capture governance information elements supporting architecting decision making and enterprise life cycle management
Op-If: Operational Information Model: Governance Information Model [DIV-1]
Architecture Users Guide and Management Plan
Step 0.4.4: Capture planning effectivity states used for strategic planning processes
Am-St: Architecture States: Planning States [N/A]
Am-Rm: Architecture Roadmap: Dashboards [N/A]
Capture Architecture Measures of Process and Diagnostics
Step 0.4.5: Capture architecting management measures by type and category
Am-Pm: Architecture Parameters: Governance Typical Measurements [N/A]
Step 0.4.6: Capture actual qualitative and quantitative architecture management measures for continual process improvements
Am-Pm: Architecture Parameters: Governance Actual Measurements [N/A]

Figure A:8 - Step 0.4: Capture and Monitor Architecture Governance

Architectural descriptions may simultaneously capture intended objective architectures as well as existing architectures to support a variety of planning states. Planning states could include concepts like “As-Is,” “To-Be,” “Should-Be,” “Could-Be,” and other planning paradigms used by the organizations. Often organizations are driven by budget cycle and financial planning states, which may necessitate the need to track multiple ***architectural descriptions***, or ***versions of configuration*** of elements, that correspond to planning states that are exploratory and not yet committed to by the organization. These planning states are defined to enhance understanding of decision making in architectural personnel processes and associated resource information which is exchanged for controls.

Architecture process ***measures*** are defined for the personnel performers and their ***functions***. Measures of overall personnel functions and organizational resources should demonstrate satisfaction of architecture governance and management objectives. These are used to generate actual measurements to capture analysis of architecture process performance over time, to support evaluation of process improvements and efficiencies. Model-Based Enterprise Architecting (MBEA) practices should show value through reduction in decision analysis workloads, agility in architectural changes, support to innovative architectural guidance for system engineering, and streamlining of architecture-to-production timelines.

When needed, architecture governance and management plans can be developed based on the models and views generated during this Step 0.4. Ideally, these documents are automatically generated from the model itself using reporting scripts and model queries. Additionally, the architectural description itself may be tied directly, or through other digital environment mechanisms, to integration authorities, architecture governance boards, managers, architects, and architecture users to support various councils, forums, working groups, and integrated product teams.

A.4.5 Modeling Profile and Environment

Step 0.5 – Capture Profile and Environment Usage – Schemas for architecture description model integration in a digital environment are captured to convey profile usages, modeling profile extensions, architectural version controls, use of metadata, and data usage by other SE and management tools to support overall use of architecture for its intended purpose within a larger context of Model-Based Enterprise Architecture and systems engineering.

An architecture description is commonly a centerpiece among a variety of architecture information items and information sources which can be integrated. Other architecture information and architecture models that exist outside of an *architecture description* model may include various kinds of depictions of cost, schedules, risks, opportunities, forecasts, benchmarks, physics, operations, etc.

First, a profile model is constructed to communicate the use of the modeling profile in the *architecture description language* that will be used for the *architecture description* model itself.

- *Architecture Description Language* – means of expression, with syntax and semantics, consisting of a set of representations, conventions, and associated rules intended to be used to describe an architecture

Next, a digital environment ***logical information model*** is captured to enable efficient integration or translation of architecture information and *correspondence* rules between different enterprises or groups within an enterprise which use different frameworks, languages or elements in their own internal architecting practices. A digital environment may separately establish understanding of management, pedigree, and authority of the information that is pulled into, out of, or exchanged with an ***architecture description***. A digital environment may also establish the needs for use of specific modeling and analysis tools, types of information technology infrastructure, data and application services, hosting, and publishing applications.

- *Correspondence* – expression of relationship among architecture description elements or among architecture descriptions

Means of correspondence, translation and usage of information, metadata and data formats must be understood in order to efficiently and effectively integrate an architecture description model to other architecture descriptions and to other types of architecture models, such as analytical models, simulation engines, and presentation tools.

Prototyping of the architecture description from Step 0.2 may drive a need to specialize some of the architecture description elements by extending these from an underlying standard modeling profile (such as the UAFML). Extension views are captured comprised of new specialized elements with their stereotype definitions, restrictions on relationships to other elements, custom iconography, use within view specifications, menu and pallet allowances, and other descriptions. Supporting rationale and comments may address guidance on use of these extensions.

Architectural descriptions, or portions of their views, are published through a variety of mechanisms to architecture users for many purposes, such as design reviews, decision forums, outreach, and general communications. Summary and overview diagrams are developed that indicate which specific architectural descriptions will exist, or the views they contain, and what publications they align with. These publications may be temporary or long-standing and align with intended usage and may include commenting and feedback to directly receive comments into an architecture description to record usage or review events, including disposition of comments and change management decisions.

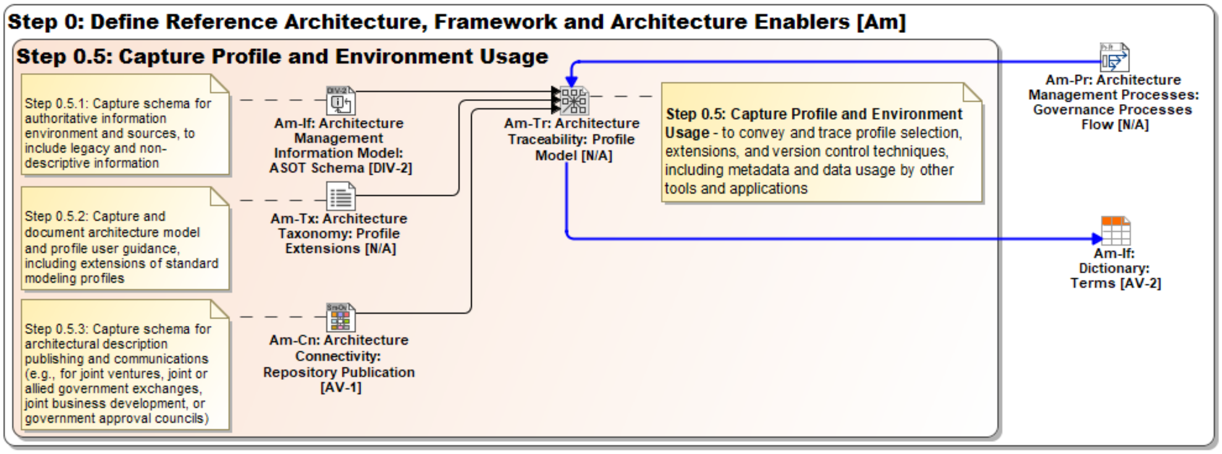

Step 0: Define Reference Architecture, Framework and Architecture Enablers [Am]
Step 0.5: Capture Profile and Environment Usage
Step 0.5.1: Capture schema for authoritative information environment and sources, to include legacy and non-descriptive information
Step 0.5.2: Capture and document architecture model and profile user guidance, including extensions of standard modeling profiles
Step 0.5.3: Capture schema for architectural description publishing and communications (e.g., for joint ventures, joint or allied government exchanges, joint business development, or government approval councils)
Am-If: Architecture Management Information Model: ASOT Schema [DIV-2]
Am-Tx: Architecture Taxonomy: Profile Extensions [N/A]
Am-Cn: Architecture Connectivity: Repository Publication [AV-1]
Am-Tr: Architecture Traceability: Profile Model [N/A]
Step 0.5: Capture Profile and Environment Usage - to convey and trace profile selection, extensions, and version control techniques, including metadata and data usage by other tools and applications
Am-Pr: Architecture Management Processes: Governance Processes Flow [N/A]
Am-If: Dictionary: Terms [AV-2]

Figure A:9 - Step 0.5: Capture Profile and Environment Usage

A.4.6 Terms and Definitions

Step 0.6 – Capture Enterprise Terms and Definitions – Once an architecture management description has been captured and defined, existing terms and elements are defined, glossed, described, or documented to support the full extent of language and local dialect conventions used by the enterprise organizations.

- **Information** – a comment that describes the state of an item of interest in any medium or form, which is communicated or received
- **Information Kind** – an enumerated measure to indicate that an **Information** item is data, pedigree information, position reference information, domain information, or information
- **Alias** – a metamodel artifact used to define an alternative name for an element
- **Definition –** a comment containing a description of an element in the architecture
- **Same As –** a tuple that asserts that two elements refer to the same real-world thing

Dictionaries do not need to define, describe or document all elements in an architecture description since many elements are self-defining in the title or name of the element. Discretion should be used in capturing dictionaries to maximize their usefulness for the architectural context. Common word definitions that exist in a human language should not be re-stated in an architecture description unless idiomatic application requires redefining words or terms to ensure clarity in context for the architecture.

Dictionaries may be organized by kinds of information, or types of the description, such as a formal definition versus a description statement, acronym spelling, or some other scheme.

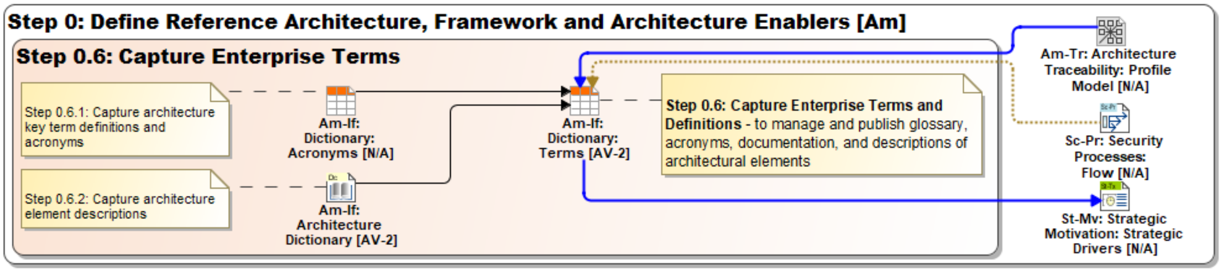

Step 0: Define Reference Architecture, Framework and Architecture Enablers [Am]
Step 0.6: Capture Enterprise Terms
Step 0.6.1: Capture architecture key term definitions and acronyms
Step 0.6.2: Capture architecture element descriptions
Am-If: Dictionary: Acronyms [N/A]
Am-If: Architecture Dictionary [AV-2]
Am-If: Dictionary: Terms [AV-2]
Step 0.6: Capture Enterprise Terms and Definitions - to manage and publish glossary, acronyms, documentation, and descriptions of architectural elements
Am-Tr: Architecture Traceability: Profile Model [N/A]
Sc-Pr: Security Processes: Flow [N/A]
St-Mv: Strategic Motivation: Strategic Drivers [N/A]

Figure A:10 - Step 0.6: Capture Enterprise Terms

A.5 Architecture View Summary for Step 0

The view specifications in UAF for this viewpoint are outlined here:

	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr
Architecture Mgmt Am	Architecture Principles Am-Mv	Architecture Extensions Am-Tx	Architecture Views Am-Sr	Architectural References Am-Cn	Architecture Development Method Am-Pr	-	-	Dictionary Am-If	Architecture Parameters Am-Pm	Architecture Constraints Am-Ct	Architecture Roadmap Am-Rm	Architecture Traceability Am-Tr
Summary & Overview Sm-Ov												

A summary of the lower-level steps and UAF views is shown below. The architecture views produced by each step are identified by the UAF grid designator Aa-Bb where Aa represents the Viewpoint in the grid and Bb represents the Aspect in the grid. The name of the architecture view matches the name of the view specification in the UAFML. In some cases, a subtitle is provided to identify what kind of information is provided in an instance of that UAF view. There will be some architecture views that are not part of the UAF specification since you will sometimes need fit-for-purpose (i.e., custom) to capture the necessary architectural information.

Architecture Management is a new Viewpoint in UAF, so the designator in this section is “Am” to indicate available views defined by UAF. However, there are some additional fit-for-purpose non-UAF views that could be useful in Architecture Management.

1	**Step 0: Define Reference Architecture, Framework and Architecture Enablers [Am]**	**Views**
2	**Step 0.1: Assemble Standards and Practices** – to review and apply best industry practices and techniques, open and approved standards, and compliance and certification criteria	Sd-Tx: Standards Taxonomy: *Architecture Management Standards* [StdV-1]
3	Step 0.1.1: Capture the organization’s use and application of architecture principles and standards	Am-Mv: Architecture Principles [N/A]
4	Step 0.1.2: Capture feedback to standards development organizations	Am-Ct: Architecture Constraints: *Standards Feedback* [N/A]
5	**Step 0.2: Conduct Problem Framing** – to identify the appropriate architectural models and views to build for a particular architecture development effort	Am-Pr: Architecture Processes: *Problem Framing Report* [N/A]
6	Step 0.2.1: Capture problem framing results and capture intended uses, purpose and scope, information and views necessary for the architecture description in viewpoint-view layouts	Am-Sr: Architecture Structure: *Conceptual Viewpoint-View Structure* [N/A]
7	Step 0.2.2: Capture prototypes for basic architecture elements and relationships to verify standard and customized model profiles and metamodels	Am-Sr: Architecture Structure: *Conceptual Schemas* [N/A]
8	Step 0.2.3: Capture types and categories of organizational and cultural architecture elements and trace them to UAF modeling language	Am-Tr: Architecture Traceability: *Language Traceability* [N/A]
9	**Step 0.3: Plan Architecture Description Standup** – to sequence and manage work for enterprise architecture description buildout	Am-Cn: Architecture Connectivity: *Workflow Connectivity* [N/A]
10	Step 0.3.1: Capture reference architecture relationships to communicate usage by generalizations, reuse or patterns	Am-Cn: Architecture Connectivity: *Reference Architecture Dependencies* [N/A]
11	Step 0.3.2: Capture architecture project dependencies correlated to architectural description references	Am-Cn: Architecture Connectivity: *Federated Usages* [N/A]
12	Step 0.3.3: Capture architecture dashboards for enterprise and system life cycle workflow management	Am-Rm: Architecture Roadmap: *Dashboards* [N/A]
13	Step 0.3.4: Capture architecture view information exchanges for architecture viewpoints	Op-If: Operational Information Model: *View Information* [DIV-1]
14	Step 0.3.5: Capture governance use of views by viewpoint maturity for enterprise and system lifecycle workflows	Am-Sr: Architecture Structure: *Viewpoint-View Governance* [N/A]

15	**Step 0.4: Capture and Monitor Architecture Governance** – for enterprise lifecycle and management process flows	Am-Pr: Architecture Development Method: *Governance Processes Flow* [N/A]
16	Step 0.4.1: Capture governance and management process structure	Am-Pr: Architecture Processes: *Governance Processes* [N/A]
17	Step 0.4.2: Capture governance and management performer connections and interfaces	Ps-Cn: Personnel Connectivity: *Governance Connectivity* [OV-2]
18	Step 0.4.3: Capture governance information elements supporting architecting decision making and enterprise life cycle management	Op-If: Operational Information Model: *Governance Information Model* [DIV-1]
19	Step 0.4.4: Capture planning effectivity states used for strategic planning processes	Am-St: Architecture States: *Planning States* [N/A]
20	Step 0.4.5: Capture architecting management measures by type and category	Am-Pm: Architecture Parameters: *Governance Typical Measurements* [N/A]
21	Step 0.4.6: Capture actual qualitative and quantitative architecture management measures for continual process improvements	Am-Pm: Architecture Parameters: *Governance Actual Measurements* [N/A]
22	**Step 0.5: Capture Profile and Environment Usage** – to convey and trace profile selection, extensions, and version control techniques, including metadata and data usage by other tools and applications	Am-Tr: Architecture Traceability: *Profile Model* [N/A]
23	Step 0.5.1: Capture schema for authoritative information environment and sources, to include legacy and non-descriptive information	Am-If: Architecture Management Information Model: *ASOT Schema* [DIV-2]
24	Step 0.5.2: Capture and document architecture model and profile user guidance, including extensions of standard modeling profiles	Am-Tx: Architecture Taxonomy: *Profile Extensions* [N/A]
25	Step 0.5.3: Capture schema for architectural description publishing and communications (e.g., for joint ventures, joint or allied government exchanges, joint business development, or government approval councils)	Am-Cn: Architecture Connectivity: *Repository Publication* [AV-1]
26	**Step 0.6: Capture Enterprise Terms and Definitions** – to manage and publish glossary, definitions, acronyms, documentation, and descriptions of architectural elements	Am-If: Dictionary: *Terms* [AV-2]
28	Step 0.6.1: Capture architecture key term definitions and acronyms	Am-If: Dictionary: *Acronyms* [N/A]
28	Step 0.6.2: Capture architecture element descriptions	Am-If: Architecture Dictionary [AV-2]

B Appendix B – Glossary

These *key concepts* are highlighted in ***bold italics*** within the Narrative and some of the less obvious concepts are listed with the associated ISO Standard *meaning* or the UAF ***meaning*** of that concept. The meaning of the element types in the UAF is based upon concepts defined in the International Defence Enterprise Architecture Specification (IDEAS)[11], which is a formal higher-order four-dimensional (4D) ontology. The first four items (type, tuple, edge, individual) are based on the IDEAS Framework illustrated in Figure B:1.

[11] **IDEAS** is a formal, higher order, 4D ontology. It is extensional, using physical existence as its criterion for identity. In practical terms, this means the ontology is well suited to managing change over time and identifying elements with a degree of precision that is not possible using names alone. [https://en.wikipedia.org/wiki/IDEAS_Group]

- **Type** – denotes a set of individuals
- **Tuple** – denotes a relationship that exists between elements
- *Edge* – denotes an edge relationship (this is called "couple" in IDEAS ontology)
- **Individual** – denotes a single instance of an element
- *Abstract* – denotes that the element has no direct use but is a means of construction
- **Enumeration** – is a complete ordered listing of all the items in a collection
- Property – a defined characteristic of an element
- *ISO* – is a term that exists in the ISO ISO-42010[12] or ISO-42020[13] standards for architecture

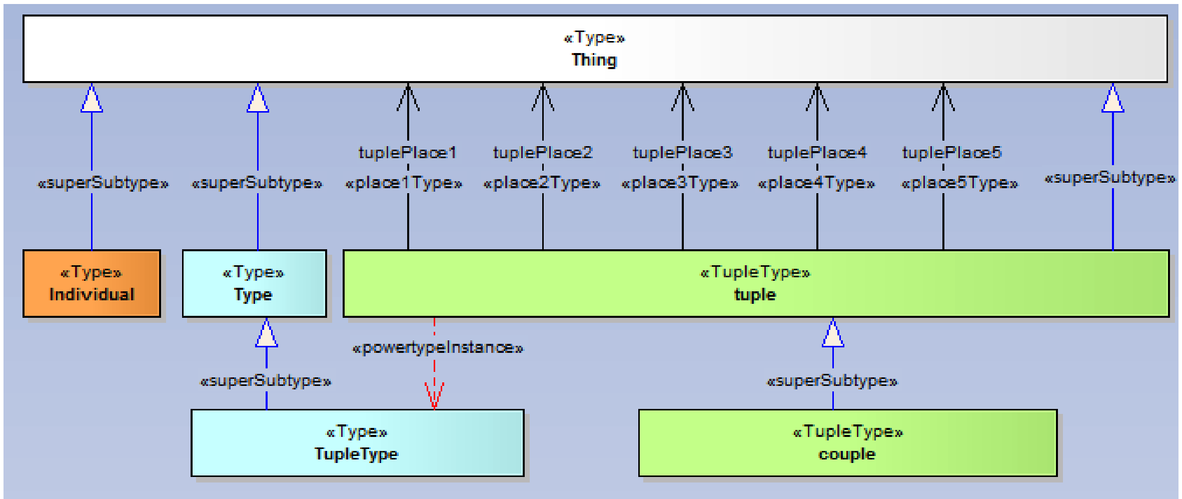


Figure B:1 - The IDEAS Upper Ontology Used as the Basis for the UAF Domain Metamodel

The terms below are used in UAF and throughout this document. The color coding of these terms is as noted above. The definitions are either verbatim from the UAF specification or a suitable paraphrase based on the context of its usage herein.

Achieves	a tuple that exists between an *Actual State* (e.g., observed/measured during testing) of an element that attempts to achieve a desired effect and an *Achiever*
Achiever	An **Actual Resource**, **Actual Project** or **Actual Enterprise Phase** that can deliver a desired effect
Activity Performable Under Condition	The **Actual Condition** under which an Activity is performed
Actual	an enumerated type of an **Actual Measurement Kind** which is based on an actual value
Actual Effect	a real-world phenomenon that follows and is caused by some previous phenomenon
Actual Enduring Task	an actual undertaking recognized by an enterprise as being essential to achieving its goals, i.e., a strategic specification of what the enterprise does

[12] ***ISO/IEC/IEEE 42010:2021 Software, systems and enterprise – Architecture description*** establishes a standard approach to describing an architecture using views and viewpoints, architecture description languages and architecture descriptions frameworks.

[13] ***ISO/IEC/ IEEE 42020:2019 – Software, systems and enterprise – Architecture processes*** establishes a set of process descriptions for the governance and management of a collection of architectures and the architecting of the entities

Actual Enterprise Phase	an individual that describes the phase of an actual enterprise endeavor
Actual Measurement	an actual value that is applied to a **Measurement**
Actual Measurement Kind	an enumerated type of an **Actual Measurement** which is based on a required value
Actual Organization	an actual formal or informal organizational unit, e.g., "Driving and Vehicle Licensing Agency", "UAF team Alpha"
Actual Organizational Resource	an instance of an *Organizational Resource* in the real world
Actual Outcome	an individual that describes something that happens or is produced as the final consequence or product and is related to one of the aims goals for the business or enterprise. Outcome is a special kind of effect, one that is usually at the end of a chain of effects, i.e., an "end effect".
Actual Person	an individual human being
Actual Post	an actual, specific post, an instance of a **Post** "type" - e.g., "President of the United States of America" where the **Post** would be president
Actual Project	a time-limited endeavor to provide a specific set of **Actual Resources** that meet specific **Capability** needs
Actual Project Milestone	an event with a start date in an **Actual Project** from which progress is measured
Actual Resource	an individual, fully-realized *Resource Performer*
Actual Resource Relationship	an abstract element that details the *Actual Organizational Resources* that are able to carry out an **Actual Responsibility**
Actual Responsibility	an actual duty required of a **Person** or **Organization**
Actual Service	an individual **Service**
Actual Strategic Phase	a phase of an **Actual Enterprise Mission**, **Value Stream** or **Actual Enduring Task** endeavor
Affectable Element	an abstract grouping of elements that can be affected by a **Risk** (Note: these include **Enterprise Goals**, *Processes*, *Assets*, **Opportunities**, and **Capabilities**)
Alias	a metamodel artifact used to define an alternative name for an element
Architecture	fundamental concepts or properties related to an entity in its environment and governing principles for the realization and evolution of this entity and its related life cycle processes
Architecture Description	work product used to express an architecture
Architectural Description	a work product used to express the Architecture of some System Of Interest. It provides executive-level summary information about the architecture description in a consistent form to allow quick reference and comparison between architecture descriptions -- It includes assumptions, constraints, and limitations that may affect high-level decisions relating to an architecture-based work program.
Architecture Description Framework	conventions, principles and practices for the description of architectures established within a specific domain of application or community of stakeholders
Architecture Description Language	means of expression, with syntax and semantics, consisting of a set of representations, conventions, and associated rules intended to be used to describe an architecture
Architecture Metadata	information associated with an **Architectural Description**, that supplements the standard set of tags used to summarize the Architecture. It states things like what methodology was used, notation, etc.
Architectural Reference	a tuple that specifies that one **Architectural Description** refers to another
Architecture View	information item, governed by an architecture viewpoint, comprising part of an architecture description which expresses the architecture of the enterprise or system-of-interest
Architecture View Component	a separable portion of one or more architecture views that is governed by the applicable *Aspect* or legend

Architecture Viewpoint	conventions for the creation, interpretation and use of an architecture view to frame one or more concerns
Asset	an abstract element that indicates the types of elements that can be affected by **Risk**. *Asset* as applied to Security views is an abstract element that indicates the types of elements that can be considered as a subject for security analysis.
Asset Role	an abstract element that indicates the types of elements that can be affected by **Risk** in the particular context. *Asset Role* as applied to Security views, is an abstract element that indicates the type of elements that can be considered as a subject for security analysis in the particular context.
Capable Element	an abstract type that represents a structural element that can **Exhibit** **Capabilities**
Capability	an enterprise's ability to achieve a desired effect realized through a combination of ways and means (e.g., **Capability Configurations**) along with specified **Measures**
Capability Configuration	a composite structure representing the physical and human resources (and their interactions) in an enterprise, assembled to meet a **Capability**
Capability Generalization	a taxonomic relationship between a more general **Capability** and a more specific **Capability**
Capability Kind	an enumerated type of a **Capability** which may include Strategic, Operational, Service, Resource, Personnel, Security, or Other
Challenge	an existing or potential difficulty, circumstance, or obstacle which will require effort and determination from an enterprise to overcome in achieving its goals
Challenge Kind	an enumerated type of a **Challenge** which may include Strategic, Enterprise, Mission, Business, or Other
Concept Item	abstract, an item which may feature in a **High-Level Operational Concept**
Concern	a “matter of relevance or importance to a stakeholder regarding an entity of interest” [ISO 42010] that will be addressed in an architecture
Concern	interest in a **Strategic Phase** (Strategic Phase is synonym for System in ISO 42010) relevant to one or more of its stakeholders
Condition	a type that defines the **Location**, **Environment**, and/or **Geopolitical Extent**
Conforms to	a dependency relationship that relates an element to a **Standard** that the element is conforming to
Command	a type of **Resource Exchange** that asserts that one *Organizational Resource* commands another
Compares To	a tuple used to relate the effect that is achieved with the originally expected Desired Effect. Providing a means of comparison, between the expectation of the *Desirer* and the actual result
Control	a type of **Resource Exchange** that asserts that an *Organizational Resource* **Controls** another *Physical Resource*
Competence	a set of abilities defined by knowledge, skills and aptitude
Correspondence	expression of relationship among architecture description elements or among architecture descriptions
Creates	a tuple used to denote that an **Actual Strategic Phase** brings into existence a *Strategic Asset*
Definition	a comment containing a description of an element in the architecture
Desirer	abstract type used to group architecture elements that might desire a particular **Effect**
Desires	a tuple relating the *Desirer* (i.e., a **Capability** or *Organizational Resource*) to an **Actual State**
Driver	a factor which will have a significant impact on the activities and goals of an enterprise
Driver Kind	an enumerated kind of **Driver** as Strategic, Operational, Service, Resource, Personnel, Security, Project, Standard, Other, or Architecture Principle
Effect	a phenomenon that follows and is caused by some previous phenomenon that could lead to downstream **Effects** or to one or more desired outcomes
Enables	a dependency relationship denoting that an **Opportunity** provides the means for achieving an **Enterprise Goal** or objective

Enhanced Security Control	a statement of security capability to: (i) build in additional but related, functionality to a basic control; and/or (ii)increase the strength of a basic control
Enterprise	human undertaking or venture that has a mission, goals, and objectives to offer products or services or to achieve a desired project or business outcome
Enterprise Goal	a statement about a state or condition of the enterprise to be brought about or sustained through appropriate means. An **Enterprise Goal** amplifies an **Enterprise Vision,** that is it indicates what must be satisfied on a continuing basis to effectively attain the **Enterprise Vision**. [BMM: OMG dtc-13-08-24.]
Enterprise Mission	captures at a high level what you will do to realize your vision
Enterprise Objective	a statement of an attainable, time-targeted, and measurable target that the enterprise seeks to meet in order to achieve its goals. [BMM: 1.3]
Enterprise Vision	describes the future state of the enterprise without regard to how it is to be achieved [BMM: OMB dtc-13-08-24]
Environment	A definition of the environmental factors in which something exists or functions. The definition of an **Environment** element can be further defined using **Environment Kind**.
Environment Kind	An enumerated type of an **Environment** as Terrain Type, Weather Conditions, Light Conditions, CBRN Environment, or Situation Type
Environment Property	A property of an **Environment** that is typed by a **Condition**. The kinds of **Condition** that can be represented are **Location**, **Geopolitical Extent Type** and **Environment**.
Estimate	an enumerated type of an **Actual Measurement Kind** which is based on an estimated value
Exchange *(abstract)*	abstract tuple, grouping **Operational Exchanges** and **Resource Exchanges** that exchange *Resources*
Exchange Item	an abstract grouping for elements that defines the types of elements that can be exchanged between *Assets* and conveyed by an **Exchange**
Exhibits	a tuple that exists between a *Capable Element* and a **Capability** that it meets under specific environmental **Conditions**
Fielded Capability	an individual, fully-realized **Capability**
Fills Post	a tuple that asserts that an **Actual Person** fills an **Actual Post**
Forecast	a dependency relationship that specifies a transition from one *Resource Performer*, **Standard**, or **Competence** to another future one, related to an **Actual Enterprise Phase** to give it a temporal context
Governed By	a tuple that exists between the **Service Contract** and the **Service** that it governs
High-Level Operational Concept	describes the *Resources* and **Locations** required to meet an operational scenario from an integrated systems point of view. It is used to communicate overall quantitative and qualitative system characteristics to stakeholders.
Impacted By	a tuple used to denote that a **Capability** is affected by an **Opportunity**
Implements	a tuple that defines how an element in the upper layer of abstraction is implemented by a semantically equivalent element (for example tracing the **Functions** to the **Operational Activities**) in the lower level of abstraction
Information	a comment that describes the state of an item of interest in any medium or form, which is communicated or received
Information Model	a structural specification of data types, showing relationships between them. The type of information captured in the **Information Model** is described using the enumeration **Information Model Kind** (Conceptual, Logical, and Physical)
Information Model Kind	an enumerated kind of **Information Model** as Conceptual, Logical, or Physical
Information Kind	an enumerated measure to indicate that an **Information** item is data, pedigree information, position reference information, domain information, or information.
Interaction Message	an abstract type that groups several types of messages used in the *Interaction Scenario*
Interaction Role	an abstract type that represents an individual participant in the *Interaction Scenario*
Interaction Scenario	an abstract type that specifies interactions between *Assets*, like *Resource Performers*, and **Services**

Interaction Scenario Generalization	a taxonomic relationship between a more general *Interaction Scenario* and a more specific *Interaction Scenario*
Is Capable To Perform	a tuple defining the traceability between the structural elements to the Activities that they can perform
ISO 8601 Date Time	a date and time specified in the **ISO8601 date-time** format including time zone designator (TZD): YYYY-MM-DDThh:mm:ssTZD
Known Resource	asserts that a known *Resource Performer* constrains the implementation of the **Operational Performer** that plays the role in the **Operational Architecture**
Legend	offers readers the conventions used in preparing a view, such as its scale, color scheme, and other symbology, thereby aiding readers in interpreting the view as intended
Maps to Capability	a tuple denoting that an activity contributes to providing a **Capability**
Measurement	a property of an element representing something in the physical world, expressed in amounts of a unit of measure
Metadata	A comment that can be applied to any element in the architecture. The attributes associated with this element details the relationship between the element and its related dublinCoreElement, metaDataScheme, category and name. This allows the element to be referenced using the Semantic Web.
Milestone Dependency	a tuple between two **Actual Project Milestones** that denotes one **Actual Project Milestone** follows from another
Mitigates	a tuple relating a **Security Control** to a **Risk**. Mitigation is established to manage **Risk** and could be represented as an overall strategy or through techniques (mitigation configurations) and procedures (**Security Processes**).
Model	abstract representation of an entity or collection of entities that provides the ability to portray, understand or predict the properties or characteristics of the entity or collection under conditions or situations of interest [ISO 42020]
Aspect	category of model distinguished by its key characteristics and modeling conventions. UAF Aspects include taxonomy, structure, connectivity, processes, states, sequences, information, parameters, motivation, constraints, roadmaps, and traceability.
Motivated By	a tuple denoting the reason or reasons one has for acting or behaving in a particular way
Motivational Element	an abstract kind of element in the model that provides the reason or reasons one has for acting or behaving in a particular way
Natural Resource	a type of *Physical Resource* that occurs in nature such as oil, water, gas or coal
Operational Activity	an activity that captures a logical process, specified independently of how the *Process* is carried out
Operational Agent	an abstract type grouping **Operational Architecture** and **Operational Performer**
Operational Architecture	a type used to denote a model of the Architecture, described from the Operational perspective
Operational Constraint	a *Rule* governing an operational architecture element i.e., **Operational Performer**, **Operational Activity**, **Operational Information**, etc.
Operational Connector	a Connector that goes between **Operational Roles** representing a need to exchange Resources. It can carry a number of **Operational Exchanges**
Operational Exchange	Asserts that a flow can exist between **Operational Performers** (i.e., flows of information, people, material, or energy)
Operational Exchange Item	an abstract grouping for elements that defines the types of elements that can be exchanged between **Operational Performers** and conveyed by an **Operational Exchange**
Operational Information	an item of information that flows between **Operational Performers** and is produced and consumed by the **Operational Activities** that the **Operational Performers** capable to perform (see **Is Capable to Perform**)
Operational Interaction Scenario	a specification of the interactions between **Operational Performers** in an **Operational Architecture**
Operational Interface	a declaration that specifies a contract between the **Operational Performer** it is related to, and any other **Operational Performers** it can interact with

Operational Message	a message for use in an **Operational Interaction Scenario** which carries any of the subtypes of **Operational Exchange**
Operational Method	a behavioral feature of an *Operational Agent* whose behavior is specified in an **Operational Activity**
Operational Mitigation	a set of **Operational Performers** intended to address against specific operational **Risks**
Operational Performer	a logical entity that **Is Capable To Perform** **Operational Activities** which produce, consume and process *Resources*
Operational Role	Usage of an **Operational Performer** or **Operational Architecture** in the context of another **Operational Performer** or **Operational Architecture**. Creates a whole-part relationship.
Operational State Description	a state machine describing the behavior of an **Operational Performer**, depicting how the **Operational Performer** responds to various events and the actions
Opportunity	an existing or potential favorable circumstance or combination of circumstances which can be advantageous for addressing enterprise **Challenges**
Organization	a group of *Organizational Resources* (**Persons**, **Posts**, **Organizations** and **Responsibilities**) that are associated for a purpose
Organizational Resource	an abstract type for **Organization**, **Person**, **Post** and **Responsibility**
Owns Risk	a tuple relating a **Risk** to an *Organizational Resource* that is responsible for executing the **Risk** mitigation
Owns Risk in Context	a tuple relating a **Risk** to an organizational role that is responsible for executing the **Risk** mitigation in the specific context or configuration
Owns Value	A tuple denoting that an *Actual Organizational Resource* owns a **Value Item**
Performs In Context	A tuple that relates an **Operational Activity Action** to an **Operational Role**, or a **Function Action** to a **Resource Role**. It indicates that the action can be carried out by the role when used in a specific context or configuration.
Person	a type of human being used to define the characteristics that need to be described for **Actual Persons** (e.g., properties such as address, telephone number, nationality, etc.)
Phaseable Element	an abstract element that indicates the types of elements that can be assigned to a specific **Actual Strategic Phase**
Portfolio	an enumerated option of **Project Kind** of an **Actual Project**
Post	a type of job title or position that a **Person** can fill (e.g., Lawyer, Solution Architect, Machine Operator or Chief Executive Officer)
Presented By	tuple denoting that a **Challenge** must be overcome for addressing a **Driver**
Process	an abstract type that represents a behavior or *Process* (i.e., a **Function** or **Operational Activity**) that can be performed by a *Performer*.
Process Edge	an abstract type that represents a behavior or *Process* (i.e., a **Function** or **Operational Activity**) that can be performed by a *Performer*
Process Generalization	a taxonomic relationship between a more general *Process* and a more specific *Process*
Process Operation	an abstract type that represents a behavior or *Process* (i.e., a **Function** or **Operational Activity**) that can be performed by a *Performer*
Process Parameter	an abstract type that represents a behavior or *Process* (i.e., a **Function** or **Operational Activity**) that can be performed by a *Performer*
Process Usage	an abstract type that represents a behavior or *Process* (i.e., a **Function** or **Operational Activity**) that can be performed by a *Performer*
Programme	an enumerated option of **Project Kind** of an **Actual Project**
Project	a type that describes types of time-limited endeavors that are required to meet one or more **Capability** needs
Project	an enumerated option of **Project Kind** of an **Actual Project**
Project Activity	an activity carried out during a **Project**
Project Sequence	a tuple between two **Actual Projects** that denotes one **Actual Project** cannot start before the previous **Actual Project** is finished

Project Theme	a property of a **Project Milestone** that captures an aspect by which the progress of **Actual Projects** may be measured (e.g., doctrine, organization, training, materiel, logistics, personnel, facilities)
Project Kind	a possible enumeration kind of a project to include a **Project**, **Programme**, or **Portfolio**
Project Milestone	a type of event in a **Project** by which progress is measured
Project Milestone Role	the role played by a **Project Milestone** in the context of a **Project**
Project Status	the status (i.e., level of progress) of a **Project Theme** for an **Actual Project** at the time of the **Actual Project Milestone**
Project Theme	a property of a **Project Milestone** that captures an aspect by which the progress of **Actual Projects** may be measured
Property Set Generalization	a taxonomic relationship between a more general *Property Set* and a more specific *Property Set*
Protocol	a **Standard** for communication over a network, which may be composite, represented as a **Protocol Stack** made up of **Protocol Layers**
Protocol Implementation	an abstract type grouping architectural elements that can implement **Protocols**
Protocol Layer	usage of a **Protocol** in the context of another **Protocol** creating a whole-part relationship
Protocol Stack	a sub-type of **Protocol** that contains the **Protocol Layers**, defining a complete stack
Provided Service Level	a sub type of **Actual Service** that details a specific service level delivered by the provider
Refine	a relationship from an architectural element which refines a text-based requirement
Required Service Level	A sub type of **Actual Service** that details a specific **Service** level required of the provider.
Required	an enumerated type of an **Actual Measurement Kind** which is based on a required value
Requirement	a statement that identifies a system, product or process characteristic or constraint, which is unambiguous, clear, unique, consistent, stand-alone (not grouped), and verifiable, and is deemed necessary for stakeholder acceptability (INCOSE 2010)
Resource	abstract type grouping all elements that can be conveyed by an **Exchange**
Resource Architecture	a type used to denote a model of the Architecture, described from the *Resource Performer* perspective
Resource Artifact	a type of man-made object that contains no human beings (e.g., satellite, radio, petrol, gasoline, etc.)
Resource Connector	a channel for exchange between two **Resource Roles**
Resource Constraint	a *Rule* governing the structural or functional aspects of an implementation
Resource Exchange	asserts that a flow can exist between *Resource Performers* (i.e., flows of data, people, material, or energy)
Resource Exchange item	an abstract grouping for elements that defines the types of elements that can be exchanged between *Resource Performers* and conveyed by a **Resource Exchange**
Resource Exchange Kind	an enumeration of a kind of **Resource Exchange** to be Resource Communication, Resource Movement, Resource Energy Flow, or Geopolitical Extent Exchange
Resource Interface	a declaration that specifies a contract between the *Resource Performers* it is related to and any other *Resource Performers* it can interact with. It is also intended to be an implementation of a specification of an Interface in the Business and/or Service layer.
Resource Message	a message for use in a Resource Event-Trace which carries any of the subtypes of **Resource Exchange**
Resource Method	a behavioral feature of a *Resource Performer* whose behavior is specified in a *Resource* **Function**
Resource Mitigation	a set of *Resource Performers* intended to address against specific **Risk**
Resource Performer	an abstract grouping of elements that can perform **Functions**

Resource Readiness Kind	a particular enumeration of the type of readiness for a *Resource* providing a **Capability**: Deployed, In Service, Out of Service, No Longer Used, or Other
Resource Role	usage of a *Resource Performer* in the context of another *Resource Performer* creating a whole-part relationship
Resource Service	a service that a *Resource Performer* provides to support higher level **Service** or **Operational Activity**
Resource State Description	a state machine describing the behavior of a *Resource Performer*, depicting how the *Resource Performer* responds to various events and the actions.
Responsibility	a type of duty required of a **Post, Person** or **Organization**
Responsible For	a tuple between an **Actual Responsible Resource** and an **Actual Responsibility** or **Actual Project**. It defines the duties that the *Actual Responsible Resource* is **Responsible For**.
Responsible Role Kind	an enumerated type that specifies a Manager or Responsible Owner designation for an **Actual Organization** or **Actual Post**
Role Kind	an enumerated type that specifies a **Resource Role** designation such as Part, Component, Used Configuration, Human Resource, Platform, System, Sub Organization, Post Role, Responsibility Role, Equipment, Sub System Part, Hosted Software, Artifact Component, Natural Resource Component, or Other
Risk	a type that represents a situation involving exposure to danger of *Affectable Elements* (e.g., Assets, Processes, **Capabilities**, **Opportunities**, or **Enterprise Goals**) where the effects of such exposure can be characterized in terms of the likelihood of occurrence of a given threat and the potential adverse consequences of that threat's occurrence.
Rule	an abstract type for all types of constraint (i.e., an **Operational Constraint** could detail the rules of accountancy best practice)
Rule Kind	an enumerated type that specifies that a *Rule* may be a structural assertion, action assertion, derivation, contract, constraint, guidance, security policy, or caveat
Same As	a tuple that asserts that two elements refer to the same real-world thing
Security Availability	details the potential impact on organizations or individuals if the information is not available to those who need to access it
Security Classification	details a classification for the exchange
Security Classification Kind	a type that defines acceptable values for the security category (SC) of an information system, where the acceptable values for potential impact are low, moderate, or high
Security Constraint	a type of *Rule* that captures a formal statement to define access control policy language
Security Control	the management, operational, and technical control (i.e., safeguard or countermeasure) to **Protect** the confidentiality, integrity, and availability of the system and its information [NIST SP 800-53]
Security Control Family	an element that organizes **Security Controls** into a family. Each **Security Control Family** contains **Security Controls** related to the general security topic of the family.
Security Enclave	collection of information systems connected by one or more internal networks under the control of a single authority and security policy. The systems may be structured by physical proximity or by function, independent of location.
Service Exchange	asserts that a flow can exist between **Services** (i.e., flows of information, people, materiel, or energy)
Service Exchange Item	An abstract grouping for elements that defines the types of elements that can be exchanged between **Services** and conveyed by a **Service Exchange**
Service Exchange Kind	an enumerated type that specifies that a **Service Exchange** may be a Material Exchange, Organizational Exchange, Energy Exchange, Information Exchange, or Configuration Exchange
Service Message	a message for use in a **Service** Event-Trace
Service Method	a behavioral feature of a **Service** whose behavior is specified in a **Service Function**
Security Integrity	details the potential impact on organization or individuals due to modification or destruction of information, and includes ensuring information non-repudiation and authenticity

Security Risk	the level of impact on enterprise operations, assets, or individuals resulting from the operation of an information system given the potential impact of a threat and the likelihood of that threat occurring [NIST SP 800-65]
Service	the specification of a set of functionalities provided by one element for the use of others
Service Architecture	an element used to denote a model of the Architecture, described from the **Services** perspective
Service Contract	a constraint governing the use of one or more **Services**
Service Exchange	a flow of **Operational Information**, **Service Signals**, people, material or energy
Service Exchange Item	an **Operational Information** element, **Service Signal**, or *Resource* that is conveyed on **Service Exchanges** between **Services**
Service Interface	a contract that defines the **Service Methods** and **Service Signals** that the **Service** realizes
Service Message	a sequenced message between two services which may convey **Service Exchanges** or **Service Methods**
Service Method	a behavioral feature of a **Service** whose behavior is specified in a **Service Function**
Service Policy	a constraint governing the use of one or more **Services**
Service Role	a behavioral feature of a **Service** whose behavior is specified in a **Service Function**
Service State Description	a state machine describing the behavior of a **Service**, depicting how the **Service** responds to various events and the actions
Stakeholder	an individual, team, organization, or classes thereof, having an interest in a **Strategic Phase** [ISO/IEC/IEEE 42010:2011]
Stakeholder Perspective	way of thinking about an entity, especially as it relates to concerns
Standard	a ratified and peer-reviewed specification that is used to guide or constrain the architecture. A **Standard** may be applied to any element in the architecture.
Standard Operational Activity	a sub-type of **Operational Activity** that is a standard operating procedure
State Description	an abstract type that represents a state machine (i.e., an **Operational State Description** or **Resource State Description**), depicting how the *Asset* responds to various events and the actions
State Description Generalization	a taxonomic relationship between a more general *State Description* and a more specific *State Description*
Status Indicators	an enumerated type that specifies a status for a **Project Theme**
Strategic Asset	an abstract element that indicates the types of strategic elements that can be affected by **Risk**
Strategic Exchange	asserts that a flow can exist between **Actual Strategic Phases** (i.e., flows of information, people, materiel, or energy)
Strategic Exchange Item	an abstract grouping for elements that defines the types of elements that can be exchanged between **Actual Strategic Phases** and conveyed by a Strategic Exchange
Strategic Information	knowledge communicated or received concerning a particular fact or circumstance that is strategic in nature that is important or essential in relation to a plan of action
Strategic Phase	a type of a current or future state of the enterprise, Mission, **Value Stream** or Enduring Task.
System	an integrated set of elements, subsystems, or assemblies that accomplish a defined objective, including products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements (INCOSE SE Handbook V4, 2015)
Value Item	an ideal, custom, or institution that an enterprise promotes or agrees with. It may be positive or negative, depending on point of view.
Value Item Kind	An enumerated type that specifies a kind of a **Value Item** as Time, Cost, Quality, Revenue, Benefit, KPI, Loss, or Other)
Value Stream	an end-to-end collection of activities that create a result for a customer, who may be the ultimate customer or an internal end-user of the **Value Stream**. **Value Stream** nested within another value stream may represent Value Stream Stage - a distinct,

	identifiable phase or step within a **Value Stream** [The Business Architecture Metamodel Guide, 2020]
Version of Configuration	a property of a **Whole Life Configuration**, used in version control of a Versioned Element. It asserts that a Versioned Element is a version of a **Whole Life Configuration**.
Version Released at Milestone	an **Actual Project Milestone** category showing a version of an element to be released
Version Withdrawn at Milestone	an **Actual Project Milestone** showing a version of an element to be withdrawn
Version Succession	a tuple between two **Version Of Configurations** that denotes that one **Version Of Configuration** follows from another
Versioned Element	an abstract type grouping *Resource Performer* and **Service** that allows **Version Of Configuration** to be related to **Actual Project Milestones**
View	an "information item, governed by an architecture viewpoint, comprising part of an architecture description" [ISO 42010] that communicates some aspect of an architecture and expressing the architecture from the perspective of specific stakeholders regarding specific aspects of the architecture entity and its environment [ISO 42020]
View	expresses the architecture of the system-of-interest in accordance with an architecture **Viewpoint** (or simply, **Viewpoint**)
Viewpoint	"conventions for the creation, interpretation and use of an architecture view to frame one or more concerns" [ISO 42010] that governs the creation of views
Viewpoint	frames (to formulate or construct in a particular style or language) one or more **Concerns**. A **Concern** can be framed by more than one **Viewpoint**.
Whole Life Configuration	a set of *Versioned Elements*
Whole Life Configuration Kind	An enumerated type that specifies a kind of a **Whole Life Configuration** (**Service**, *Organizational Resource*, or *Resource Performer*)
Whole Life Enterprise	a purposeful endeavor of any size involving people, organizations, and supporting systems made up of temporal and structural parts

C Appendix C – Acronyms

CDD	Capability Description Document
CIM	Conceptual Information Model
CONEMP	Concept of Employment
CONOPS	Concept of Operations
CONUSE	Concept of Use
DMM	Domain Metamodel
DoDAF	Department of Defense Architecture Framework
DOTMLPF	Doctrine, organization, training, materiel, leadership and education, personnel, and facilities
EA	Enterprise Architecture
ECD	Enterprise Capability Document
HF	Human Factors
HFI	Human Factors Integration
ICD	Initial Capability Document
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Standards Organization
IT	Information Technology
LIM	Logical Information Model
MBSE	Model Based Systems Engineering
MOD	Ministry of Defense (UK)
MOE	Measure of Effect (or Effectiveness)
MOP	Measure of Performance
NAF	NATO Architecture Framework
OMG	Object Management Group
SLA	Service Level Agreement
SOC	Statement of Capability
SysML	Systems Modeling Language
TEPIDOIL	Training, equipment, personnel, information, concepts and doctrine, organization, infrastructure, and logistics
UAF	Unified Architecture Framework
UAFML	Unified Architecture Framework Modeling Language
UML	Unified Modeling Language