Service-Oriented Views in UPDM

MITRE Technical Report

Fatma Dandashi

January 10, 2014

Dept: J84B

Document No: 140007
McLean, VA

MITRE

Service-Oriented Views in UPDM

Keywords:
Service, Architecture, Service Interface, Service Specification, Service, Architecture Description, Viewpoint, View, Department of Defense Architecture Framework (DoDAF), Ministry of Defence Architecture Framework (MODAF), NATO Architecture Framework (NAF)

1 Abstract

This paper provides guidance on how to use the **U**nified **P**rofile for the **D**epartment of Defense Architecture Framework (DoDAF) and **M**inistry of Defence Architecture Framework (MODAF) (UPDM) to describe a set of Service-Oriented viewpoints and associated views. Systems development is moving towards a dynamic producer/consumer environment where components are developed once, deployed to the cloud, and offered for reuse by authorized users. A service-oriented approach to architecture leads to an environment where third party-deployed components can be leveraged and reused through the flexible connectivity of applications implemented as services. Such services have well-defined, platform-independent specifications that hide the underlying technical complexity of the implementation (encapsulation), are self-contained (loosely coupled), and reusable. Creating a platform-independent architecture description as a first step ahead of developing the systems views (which are platform-specific) supports reuse and interoperability. UPDM supports describing such an architecture.

2 Introduction

Systems design in a Systems-Centric Architecture results in monolithic systems — it is difficult to separate functions from the systems that implement them, and to manage these functions as distinct, reusable components. Figure 1 illustrates the tight coupling between elements in the operational views and elements in the systems views.

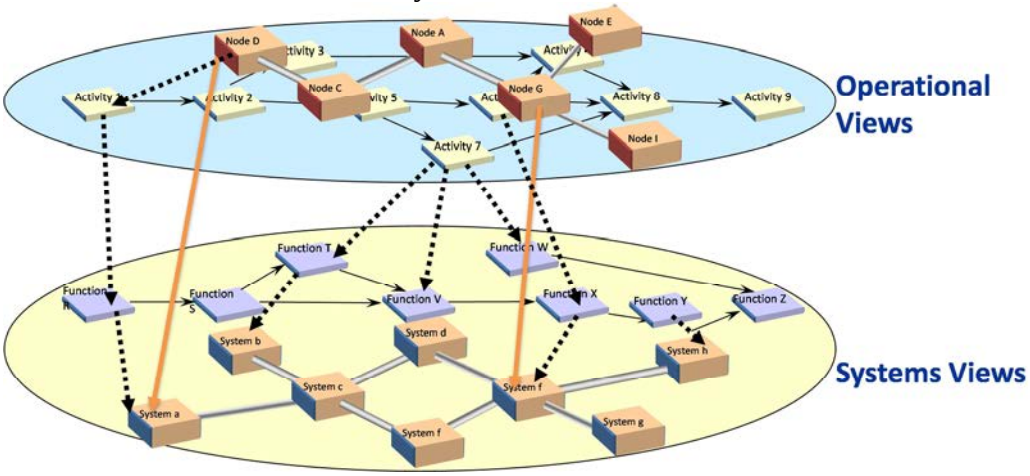

Figure 1. Systems are Tightly Coupled to Operations

A Service-Oriented Architecture (SOA) is an Information Technology architectural style that supports the transformation of a business (enterprise, mission, warfighting capability) into a set of ***linked services***, such that these services can coalesce to accomplish a specific stakeholder capability or operational need. The use of reusable services and standardized data exchanges enables capability stakeholders to quickly adapt to changing conditions and requirements.
In this context, a **Service** is defined as *a mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description.*[OASIS 2006]

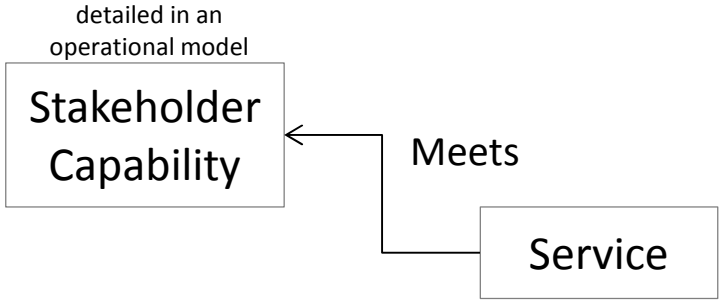

Figure 2: Service Is A Logical Design Unit That Is Aimed At Meeting A Stakeholder's Capability

As illustrated in Figure 2, service-oriented views describe services that support stakeholder capabilities as described in operational views. A service is defined as a unit through which a particular resource (automated or otherwise) provides a useful result to a consuming resource. In UPDM [UPDM 2012], a service is a logical design unit that provides a set of strictly delineated functionality, restricted to answering the what-question, independent of platform, development, or implementation issues. The Service-Oriented viewpoints and associated set of views[1] provide a description of services required to directly support operations. Service views describe services from the subject architecture developer's viewpoint, and constitute a logical design (technology and implementation independent) aimed at addressing operational needs as described in operational views. Services are defined via their interfaces (service descriptions) which are exposed to potential consumers. Non-functional consumer-required performance levels are rigorously described. But what does it mean for an architecture description to be service oriented?

2.1 Service-Oriented Design Principles

When developing service taxonomies for logical design units one must design services with the following principles (adapted from [ERL 2008]) in mind:

1. Services are **Loosely Coupled**. Coupling refers to a connection or relationship between two things. The principle of loose coupling promotes the independent design and evolution of a service's logic and implementation while still guaranteeing baseline interoperability with consumers that have come to rely on the service's capabilities.
2. Service functionality is **Well Encapsulated**:
   a) Services have a **Standardized Service Contract**. This requires that specific considerations be taken into account when designing a service's public technical

[1] This paper uses standard terms defined in [ISO/IEC/IEEE 42010:2011]

interface and assessing the nature and quantity of content that will be published as part of a service's official contract.

b) The **Abstraction Principle** emphasizes the need to hide as much of the underlying details of a service implementation as possible. For a service-oriented model, this principle means that units of functionality required to meet operational needs are only described from an external perspective. The extent of abstraction applied can affect service contract granularity and can further influence the ultimate cost and effort of governing the service.

3. Reuse is a core part of typical service analysis and design processes. The principle of **Service Reusability** emphasizes the positioning of services as enterprise resources with agnostic functional contexts. Numerous design considerations are raised to ensure that individual service capabilities are appropriately defined in relation to an agnostic service context, and to guarantee that they can facilitate the necessary reuse requirements.
4. **Service Composability**. As the sophistication of service-oriented solutions continues to grow, so does the complexity of underlying service composition configurations. The ability to effectively compose services is a critical requirement for achieving some of the most fundamental goals of service-oriented computing. Services are effective composition participants, regardless of the size and complexity of the composition.

2.2 Service-Oriented Viewpoint and Interoperability

In addition to the principles above, the point of describing the architecture of systems that support operational needs (the objective of DoDAF [DODAF 2010] and MODAF [MODAF 2012]) is to guide the development of systems that are interoperable. The introduction of a service-oriented model layer facilitates this objective. Interoperability of services is enabled by the above principles since, as defined in [ERL 2008]:

- Service contracts are standardized to guarantee a baseline measure of interoperability associated with the harmonization of data models.
- Reducing the degree of service coupling fosters interoperability by making individual services less dependent on other components and therefore more open for invocation by different service consumers.
- Abstracting details about the service limits all interoperation to the service interface, governed by the service contract, increasing the long-term consistency of interoperability by allowing underlying service logic to evolve more independently.
- Designing services for reuse implies a high-level of required interoperability between the service and numerous potential service consumers.
- Finally, for services to be effectively composable they must be interoperable. The success of fulfilling composability requirements is often tied directly to the extent to which services are standardized and cross-service data exchange is optimized.

3 Service-Oriented Viewpoints and Associated Views in UPDM

With UPDM, the goal of service-oriented views is to describe a model layer consisting of services that fit the definition of Big SOA as described in Figure 3 below. Little SOA design and implementation takes place during systems design time (systems views).

Little SOA	Big SOA
Architecture models technology mapping and a standardization process that focuses on maximizing consistency across software development projects **Service**: standalone software component with open, remotely addressable interface	**Architecture** model facilitates engagement and an information sharing process that works enterprise wide, driven by a stakeholder's capability/operational need **Service**: delivery of a capability in line with a published and agreed contract

Figure 3. The Aim Is To Describe Services That Are Logical Design Units

Big SOA structures the enterprise into a set of reusable logical design units that are offered as services (loosely coupled, well-encapsulated, etc.) A service is a logical design element to meet an operational need or a stakeholder's capability. It does not specify how the solution will be provided. The services views constitute a model layer that is a Platform Independent Model (PIM)[2], independent of service provider location, or service provider platform.

An architecture design that is service-oriented results in a three layered architecture description that separates operational concerns (needs description) from the Platform Specific Model (PSM)[3] of systems designed to meet these needs, by first describing a PIM and set of views, the Service oriented views form a middle architecture description layer.

The purpose of the service-oriented viewpoints and associated set of views is to develop a logical design model layer that rigorously defines interfaces and details a set of platform independent services that adhere to SOA principles. The operational viewpoints drive the design of the services. These services then drive the design of the systems in a PSM layer, effectively decoupling operations from systems. Figure 4 illustrates this concept. The European Space Agency Architecture Framework [ESAAF 2012] describes a similar approach to the services views.

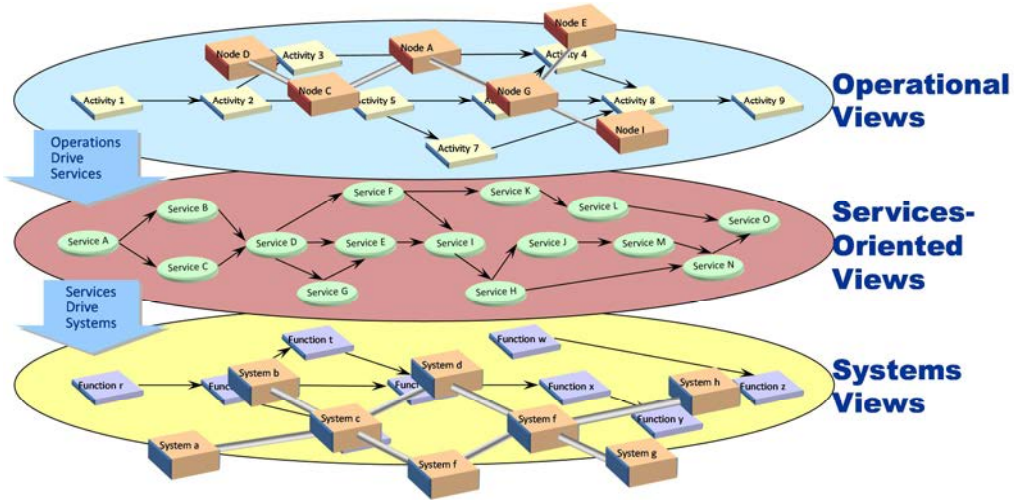

Figure 4. A Three-layered Approach

[2] PIM and PSM are defined terms by OMG's MDA [MDA 2003]
[3] PIM and PSM are defined terms by OMG's MDA [MDA 2003]

3.1 Service-Oriented Viewpoints – Key Views

The following sections discuss a set of viewpoints and associated views that are relevant for building a service-oriented model layer. This set is a selected and adapted list of relevant viewpoints and associated views from DODAF v2.02 and NAF 3.0 [NAF 2011] that are deemed relevant to constructing a service-oriented model layer as described in this paper. In DoDAF, services views are prefixed with SvcV, while NAF prefixes the view with NSOV. This paper provides a description of a set of service-oriented views by providing a name (listing both DODAF as well as NAF short and full names), a definition that is based on a synthesis of what is required to describe a service oriented model layer as described above, and a sample diagram developed using UPDM.

CAVEAT: Services Viewpoint in DoDAF defines a set of service views as an alternative Systems view set, and includes SvcV-1...SvcV-10c. This is not the intended purpose of a service-oriented viewpoint. The sections below deviate from the services viewpoint treatment in DoDAF and provide a technically sound alternative to development of a service-oriented model layer consisting of an integrated set of viewpoints and associated views. While it is true that each model layer (operational, services, and systems) must include a set of structure, behavior, and performance viewpoints described in a set of associated views (see Figure 5), the DoDAF-defined set of services views are not adequate to create a service-oriented model layer as defined in the sections above.

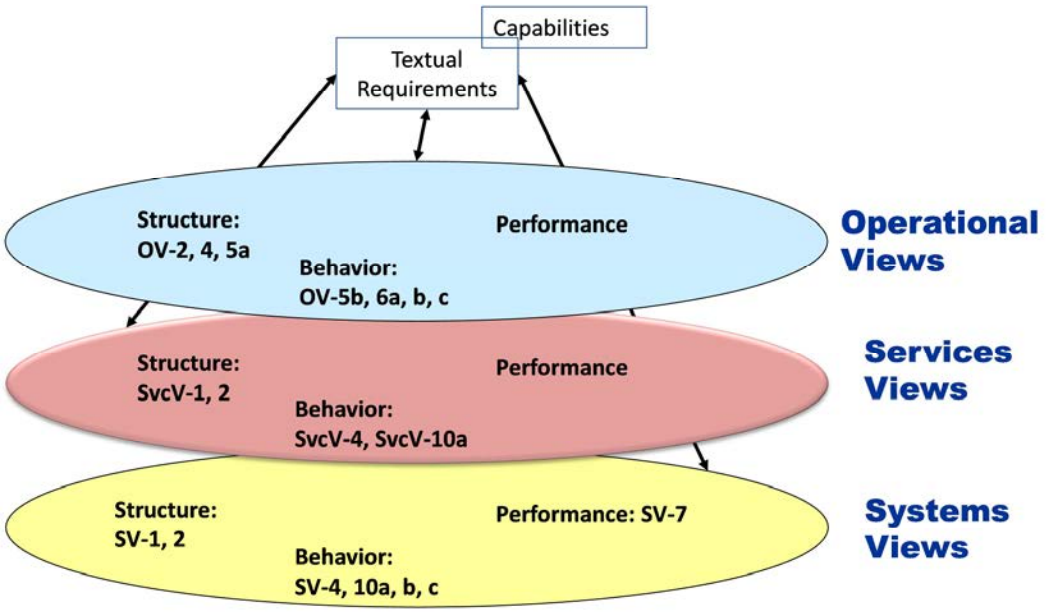

Figure 5. Three Layers, Pattern Repeats

To illustrate, Table 1 below lists the various views defined for each model layer as a set of corresponding three-layered rows.

Table 1. View Correspondence for Three Layers

Diagram Type	Main Element Type	DoDAF Operational Views (OV)	Required Service-Oriented Views	DoDAF Systems Views (SV)
Structure	**Data**	DIVs (1-3)	DIVs (1-3)	DIVs (1-3)
Structure	**Taxonomy**	*Capability View*: CV-2/NCV-2 Capability Taxonomy	NSOV-1 Service Taxonomy	
Structure	**Interface**	OV-2: Operational Resource Flow Description	NSOV-2 Service Definitions. SvcV-1, which DoDAF defines as: "The identification of services, service items, and their interconnections."	SV-1 Systems Interface Description
Structure	**Network Infrastructure**		SV-2 is a network diagram that adds communications infrastructure detail to the SV-1 diagram, an equivalent SvcV-2 is not relevant for services.	SV-2 Systems Resource Flow Description
Structure	**Participant**	OV-4: Organizational Relationships Chart		New to DoDAF 2.0: SV-1 shows both automated systems as well as human participants
Structure	**Behavior-Decomposition**	OV-5a: Operational Activity Decomposition Tree		SV-4 Systems Functionality Description (Decomposition)
Behavior	**Activity**	OV-5b: Operational Activity Model	SvcV-4/NSOV-5 Services Functionality Description/ Service Behavior	SV-4 Systems Functionality Description
Behavior	**Rule**	OV-6a: Operational Rules Model	SvcV-10a Services Rules Model /Service Constraints View	SV-10a Systems Rules Model
Behavior	**State Machine**	OV-6b: State Transition Description	SvcV-10b Services State Transition Description	SV-10b Systems State Transition Description
Behavior	**Event Trace**	OV-6c: Event-Trace Description	SvcV-10c/NSOV-4 Services Event-Trace Description/ Service Orchestration	SV-10c Systems Event-Trace Description
Performance	**Measure**	Measures of Effectiveness (MOEs) associated with structural operational elements and capabilities	Measures of Performance (MOPs) associated with services	SV-7 Systems Measures Matrix

A detailed discussion of each of the service views listed above follows. Each view subsection consists of a definition for the view, a purpose, and an example diagram from a Search and Rescue (SAR) example (for a more complete example of SAR using UPDM, see [UPDM 2012]. The information provided below was adapted from a variety of resources [DODAF, NAF, MODAF, UPDM], and constitutes this author's synthesized distillation. The names of the views

are listed in terms of DoDAF v 2.02 short and long names and the corresponding NATO Architecture Framework (NAF) names for the readers' benefit.

3.1.1 NAF NSOV-1: Service Taxonomy

Description: Specifies a hierarchy of services. The elements in the hierarchy are service specifications (rather than service implementations), and the relationships between the elements are generalization/specialization.
Purpose: Logically group identified services and identify spheres of control for service provision across multiple providers. The service taxonomy defines a service classification hierarchy and a grouping of these services, as required to meet an operational need. The identified groups should organize the services to reflect the purpose of the taxonomy itself such as services to be leveraged (provided by enterprise infrastructure), services to be outsourced (consumed from third party providers), and services to be provided (further described in the systems views). A sample NSOV-1 for a SAR capability is shown in Figure 6.

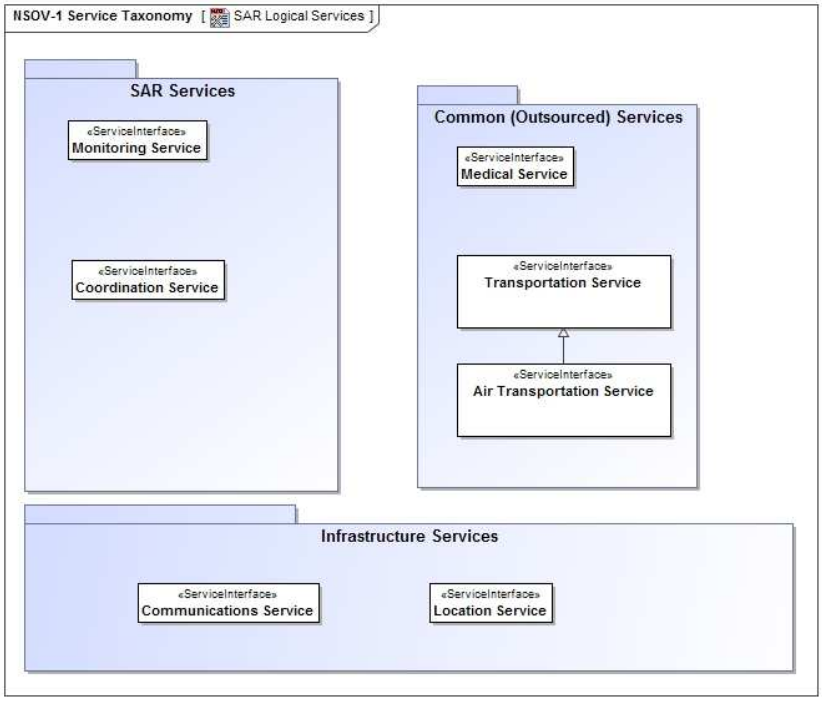

Figure 6. Services taxonomy for SAR

3.1.2 SvcV-1/NAF NSOV-2: Services Context Description/Service Definition

Description: Defines the interfaces provided or consumed by a service. These interfaces will be used to define access ports on corresponding systems views.
Purpose: To define service interfaces. A Service presents one or more interfaces to consumers (a "consumer" being any agent capable of using the service - a person, an organization, a system, or another service). In this case, the architect specifies provided interfaces. A service may also be capable of using interfaces exposed by other services, and the architect may specify these as consumed interfaces.
NOTE: DoDAF defines a SvcV-2 as a "Services Resource Flow Description: A description of Resource Flows exchanged between services." This is redundant with DoDAF's definition of SvcV-1 which includes a description of service "interconnections;" and it is meaningless in

describing a service-oriented viewpoint, where service-to-service flows (point-to-point interfaces) are not emphasized.
A service interface definition in UPDM specifies the connection point (similar to a plug in the wall that supplies electricity), to be represented as a SysML [SysML 2012] port in the corresponding SV-1 diagram, through which a service provider or consumer may provide or consume an element (e.g. information) to/from this service. This "interface" must include the following information:

- a name for the service interface
- the functionality offered
- the "type" of the element that is consumed and provided

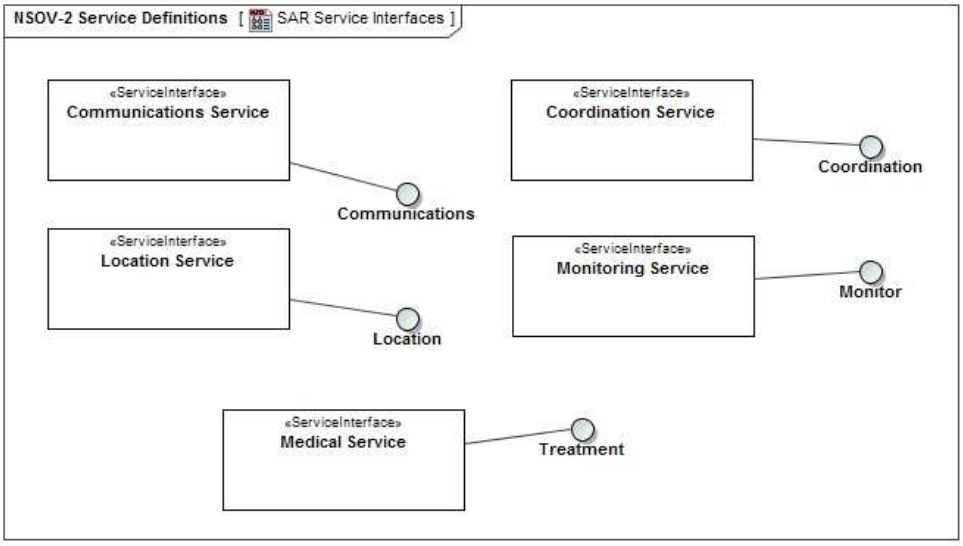

Figure 7. Service Definitions for SAR example

3.1.3 SvcV-4/NSOV-5 Services Functionality Description/ Service Behavior

Description: Defines the behavior of a service in terms of the functions it is expected to perform and the flows (control and object flows) across service functions.
Purpose: Specifies the functions and behavior of services.
An NSOV-5a may be developed to show the set of service functions that are then used in a SysML activity diagram (NSOV-5b). There is no hierarchy to the functions shown in the example (see Figure 8).

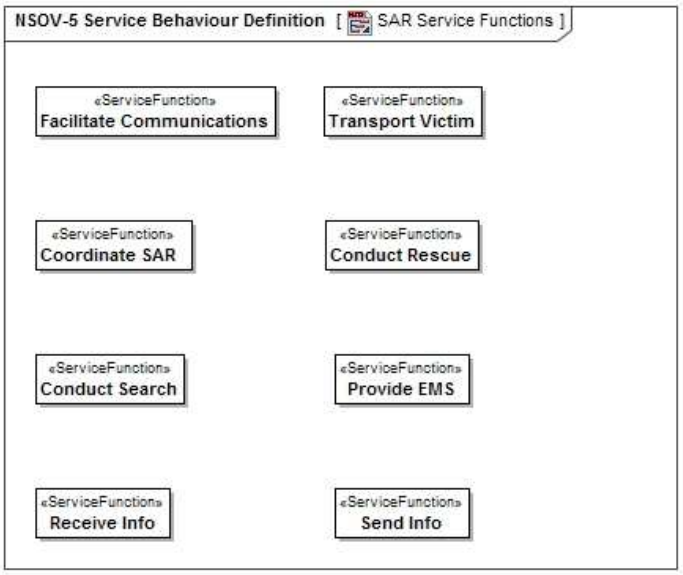

Figure 8. Example NSOV-5a

An NSOV-5b shows service functions and flows between service consumers and service providers. Swimlanes represent services defined in NSOV-1, service functions appear within swimlanes and reflect the behavior of services as they interact in support of some operational need. Figure 9 shows an example of this behavior view.

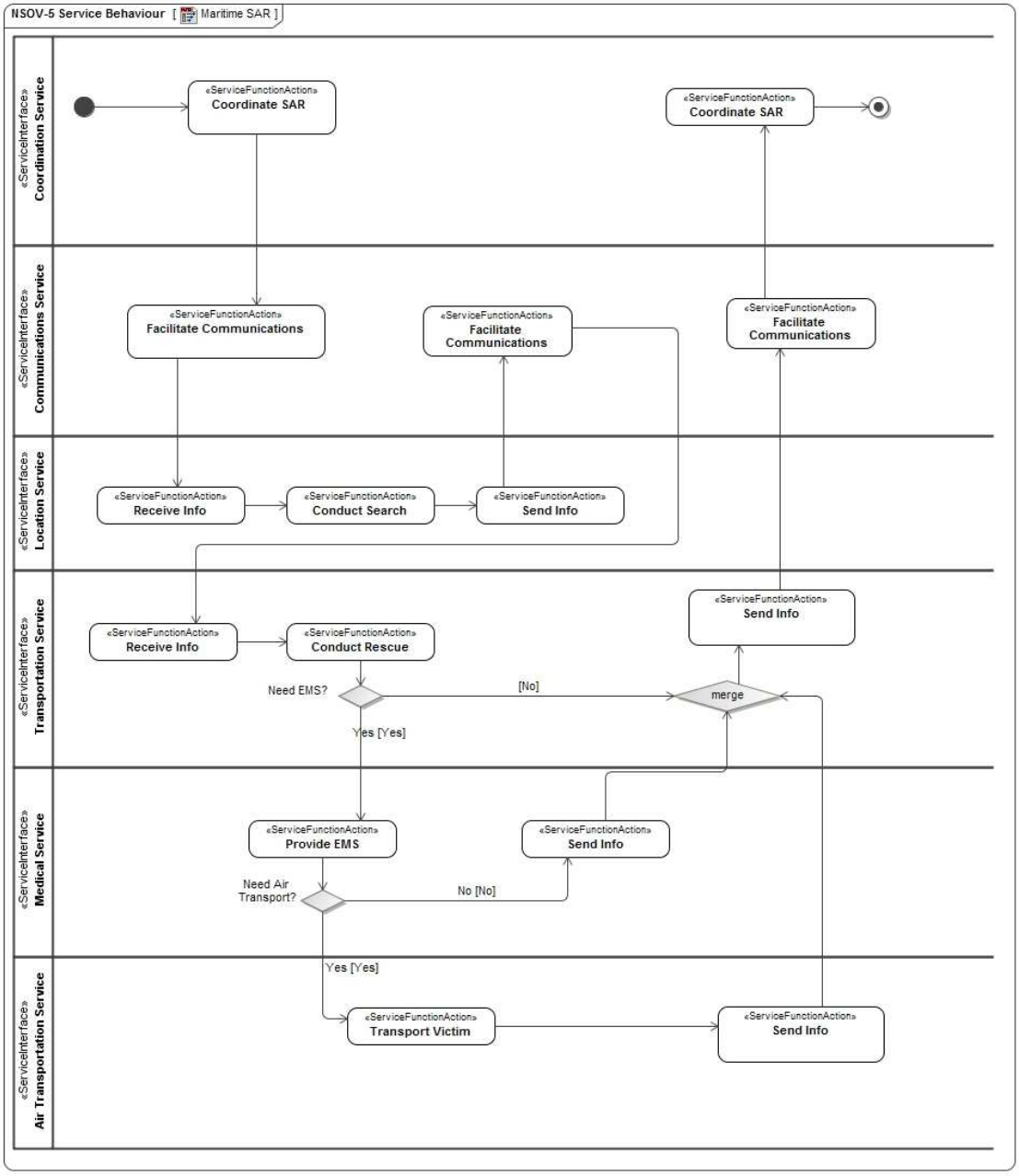

Figure 9. Example NSOV-5b for SAR

3.1.4 SvcV-10a/NSOV-4a: Services Rules Model /Service Constraints View

Description: Specifies the policies regarding security, commercial conditions, and applicable laws for a provided service.

Purpose: To specify constraints to implementations of services.

3.1.5 SvcV-10c/NSOV-4c Services Event-Trace Description/ Service Orchestration

Description: Specify how a service interacts with external agents, and the sequence and dependencies of those interactions.

Purpose: Describes how services may be orchestrated together to support operational processes. Orchestration at this level focuses on the combined use of services, by service consumers, in support of the execution of operational activities and processes.

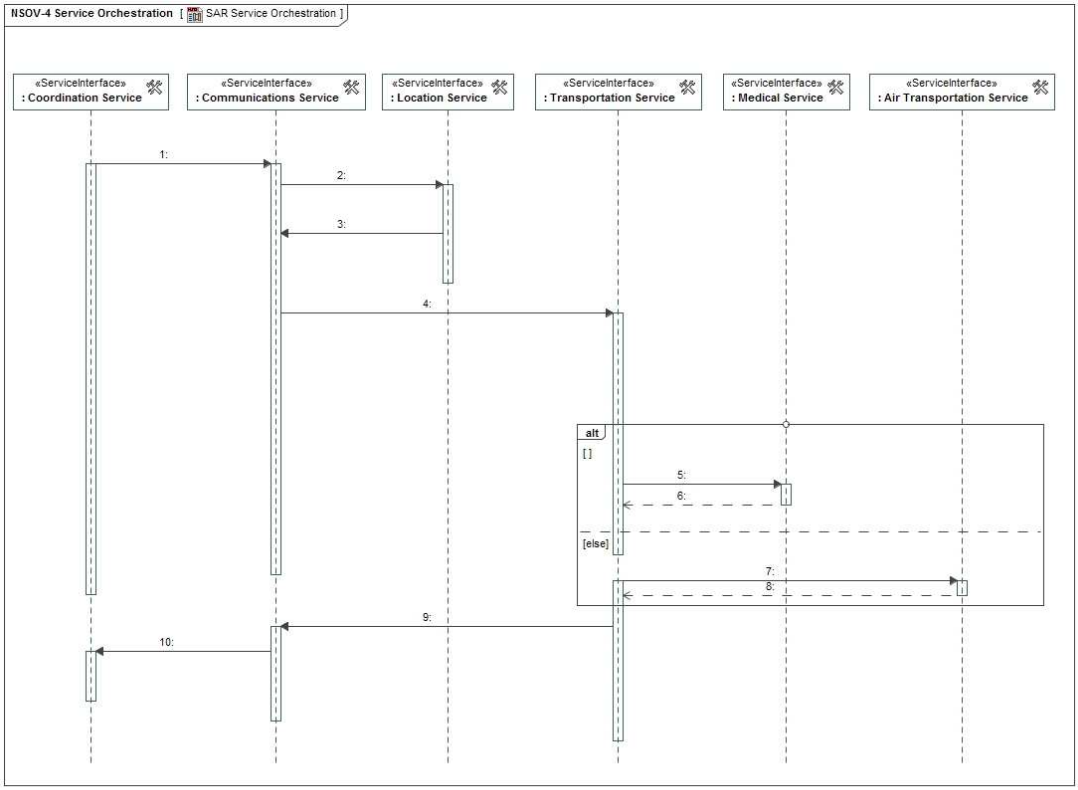

Figure 10. Example NSOV-4c for SAR

3.2 Other Related Views

A key model integrator view is **CV-7/NCV-7 Capability to Service Mapping** which establishes the relationships between a capability and services - defined as dependency relationships. The matrix shows which services support which stakeholder capability.
The following section provides a brief discussion of SV-1 as it relates to the services defined in the service-oriented views. This discussion does not constitute an exhaustive description of the systems viewpoints and views.

3.3 Systems (Platform-Specific) Views or Service Implementations

3.3.1 SV-1/NSV-1 Systems Interface Description/Resource Interaction Specification

Description: The identification of systems, system items, and their interconnections.
Purpose: SV-1 shows systems, their parts (subsystems), and their interfaces that cross organizational boundaries (key interfaces). Some systems can have numerous interfaces. Detailed versions of SV-1 views may be developed as needed, for use in system acquisition, as part of requirements specifications, and for determining system interoperability at a finer level of technical detail.

In the systems views, a modeler shows how a system (a provider) provides a service by exposing its interfaces as a port that consumers can use to consume the service. In the SAR example, we show the following sample SV-1/SysML block definition diagram (bdd).

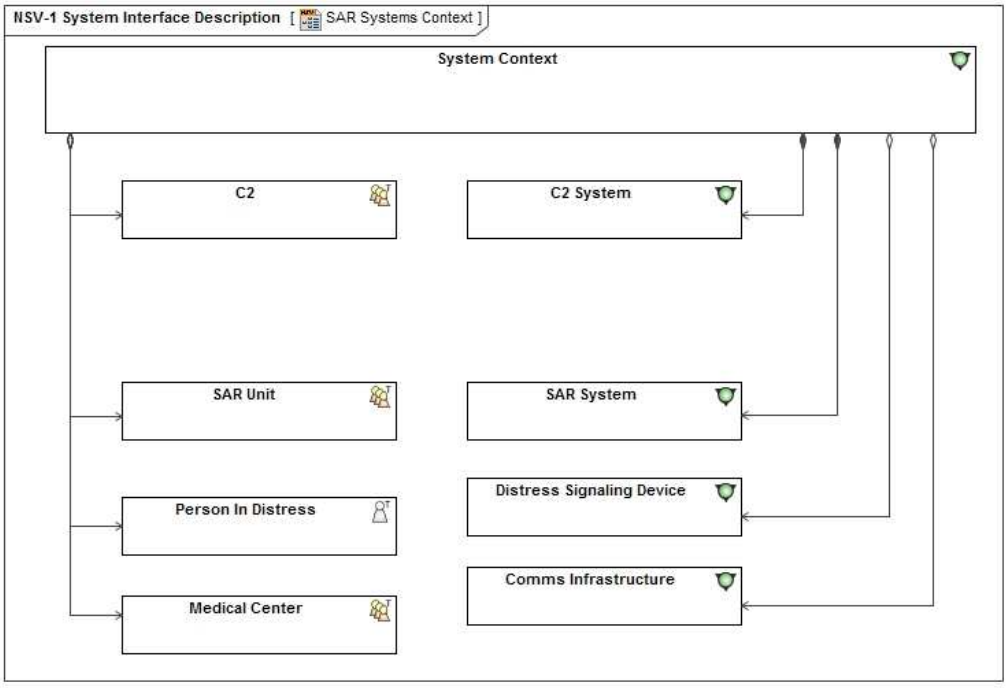

Figure 11. SV-1 bdd for SAR Example

The corresponding parts diagram, an SV-1/SysML internal block diagram (ibd), is shown in Figure 12. The figure shows service interfaces that were defined in the SOVs as ports on the system providers and system consumers. The systems shown are components that implement and expose a service and components that expect to consume a service.

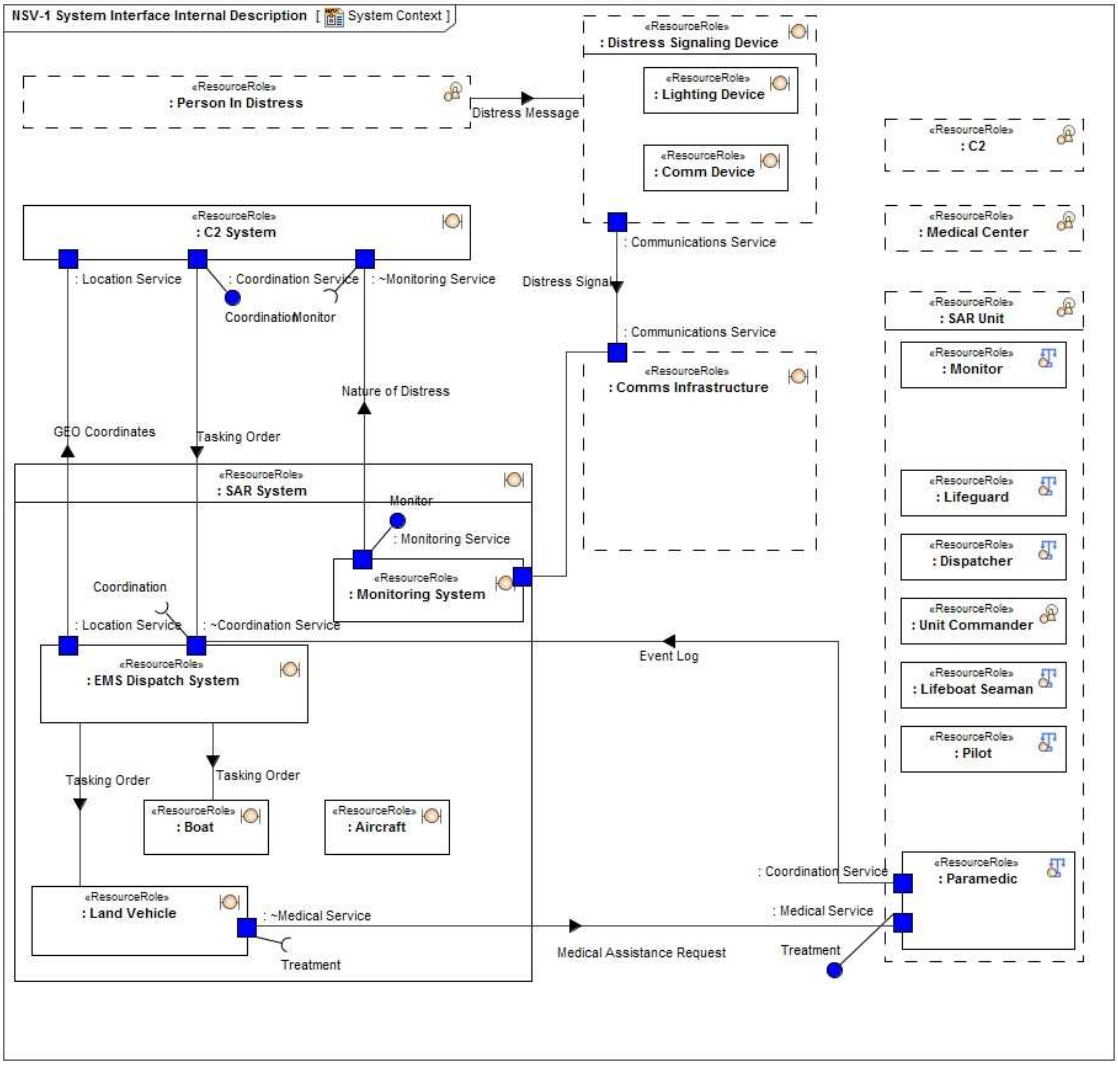

Figure 12. SV-1 ibd for SAR Example

4 Summary

This paper provided guidance on how to use UPDM to describe a set of Service-Oriented viewpoints and associated views that describe services with well-defined, platform-independent specifications (encapsulation), are self-contained (loosely coupled), and reusable. Creating a platform-independent architecture description as a first step ahead of developing the systems views (which are platform-specific) supports reuse and interoperability. Absent such a model layer, systems design leads to describing systems and system components where it is difficult to separate functions from the systems that implement them, and more difficult to manage these functions as distinct, reusable components.

In UPDM, the Service-Oriented viewpoints and associated set of views provide a description of services required to directly support operations. Services are described from the subject architecture developer's viewpoint, and constitute a logical design (technology and implementation independent) aimed at addressing the operational needs as described in operational views. Services are defined via their interfaces (service descriptions) which are

exposed to potential consumers. Non-functional consumer-required performance levels are rigorously described.

References

[DODAF 2010] DoDAF Architecture Framework Version 2.02, DOD CIO, Architecture & Infrastructural Directorate, August 2010, http://dodcio.defense.gov/dodaf20.aspx

[ERL 2008] Erl, Thomas, "SOA Principles of Service Design," by Prentice Hall/Pearson PTR, ISBN: 0132344823

[ESAAF 2012] System of Systems Development Environment Consolidation – ESA-AF Reference, Issue 1.2, 07 May 2012,VEGA Space GmbH, www.vegaspace.de

[ISO/IEC/IEEE 42010:2011] Systems and software engineering – Architecture description http://www.iso.org/iso/catalogue_detail.htm?csnumber=50508

[MDA 2003] Overview and guide to OMG's architecture, Model Driven Architecture, OMG, http://www.omg.org/cgi-bin/doc?omg/03-06-01

[MODAF 2012] The Ministry of Defence Architecture Framework (MODAF), https://www.gov.uk/mod-architecture-framework

[NAF 2011] NATO Architecture Framework (NAF), Version 3, NATO C3 BOARD (AC/322-D(2007)0048) http://www.nhqc3s.nato.int/HomePage.asp

[OASIS 2006] OASIS Reference Model for Service Oriented Architecture 1.0, Official OASIS Standard (Normative PDF), Oct. 12, 2006

[SysML 2012] Systems Modeling Language, OMG, http://www.omgsysml.org/

[UPDM] Object Management Group, Unified Profile for DoDAF and MODAF Specification, Version 2.1, August 2013 http://www.omg.org/spec/UPDM/2.1/