The MODAF Service Oriented Viewpoint

Viewpoint Summary

The Service Oriented Viewpoint provides a perspective that enables the specification of services[1] which are to be used in a Service Oriented Architecture (SOA).

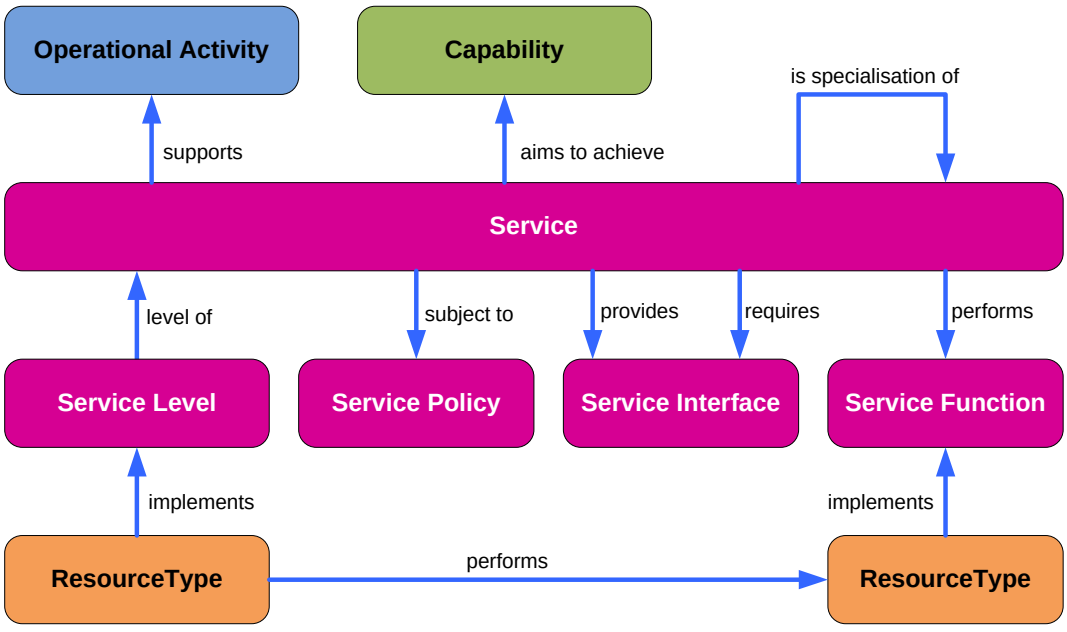
The viewpoint consists of 7 Service Oriented Views (SOVs) that specify the services used in an architecture (their behaviour and the interfaces they provide and require), the capabilities that the services deliver, and the policy governing the use of the services. It should be noted that the views do not focus on the detailed implementation of the service, but on the requirement the service fulfils – i.e. there may be many different implementations of a given service that is specified in the SOVs.

The SOVs are based on the same meta model elements as the NATO Architecture Framework (NAF) v3.1. However, a different set of views has been defined for MODAF. The definition of the MODAF SOVs include equivalency statements to indicate where the MODAF views differ from those in NAF.

In addition, service elements have been added to some existing MODAF views, specifically:

- OV-2, Operational Node Relationship Description.
- OV-5, Operational Activity Model – used for service orchestration.
- OV-6c, Operational Event-Trace Description.
- SV-5, Function to Operational Activity/Service Function Traceability Matrix.
- SV-12a&b, Service Composition and Implementation.

The diagram below summarises how services are linked to other elements in MODAF.



It should be noted that the SOVs are intended for specifying services for use in an SOA (i.e. loose-coupled, opaque service specifications). Services should not be used for simple interface management – SV-1 provides the concept of Resource Interface for this purpose.

[1] In MODAF terms, a service is an implementation-independent specification of a packaged element of functionality.

Views

There are 7 SOVs, including sub-views, that make up the Service Oriented Viewpoint:

SOV-1 - Service Taxonomy

The SOV-1 specifies a hierarchy of services. The elements in the hierarchy are M3 Services (i.e. service specifications rather than service implementations), and the relationships between the elements are specialisations – i.e. one service is a special type of another. Service attributes, interfaces and constraints are inherited down a service taxonomy – e.g. if Service A is a specialisation of Service B then it also inherits all the features of Service B.

NAF V3 Equivalency

The equivalent NAF v3 view to SOV-1 is NSOV-1, Service Taxonomy.

Background

The purpose of an SOV-1 is to provide a governance structure for a Service-Oriented Architecture. Along with SOV-2, Service Interface Specification, it defines a standard library of service specifications for an enterprise, which service implementers are expected to conform to.

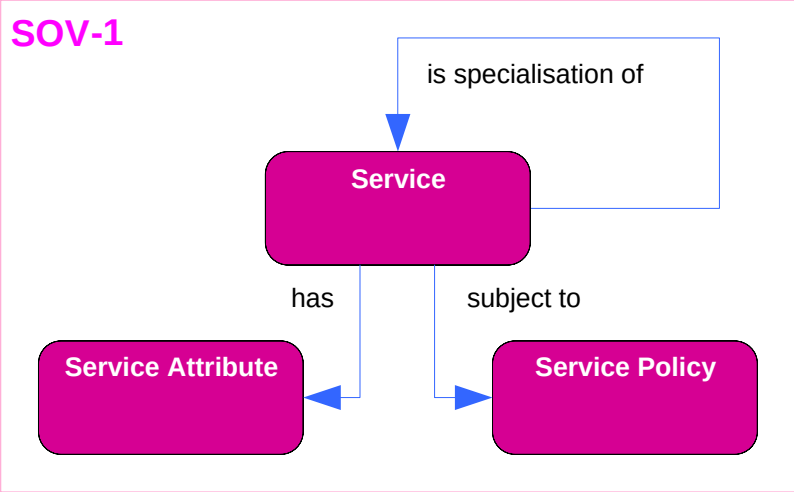
Usage

- SOA governance.
- Identification of services.
- Service planning.
- Service audit.
- Service gap analysis.
- Providing reference services for architectures.
- Tailoring generic services for specific applications.

Data objects

The data in an SOV-1 can include:

- Service.
- Service Generalisation (the specialisation relationship).
- Service Attribute.
- Service Policy (optional, also shown in SOV-4a, Service Constraints).



Relationships between Key Data Objects (Simplified from M3)

Representation

- Tabulation.
- Hierarchical (connected shapes).
- UML class diagram.

Detailed Product Description:

In MODAF terms, a service is an implementation-independent specification of a packaged element of functionality and/or capability.

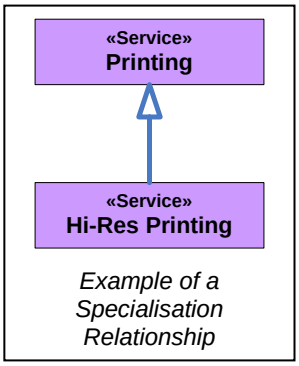
There is, however, potential for confusion between services and capabilities. To help clarify this:

> The key indicator of a service is that it provides a standard interface to consumers. This means that services may be used as "wrappers" for one or more capability in order to provide a standard method of access to the capability. A well-designed service taxonomy provides a set of specifications for capability providers to adhere to.

An SOV-1 depicts services, specialisation relationships between services, service attributes and service policy (i.e. constraints). A service taxonomy persists over time (an architect may wish to specify historical, current or future services) and may be referenced by multiple architectures.

In SOV-1, services are only defined in the abstract, i.e. SOV-1 does not specify how a service is to be implemented. An SOV-1 is structured as a specialisation hierarchy of services, with the most general at the root and most specific at the leaves.

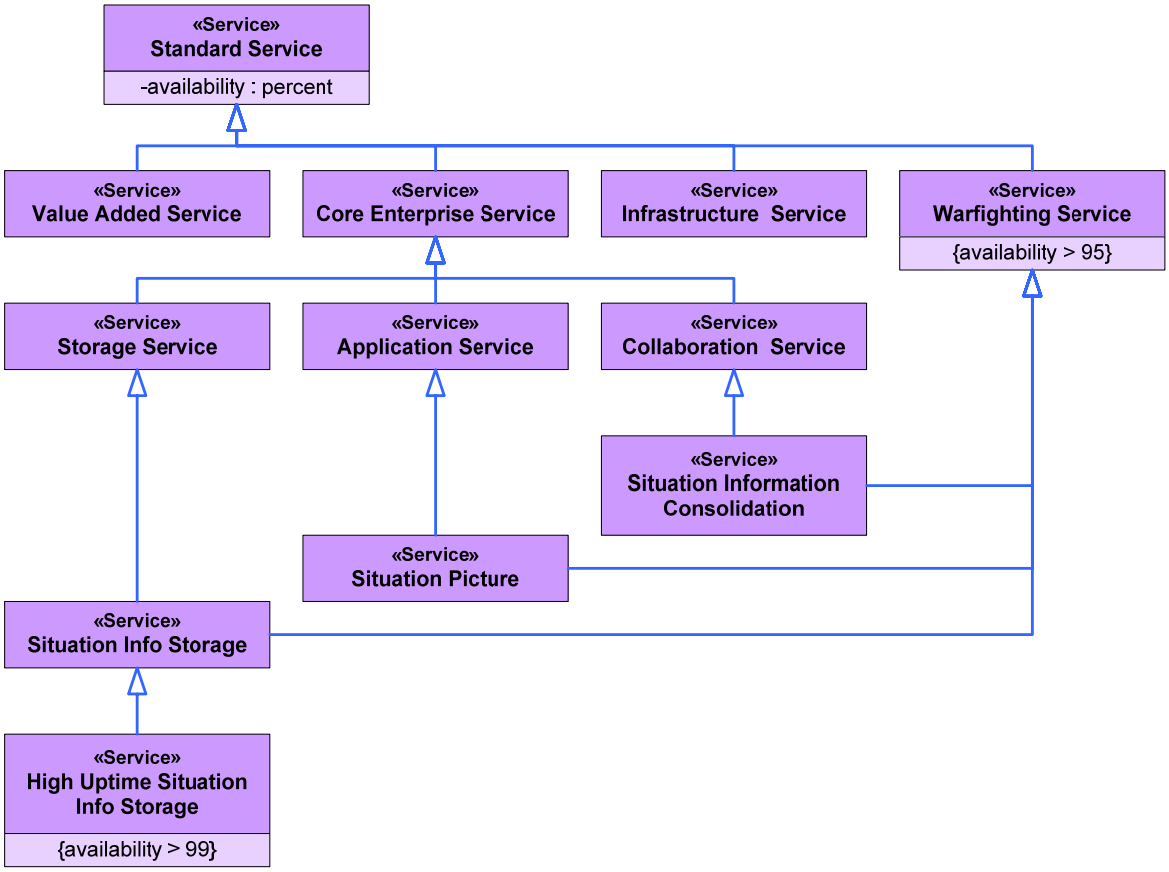
In contrast to AV-2, Integrated Dictionary, an SOV-1 is structured using only one type of specialisation relationship between elements: super-subtype. A super-subtype relationship is a relationship between two classes with the second being a pure specialisation of the first. Any service that specialises from another must implement all the functionality of its parent, and provide all the same input and output interfaces of its parent; in other words, any specialised service shall be entirely compatible with its parent (however, it may add functionality and interfaces). For example, if a service, "Printing", requires input of paper size and ink colours, a service, "Hi-Resolution Printing", that specialises from it must also accept these parameters and produce equivalent behaviour when initiated.



Example of a Specialisation Relationship

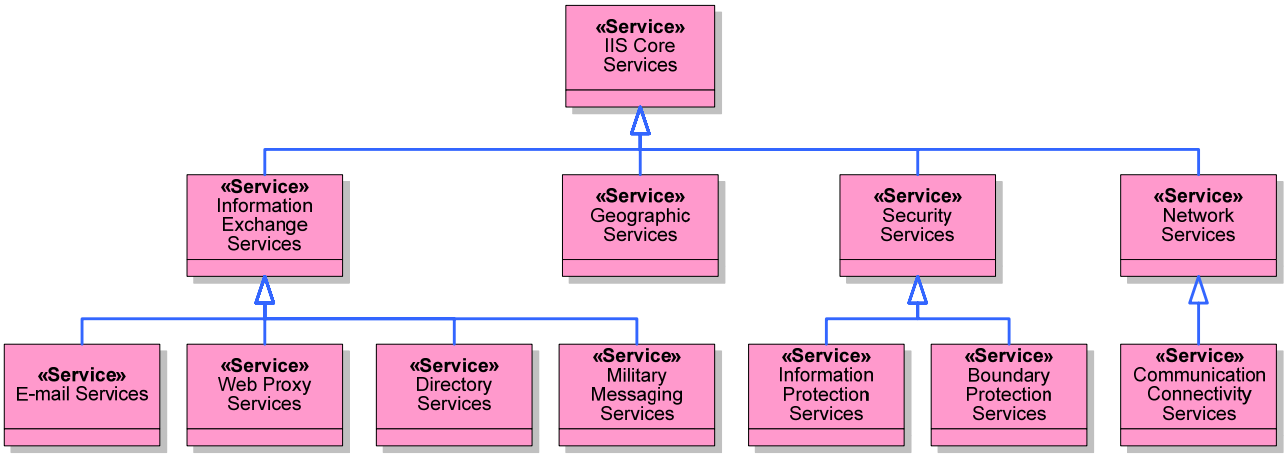
Services may have attributes and constraints (service policy) defined against them. Attributes are inherited by specialised services. Where an attribute is specified for a service, implementations of

that service shall specify their values for the attribute. The example below shows an availability attribute defined against the top service. All other services inherit that attribute, and the WarfightingService sets a constraint (service policy) that the availability shall be greater than 95%. This policy is then inherited by the three situational awareness services. Note that policy may be overridden in specialised services.



Sample Service Hierarchy

UML is a useful modelling language in which to develop service taxonomies as the object oriented approach naturally includes the concept of generalisation-specialisation.



UML Sample Service Hierarchy

SOV-2 - Service Interface Specification

The SOV-2 defines the interfaces provided and required by a service.

NAF V3 Equivalency

The equivalent NAF v3 view to SOV-2 is NSOV-2, Service Definitions. However the MODAF view is more restricted in that it only specifies the service interfaces (attributes are specified in SOV-1).

Background

A service presents one or more interfaces to consumers (a “consumer” being any agent capable of using the service; i.e. a person, an organisation, a system or another service). A service may also be capable of using interfaces exposed by other services, and the architect may specify these as used interfaces.

Specifying the interfaces that a service provides and requires defines compatibility between services - e.g. if Service A provides interface X, and Service B can use Interface X, then Service B can call upon at least some of the functionality of Service A.

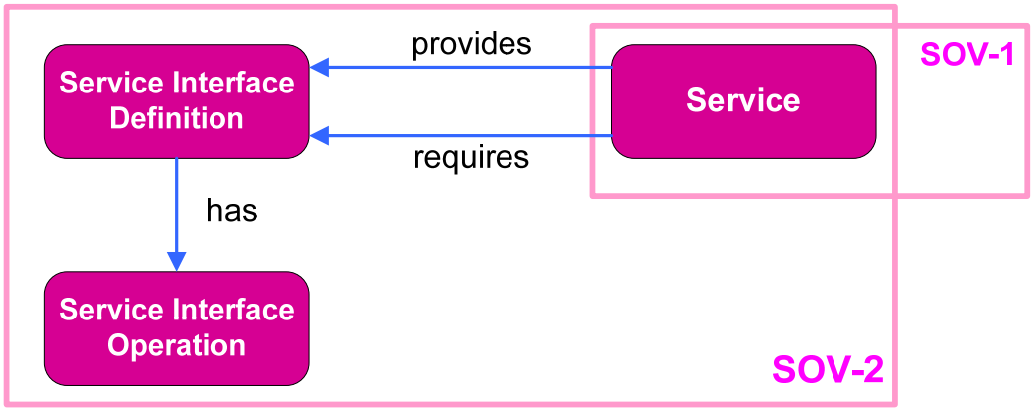
Usage

- SOA governance.
- Detailed service specification.
- Service interoperability.

Data objects

The data in an SOV-2 can include:

- Service.
- Service Interface.
- Service Interface Operation.
- Service Interface Parameter.



Relationships between Key Data Objects (Simplified from M3)

Representation

- Tabular.
- UML.

Detailed Product Description

Service interfaces are defined in terms of their operations (methods of access) and parameters (data that must be passed to the service, or produced by the service). The interfaces and their operations all have names. Parameters may be simple types (text, numbers, Boolean) or typed by an entity that shall be specified in a Physical Schema (SV-11).

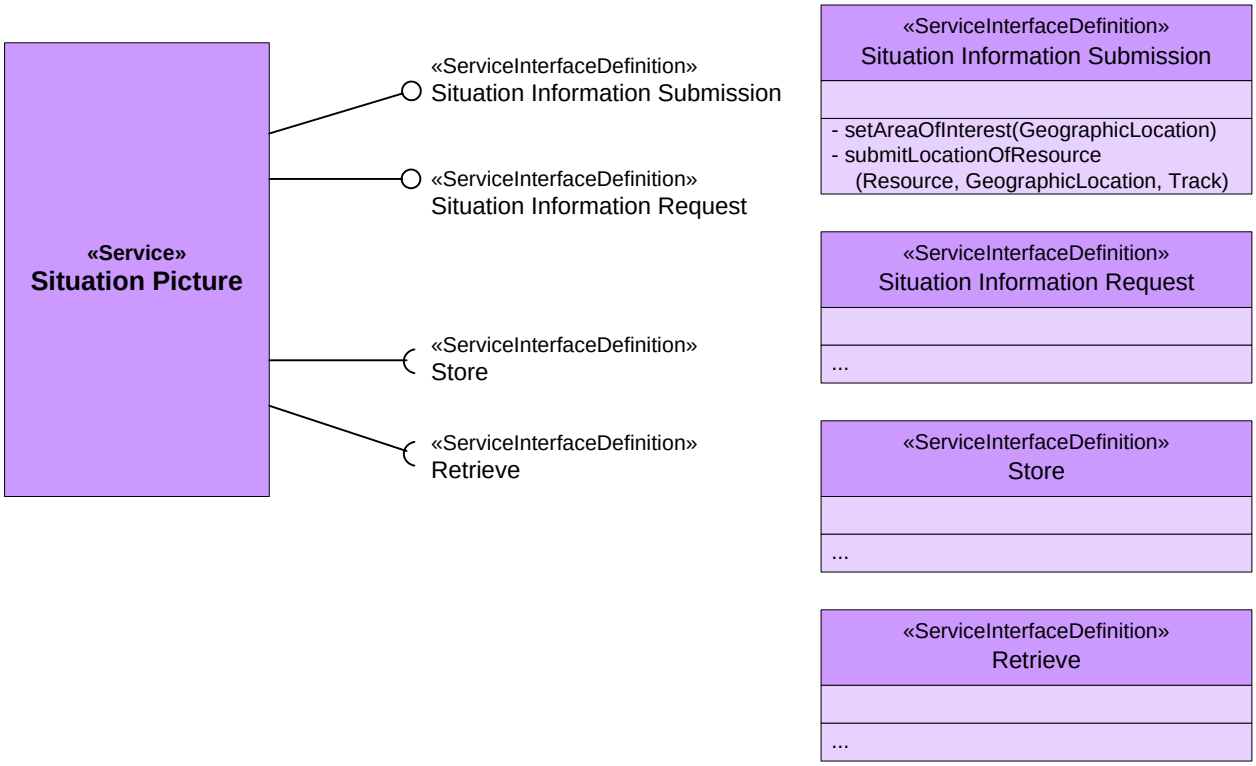
SOV-2 products can be presented in tabular form, or as diagrams (usually UML). If a tabular approach is used, the first five columns are mandatory and should be in the following order:

1) Service Name.
2) Interface Name.
3) Provided / Used (note that if the service both provides and uses a particular interface, this should be recorded as two entries in the table).
4) Operation Name.
5) Parameters.

Service	Interface	Provided / Used	Operation	Parameters
Situation Information Consolidator	Situation Information Submission	P	setAreaOfInterest	Geographic location
			submitLocationOfResource	Resource, geographic location, track
	Situation Information request	P	requestPictureForArea	Geographic location, situation picture
			requestLocationOfResource	Resource, geographic location
	Store	U	storeInformation	Situation info package
	Retrieve	U	retrieveInformation	Geographic location, time, situation info package
Situation Information Storage	Store	P	storeInformation	Situation info package
	Retrieve	P	retrieveInformation	Geographic location, time, situation info package
Situation picture	…			

SOV-2 Tabular Representation

If presented in UML format, the information above would look like this:



SOV-2 UML Representation

SOV-3 - Capability to Service Mapping

The SOV-3 specifies the capabilities that services provide.

NAF V3 Equivalency

The NAF v3 equivalent view is NCV-7, Capability to Service Mapping.

Background

An SOV-3 presents a simple mapping of services to capabilities, showing which capabilities a given service provides.

Note that in MODAF v1.2.003, the semantics of the relationship between services and capabilities was unclear. If more than one service mapped to a given capability, it was not clear if each of those services provided the capability in and of itself, or if all the services were required together in order to deliver the capability.

From v1.2.004, the relationship between capability and service indicates that the service provides the capability. If more than one service maps to a capability, each of those services provides the capability in and of itself. Should the architect wish to express the need to bring together multiple services to deliver a capability, this should be expressed in OV-2, Operational Node Relationship Description, (mapping the capability to a node) and OV-5, Operational Activity Model (orchestrating the services against the activities performed by the node).

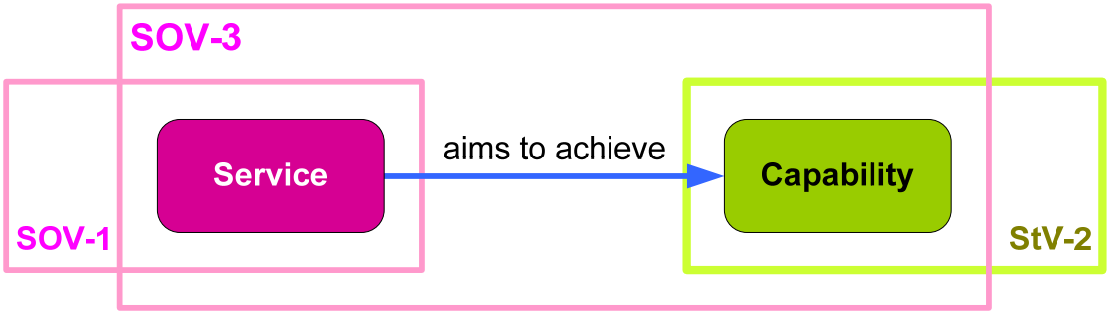
Usage

- Service specification & planning.
- Governance.

Data objects

The data in an SOV-3 can include:

- Service.
- Capability.
- Service Aims to Achieve (relationship from Service to Capability).



Relationships between Key Data Objects (Simplified from M3)

Representation

- Tabular.
- UML.

Detailed Product Description

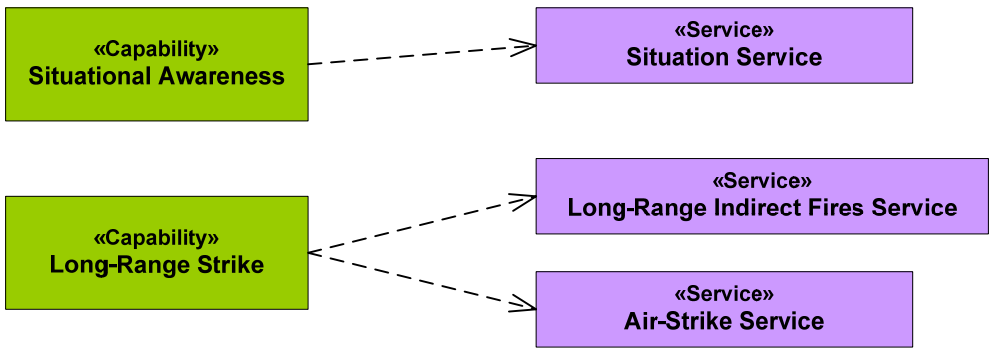
An SOV-3 can be presented as a matrix with capabilities on one axis and services on the other.

	Services		
Capabilities	Long Range Indirect Fires Service	Air-Strike Service	Situation Service
Long-Range Strike	**X**	**X**	
Situational Awareness			**X**

Tabular Representation of SOV-3

The relationship between capability and service is many-to-many. A given service may provide one or more capabilities, or a given capability may be provided by more than one service. Note that if a combination of services are required deliver a certain capability, this should be modelled using a combination of OV-2 (to map the capability to a node) and OV-5 (to orchestrate the services against activities performed by the node).

Alternatively, SOV-3 can be presented as a diagram showing tracing relationships from capabilities to services.



UML Representation of SOV-3

SOV-4a - Service Constraints

The SOV-4a specifies constraints that apply to implementations of services.

NAF V3 Equivalency

SOV-4a has no direct equivalent in NAF v3, though service policy constraints can be shown in NSOV-1, Service Taxonomy.

Background

To better enable consistency and re-use of service specifications, it is important to set constraints on how a service should behave. An SOV-4a specifies constraints against services to which implementations of must conform.

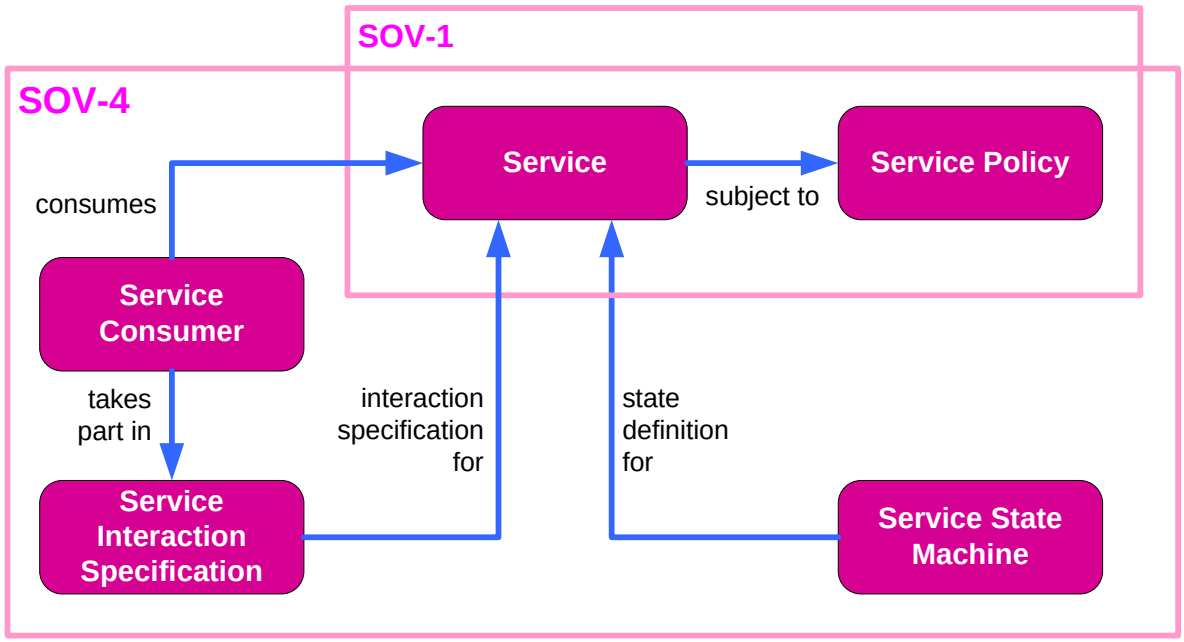
Usage

- Service specification.
- Service governance.

Data objects

The data in an SOV-4a can include:

- Service.
- Service Policy.



Relationships between Key Data Objects (Simplified from M3)

Representation

- Tabular.
- UML.

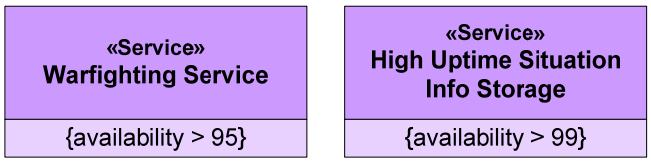
Detailed Product Description

SOV-4a products are usually tabular, with services as rows and constraints as columns.

	«ServiceConstraint» availability	
«Service»		
Warfighting Service	> 95%	
High Uptime Situation Info Storage	> 99%	

SOV-4a tabular representation

It is also possible to present the constraints as adornments to services in a diagram (e.g. a compartment in a UML class).



SOV-4a UML representation

SOV-4b - Service State Model

The SOV-4b specifies the possible states a service may have, and the possible transitions between those states.

NAF V3 Equivalency

SOV-4b has no direct equivalent in NAF v3.

Background

In specifying a service, it is often necessary to specify the allowable states so as to constrain how implementations of the service will behave. SOV-4b is a specification of those states, and the possible transitions between them.

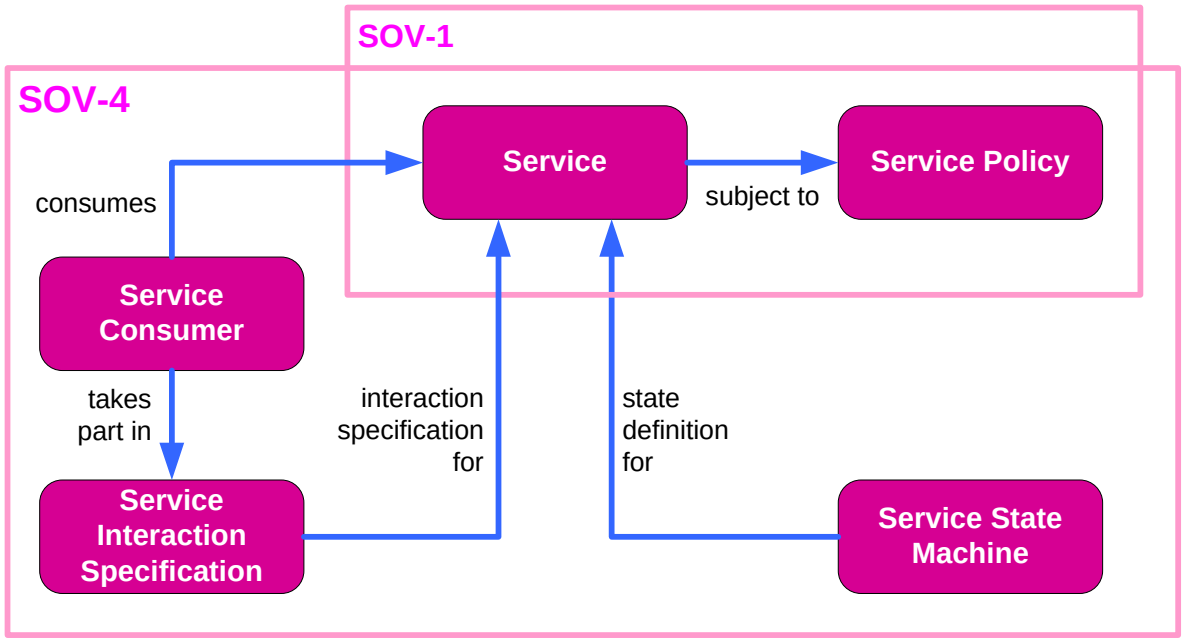
Usage

- Service specification.

Data objects

The data in an SOV-4b can include:

- Service.
- Service State Machine.



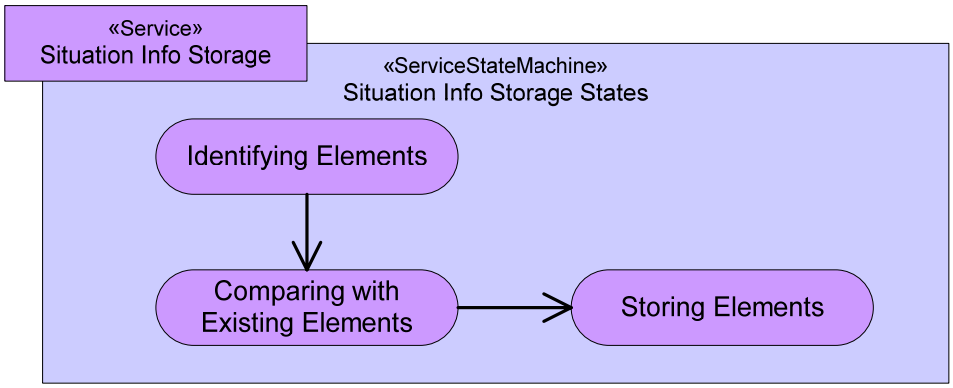
Relationships between Key Data Objects (Simplified from M3)

Representation

- UML.
- Other state transition models.

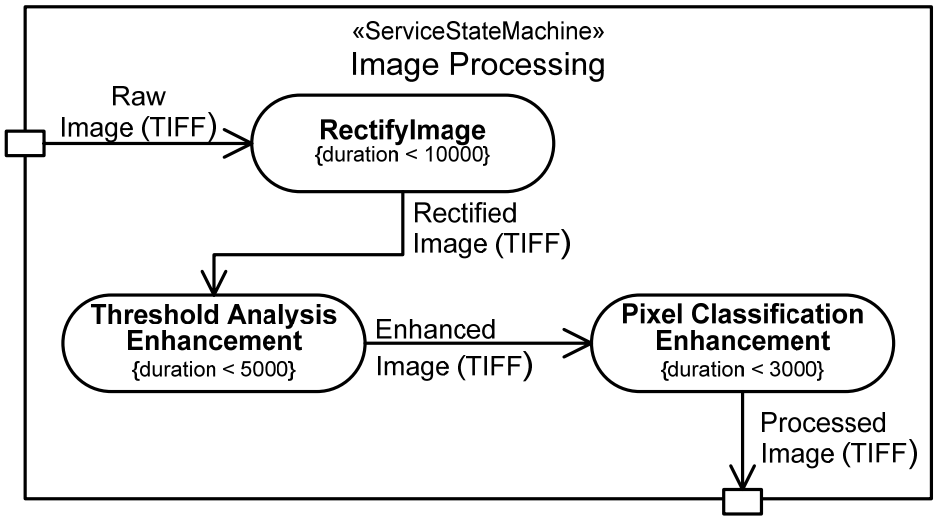
Detailed Product Description

SOV-4b products are usually UML (or similar) state transition models.



SOV-4b state transition model representation

An SOV-4b may also specify performance constraints (ie the maximum duration a service may be in a particular state):



SOV-4b showing performance constraints

SOV-4c - Service Interaction Specification

The SOV-4c specifies how a service interacts with external agents, and the sequence and dependencies of those interactions.

NAF V3 Equivalency

The equivalent NAF view is NSOV-5, Service Behaviour.

Background

The purpose of the SOV-4c is to specify the general sequence of interactions that are possible for a given service.

Usage

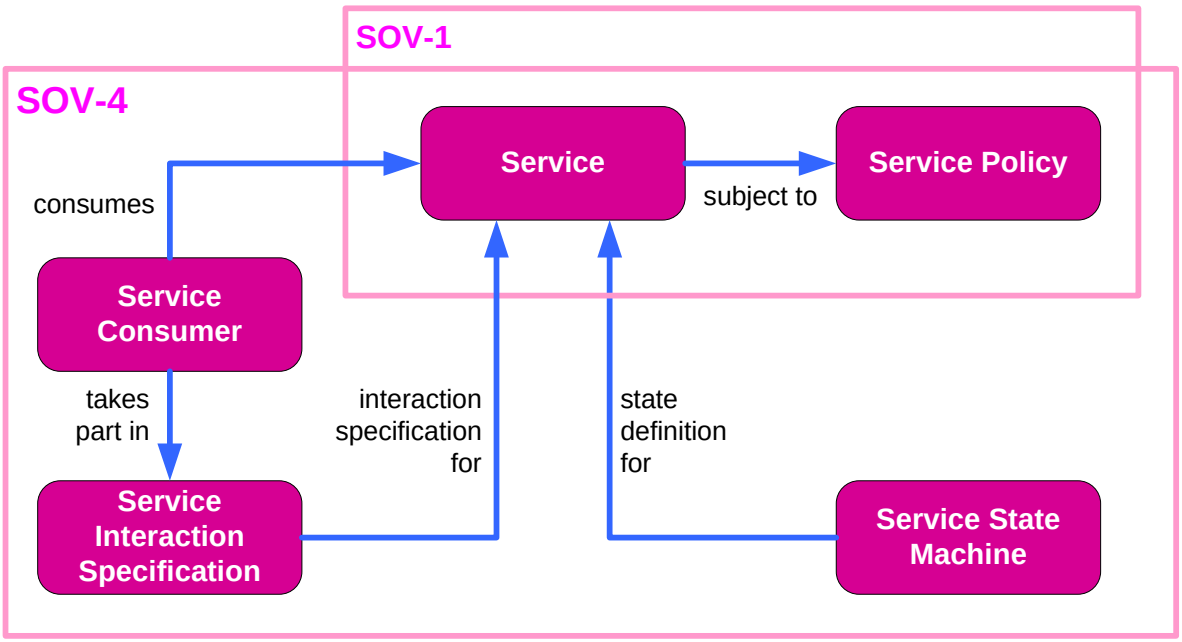
- Service specification.

Data objects

The data in an SOV-4c can include:

- Service.
- Service Interface.
- Service Lifeline.
- Service Consumer

.



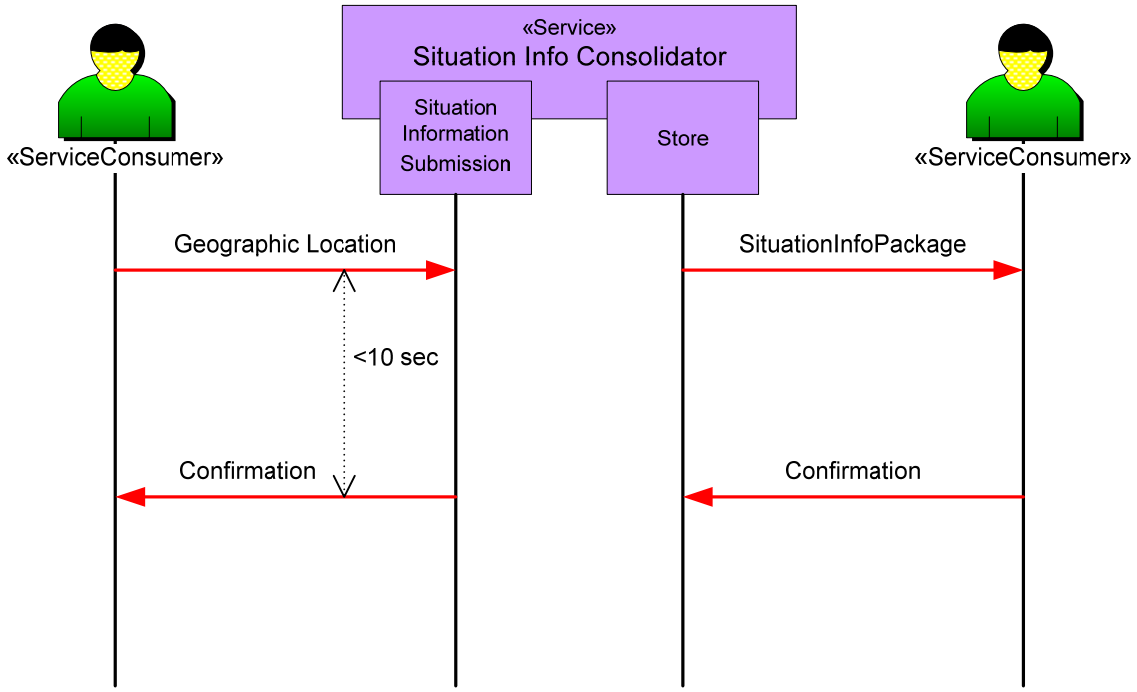
Relationships between Key Data Objects (Simplified from M3)

Representation

- UML.

Detailed Product Description

SOV-4c specifies how a service interacts with any given agent that has compatible interfaces. The representation is usually a UML Sequence Diagram.



SOV-4c Sequence Diagram representation

The product shows a service and the interfaces it exposes – quite often the diagrams can be cluttered if all interfaces are shown, so it is advisable to consider producing multiple SOV-4c products for a given service. Each interface in an SOV-4c has a “lifeline” to which messages are shown passing. It is also possible to show timing constraints between messages.

SOV-5 - Service Functionality

The SOV-5 defines the behaviour of a service in terms of the functions it is expected to perform.

NAF v3 Equivalency

The NAF v3 equivalent view is NSOV-5, Service Behaviour.

Background

SOV-5 is the key behavioural specification for services. Equivalent in nature to OV-5, Operational Activity Model, and SV-4, Functionality Description, it specifies a set of functions that a service implementation is expected to perform.

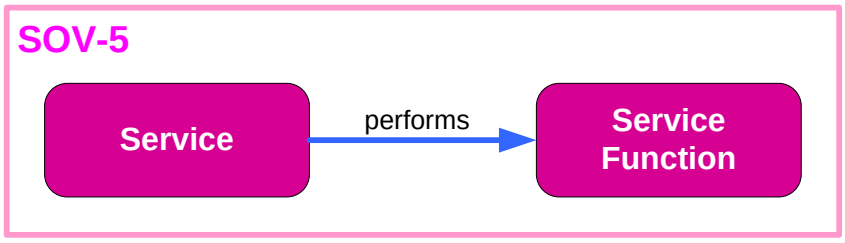
Usage

- Service specification.
- Functional requirements definition.

Data objects

The data in an SOV-5 can include:

- Service.
- Service Function.



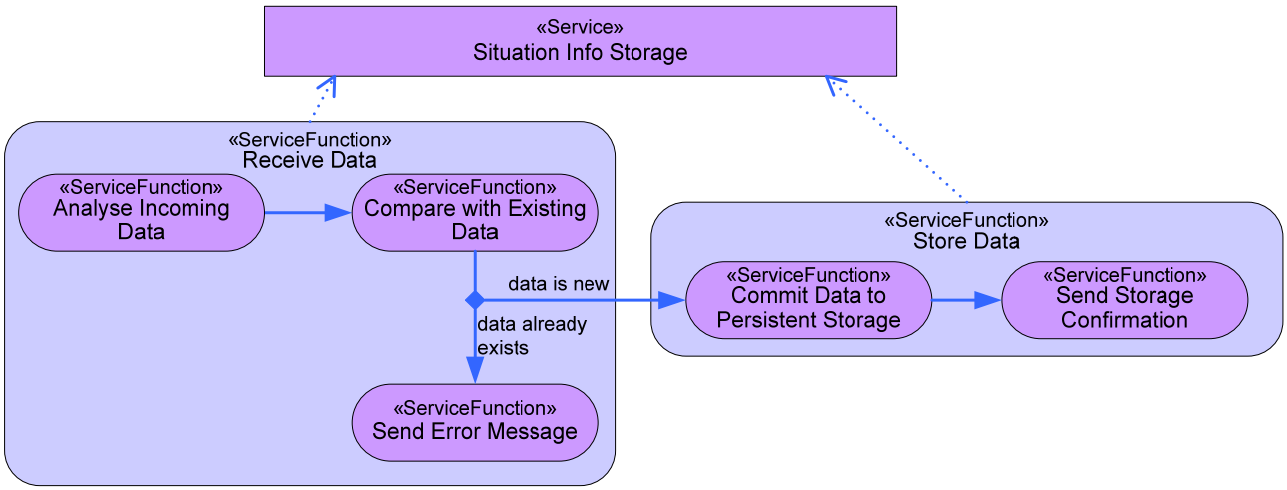
Relationships between Key Data Objects (Simplified from M3)

Representation

- Diagram.
- UML.

Detailed Product Description

An SOV-5 specifies the required functionality that an implementation of a service is expected to have; (the implementation of that behaviour is represented in SV-4, Functionality Description and SV-5, Function to Operational Activity / Service Function Traceability Matrix). An SOV-5 is usually presented as a functional diagram, with optional flows. An SOV-5 product should also show which service the functions correspond to.



Representation of SOV-5

Note that an SOV-5 should be a statement of what a service implementation is to do rather than how it is to do it. The functions specified in an SOV-5 should be essential to the service rather than an attempt to steer a particular implementation approach.