model futures
www.modelfutures.com

An Analysis of Services

Chris Partridge
Ian Bailey

Version 1.3 11 May 2010

Prepared for:

MINISTRY OF DEFENCE

Produced by Model Futures Ltd. under FATS/III

Model Futures Limited is a company registered in England and Wales with company number 05248454
Registered Company Address: 1 Nelson Street, Southend-On-Sea, Essex, SS1 1EG
VAT Number: 848 7357 75
MOD FATS/II: FATS/2/MFL
DGFM Supplier Code: 56945

Contents:

Abbreviations and Acronyms....3
Introduction....3
Approach Taken....3
Background....3
Findings....4
Conceptual Model....5
Informal Mapping to Zachman Framework....6
Analysis....7
Background....7
Analysis Method....7
Different definitions of Service....8
A Service's Provider and Consumers....11
What is a Service Use?....14
What happens when a service is consumed....16
Service Contract....18
Agreed Service Description....20
Service, Service Contract and Levels of Service Presence....21
Service Interaction or Interfaces....22
Service Interfaces and Messages....23
Service Categorisation - Taxonimisation - Classification....24
A common approach....24
Delivering a capability....24
A parallel capability taxonomy....24
A derived capability based service classification....25
A finer grained service classification....25
A service transaction or relationship – joint actions, plural subjects....26
Appendix A – End-to-end coverage....28
Appendix B – Gartner's SOA without Governance....29
Appendix C – Inputs....30
IDEAS Nations Meta-Models....30
Other Reference Models....30
Reference models briefly reviewed....30
Existing taxonomies....30
NEC Services Strategies....30
MoD - NEC....30
US DoD – NCW....30
Appendix D – Relationship between the SOA Models....31
Appendix E – OMG SoaML – Annex B....32
Relationship to OASIS Services Reference Model....32
Appendix F – Some Business Examples....37
OASIS- RM - 2.1.1 A worked Service Oriented Architecture example....37
SoaML - Example Participant Services Architecture....37

Abbreviations and Acronyms

Acronym	Term
OASIS	Organization for the Advancement of Structured Information Standards
OASIS-RM	OASIS Service Reference Model
OMG	Object Management Group
SOA	Service Oriented Architecture
SoaML	The OMG's SOA Modelling Language Standard
TOG	The Open Group
TOG-SO	The Open Group SOA Ontology
TOG-SSB	The Open Group SOA Source Book

Introduction

The goal of this report is to provide an in-depth common conceptual understanding of services end-to-end across the enterprise – one that encompasses business, IT and technical services and gives a picture of what, in essence, a service is.

Approach Taken

Getting a grip on the nature of services is very difficult. The word "service" is widely used, and has a number of different senses, some of which are only subtly different. Any attempt to define services from first principles is almost certainly going to lead to either a replication of one of the currently used senses of the word, or an altogether new sense of the word – neither of which is much use to the MOD.

The analysis documented in this report, therefore, is a forensic one. It begins with a survey of the major standards around service orientation that are of interest to MOD (OASIS, OMG, The Open Group, NATO and MODAF). The standards and specifications have been analysed to look for common meaning and purpose. From this initial analysis, the common aspects have been assembled into a conceptual model – a first stage towards developing a formal[1] ontology of services.

The key findings are summarised on page 4, and the analysis is documented in detail on page 7. The appendices that follow list background information from the various standards and specifications covered in the analysis.

Background

The MOD asked Model Futures to undertake the following:

> "Background: MOD has a Service Oriented approach for the delivery of capabilities, but there is no common understanding or definition of what a service is, or the types of services used in Defence.
>
> Task: The task is to conduct an analysis of services in Defence, to help provide a clear and unambiguous definition of what services are, how they are categorised, and how they relate to other services, systems and capabilities.
>
> The analysis should include an examination of service-related activities going on elsewhere (such as within NATO, the IDEAS Nations and the OMG), but the level of this examination should be proportionate to the amount of time available. This examination should inform, but not necessarily drive the solution, unless a best practice is identified."

[1] The term "ontology" has nearly as many senses as "service". In this case, we are referring to a formal model such as the IDEAS Foundation (www.ideasgroup.org) rather than the looser sense of ontology that is, for all intents and purposes, a conceptual model

Findings

The detailed analysis (detailed in next section – see page 7) of the current crop of leading standards for SOA (OASIS, The Open Group – TOG – and OMG) revealed three main issues:

- It confirms that "there is no common understanding or definition of what a service is". Though there is a broad similarity between the standards, there are significant and important differences in detail. For example, they have different notions of what a service is. In some cases (e.g. OASIS, OMG & DM2) the service is simply the mechanism for accessing a capability. In others (e.g. M3 & TOG) the service is the combination of the capability and the mechanism used to access it. It should also be noted that there are discrepancies and contradictory definitions in documents that originate from the same organisation (particularly the NATO Architecture Framework).
- There is not yet a clear conceptual picture of the underlying nature of what a service is – encompassing both its business and formal characteristics.
- Though there is a clear aspiration amongst the standards considered to provide an all-encompassing framework for services – spanning business and IT – there is still work to be done to achieve this (see Appendix A). There has been significant work on the formal and technical IT aspects, but these are not yet fully integrated with the business view.

In addition, the analysis (see page 7) suggests that the categorisation of services could be improved by:

- Replacing service taxonomies which indirectly classify capability with capability taxonomies.
- Considering where it is more appropriate to classify the service components than the overall service.
- Considering whether the service delivery (and what aspects of it) should be joint actions – i.e. is the service delivered in partnership with the consumer or simply as a contracted service with a clear delineation between provider and consumer

These reflect the early stage of maturity of work in this area (which is not surprising given the first standard was only published in 2006). The issue of whether the service is simply an interface to some capability, or the combination of the capability and its interface is significant – especially as MODAF is in the minority with its assumption of the latter.

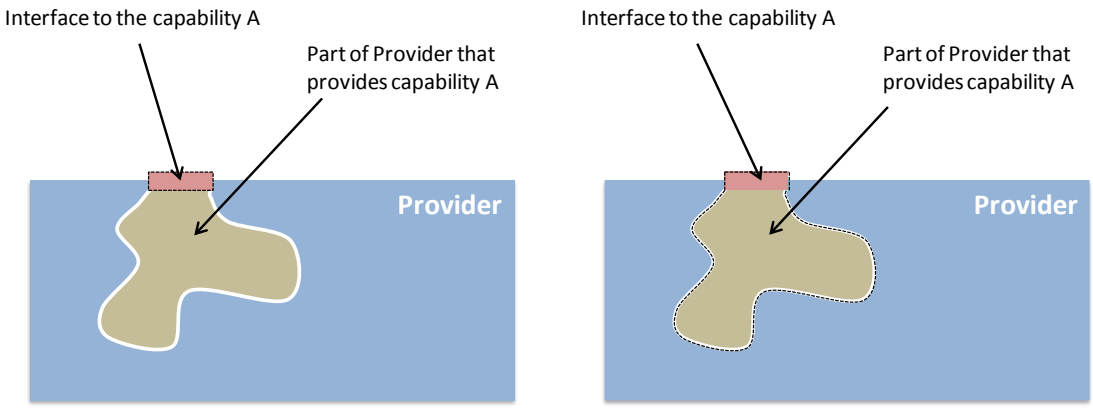


Figure 1 Illustration of the two main understandings of the extent of a service (shown by dotted line)

The potential for misunderstandings that result from this issue are significant – especially from a contractual point of view. In one case we are contracting for simply a method of access to capability, in the other we are contracting for the capability to be delivered *with* the method of access.

A clear conclusion from this analysis is that a more consistent approach to service categorisation (see page 24) can result in more coherent service governance, and even a way to measure the maturity of an enterprise's service-orientation. In the context of defence, this also offers a possibility for measuring the level (or extent) of "net-centricity" or "network-enablement" of a defence force.

The analysis provides a business-driven ontological picture of service's underlying nature that spans the business-IT divide and resolves the identified differences. The goal of the analysis has not been to produce 'Yet Another Reference Model' (YARM), but instead to provide a common understanding for a common reference model. However, as none of the existing reference models has a top ontology[2], we used the BORO-based IDEAS model as the foundation for the ontological analysis.

Conceptual Model

The ontological picture has been summarised into the conceptual model below.

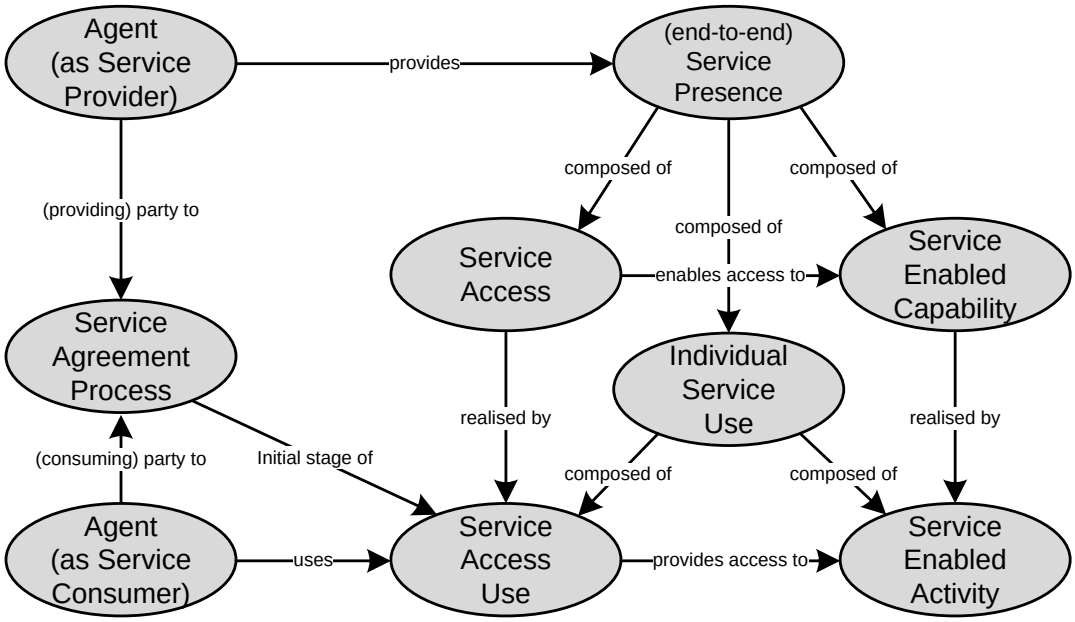


Figure 2- Conceptual model of service

The definitions for each of these elements can be found in the analysis section, beginning on page 7.

Key aspects of the conceptual model are:

- It frames the Service elements in terms of their business nature – so that the business drives the structure.
- It includes the OASIS notion of considering Service as necessarily having a way of accessing a capability rather than the accessed capability. To make this clear, this is named 'Service Access' in the conceptual model. This can be used as a tool to identify simplistic re-badging of processes as services.
- It clearly distinguishes the business elements of service.

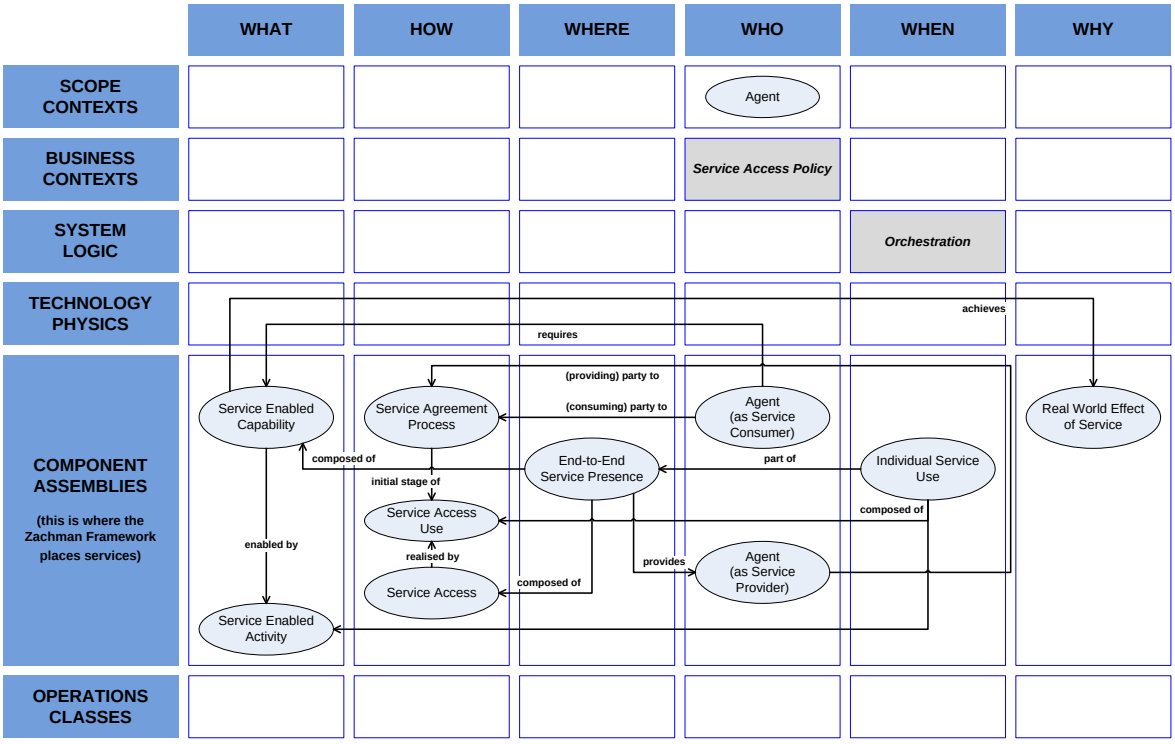
In addition, the IDEAS Nations' MODAF M3 and DoDAF DM2 metamodels were reviewed in the context of the conceptual model. The main recommendations are:

[2] DM2 uses elements of the IDEAS top ontology, but its Services Model is currently at too early a stage of maturity to be a basis for the analysis.

- The common ontological model is used to harmonise the notion of service across the two metamodels. Currently the two models have very different pictures of services (also different from the OASIS, The Open Group and OMG reference models) and like these, do not (yet) have a clear conceptual picture of services. Clarifying the picture should harmonise the two IDEAS metamodels.
- SOA Governance is included in the metamodels. Currently there is no SOA Governance in the metamodels. As Gartner has noted, SOA without governance descends into Wild West SOA (see Appendix B). We suggest that the work starts with Service Agreement elements, which are the (business) basis for SOA Governance.

Informal Mapping to Zachman Framework

The Zachman International training material asserts that services exist at the "Component" level. If one accepts that this is correct, then the most likely mapping of the concept model to the Zachman Framework will looks something like:



It is the authors' belief that this is too simplistic a mapping to be useful, however. The Zachman Framework is context-sensitive – the cells in which different concepts end up depend to a large extent on what the framework is being used for. If the whole enterprise is to be analysed, there may well be a case for suggesting that the services live only at the Component level. However, if the service itself is the subject of the analysis, then it is reasonable to expect the architect to populate many more cells than just the Component cells. The MODAF Service-Oriented Views cover service taxonomy, interfaces, behaviour, interactions, etc. To cover these in Zachman would mean making extensive use of the scope and business context rows (esp. for MODAF SOV-3 which maps service so the capabilities they deliver).

Other aspects of service management such as the orchestration and policy are probably more appropriately placed at higher levels in the Zachman Framework.

Analysis

Initial analysis of inputs (listed in Appendix C) identified a core of emerging standards that provide a reference framework for SOA. These were used as the main inputs for the detailed analysis – they are:

OASIS	SOA-RM 1.0	OASIS-RM	12 October 2006
The Open Group	SOA Ontology	TOG-SO	14 July 2008
OMG	SoaML 1.8	SoaML	25 August 2008
The Open Group	SOA Source Book	TOG-SSB	29 April 2009
OASIS	SOA-RAF 1.0	OASIS-RAF	14 October 2009

Appendix D describes the ways in which these have influenced each other.

Annex B of OMG's SoaML (copied in Appendix E) collates the main definitions used in the first three standards. This provides a useful benchmark for where the understanding of SOA is (and how much of a common understanding has been achieved – there is still a way to go).

Other inputs are listed in Appendix C – Inputs.

None of the standards has yet attempted an ontological analysis or worked with a top ontology. Our analysis has done this for the key elements of the standards.

Background

From a business perspective, the notion of service as the 'performance of any duties or work for another' is well-established and supported by economic and legal notions as well as common sense – where service is 'an act of helpful activity' as in 'to do someone a service'.

The emergence of IT services (particularly web services) has brought the need for (and focus on) a sufficiently formal structure for services to enable them to be automated. The increase in scale of IT services has driven a need for an architectural approach. This is a key driver for the SOA standards. It is also the root for one of its challenges – how to interpret the formal structures in business terms. And, in particular, what the parallels for these structures are in non-IT business services.

There is also a keen appreciation that an SOA approach can deliver two major categories of business value:

- Sharing (also called leverage and reuse)
- Agility (ability to change more rapidly)

The aspiration is that this approach can be applied broadly across the enterprise – to business as well as IT systems.

Analysis Method

The goal of the analysis was to build a common understanding of services on the foundation of the standards. This proceeded through a detailed ontological analysis of their elements, comparing the results for similarities and differences and synthesising a common picture. In addition, where necessary a business perspective was added. The goal of the synthesis was to move from a number of narrower perspectives to a broader, more rounded perspective – to move from (a) to (c) as shown diagrammatically in the figure below.

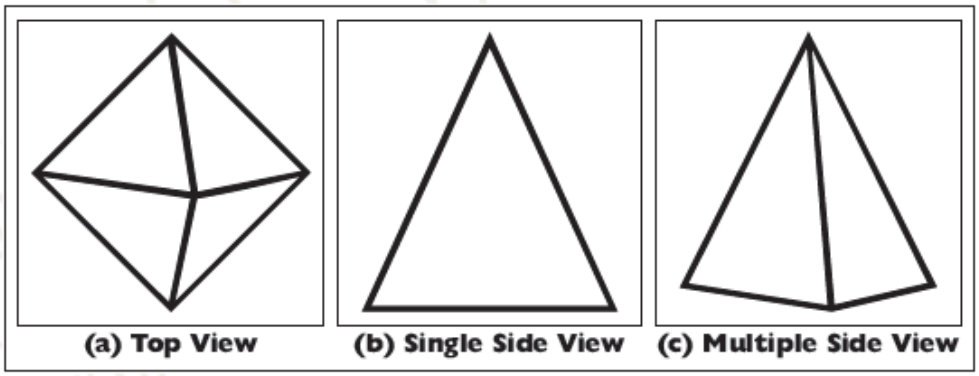


Figure 3 Synthesising different perspectives of a pyramid

Different definitions of Service

The core notion is Service, and the definitions are copied below.

OASIS RM[3]	*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.*
TOG SO	*A logical representation of a repeatable business activity that has a specified outcome (e.g., check customer credit; provide weather data, consolidate drilling reports). It is self-contained, may be composed of other services, and is a "black box" to its consumers.*
SoaML	*Service is defined as a resource that enables access to one or more capabilities. Here, the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description. … A service is provided by an entity - called the provider - for use by others. The eventual consumers of the service may not be known to the service provider and may demonstrate uses of the service beyond the scope originally conceived by the provider.* *Identifies or specifies a cohesive set of functions or capabilities that a service provides.*

TOG-SSB also includes a second (different) definition: "A service is a repeatable activity that has a specified outcome"[4]. This effectively moves the definition down a level of representation.

NATO's NAF uses the same M3 metamodel as MODAF, and so, at that level, the same definition. However, in its associated documentation, the initial description (1.10.5) of a service as a unit of work bears some similarity to TOG's 'repeatable business activity'. Their subsequent definition (3.3.13), like TOG's is at a higher level of representation and seems to focus on the deliverables (what OASIS et. al. call 'real world effect').

NATO NAF	*1.10.5 NATO Service-Oriented View*	*A service, within the NSOV, is understood in its broadest sense, as a unit of work through which a provider provides a useful result to a consumer.*

[3] The OASIS-RM definition is used in DM2 and noted in M3.

[4] http://www.opengroup.org/projects/soa-book/page.tpl?CALLER=page.tpl&ggid=1317. Compare this with http://www.opengroup.org/projects/soa-book/page.tpl?CALLER=page.tpl&ggid=1314.

NATO NAF	*3.3.13 Service*	*Definition* *A service is implementation independent specification of the deliverables that has added value to the consumer of that service in accordance with the consumer's goals and objectives.* *An operational service is a service providing an observable outcome which fulfils an operational need.* *An information service is a service providing data which fulfils information requirements.* *An application service is a service delivering automated functionality which fulfils the needs and requirements of the user, provided by an automated application.*

OASIS-RAF has no direct definition of a Service.

The core of the OASIS-RM and SoaML definitions are broadly similar. These talk about a Service as a mechanism or resource that enables access to a capability, where this is (according to SoaML) "The ability to act and produce an outcome that achieves a result". This would seem to clearly block any identification of the service with the capability or even an overlap between them. One can picture the mechanism or resource as an access point to the capability – see the figure below. (Later we look at how one can have an access point to capability - which is an ability.)

What is odd in SoaML's case, is that it appears to have two layers of access. There is a ServicePoint (see below) that provides access to the service and then the Service that provides access to the Capability – see the figure below. It is not clear what the motivation for this is (apart from conforming to the UML MetaModel). Until a good motivation is provided, it is proposed that this second layer is disregarded.[5]

SoaML	*7.3.11 ServicePoint*	*A ServicePoint is the offer of a service by one participant to others using well defined terms, conditions and interfaces. A ServicePoint defines the connection point through which a Participant offers its capabilities and provides a service to clients.* *Description* *A ServicePoint is a mechanism by which a provider Participant makes available services that meet the needs of consumer requests as defined by ServiceInterfaces, Interfaces, and ServiceContracts. A ServicePoint is represented by a UML Port on a Participant stereotyped as a "ServicePoint."*

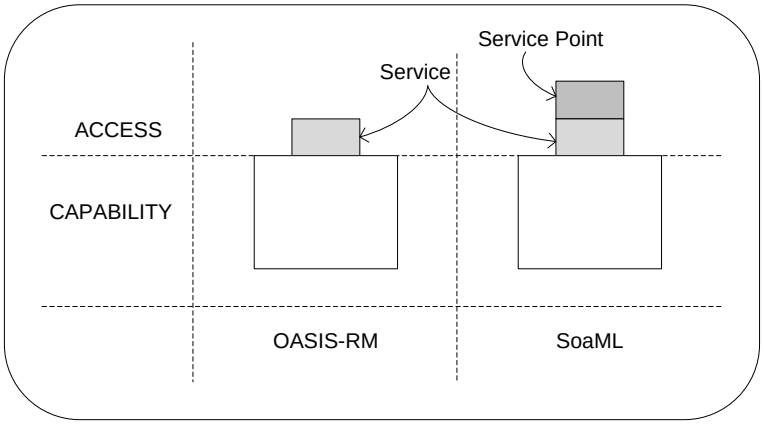


Figure 4 Contrasting the OASIS-RM and SoaML views of Service

[5] Both DM2 and M3 have the equivalent of SoaML ServicePoints.

There is a clear difference between OASIS-RM and SoaML definitions, and TOG-SO definition. The TOG-SO refers to a 'logical representation' whereas OASIS-RM and SoaML refer to, variously, a mechanism and a resource[6]. The TOG-SO definition is clearly at a higher level of representation than the OASIS-RM and SoaML definitions.

The TOG-SO "repeatable business activity" is not the same thing as the OASIS-RM and SoaML's mechanism and resource. But it is similar to the manifestation of the capabilities that the mechanism and resource enable.

The core distinction that divides the two sets of definitions is between the dispositional and manifestation[7] aspects of Service. OASIS and SoaML see the service as the dispositional ability to deliver the effect, whereas TOG focuses on the process that delivers the effect. Consider a taxi service. The dispositional view sees a key feature of the taxi service is its ability to provide taxis. Under this view, the service exists whether or not any taxis are actually in use, provided the ability exists. The manifestation view sees the service as the process of providing a taxi ride. This however does not exist when there are no taxi rides being provided. So depending on which view one uses, one could give different answers to the question whether there is a taxi service.

Another concern is that none of the definitions match up well with the ordinary language sense of service[8]. The TOG-SO definition is at a different level of representation – in other words, it is the specification or description of a service rather than the service itself. If we revise it down a level we can make a connection to the ordinary language sense. However the OASIS-RM and SoaML definitions seem to be referring to something quite different. The man on the Clapham Omnibus, when talking of a taxi service would regard the provision of the taxi (the "repeatable business activity" – but not its representation) as the service. He would find a sense that excluded this and only focussed on the related "enabling access" – which in this case might be a telephone call booking the taxi – as unusual. OASIS-RM explain their motivation for choosing this unorthodox sense as follows:

> *"The service concept above emphasizes a distinction between a capability that represents some functionality created to address a need and the point of access where that capability is brought to bear in the context of SOA. It is assumed that capabilities exist outside of SOA. In actual use, maintaining this distinction may not be critical (i.e. the service may be talked about in terms of being the capability) but the separation is pertinent in terms of a clear expression of the nature of SOA and the value it provides."*

The emphasis on this distinction can help to make clear the need for a Service Access, and avoid situations where processes are re-badged services without there being any Service Accesses.

So one challenge is clarifying what the extent of the Service should be – the access point, what is accessed or both (as shown in the Figure 5)? Or, more relevantly; which of these extents is useful – and which are not?

---

[6] OASIS-RM does not define a 'mechanism' and SoaML does not define a 'resource'. For our purposes, I propose to consider them as similar or effectively the same.

[7] These are key general terms in IDEAS.

[8] "an act of helpful activity; help; aid: to do someone a service" Random House Dictionary.

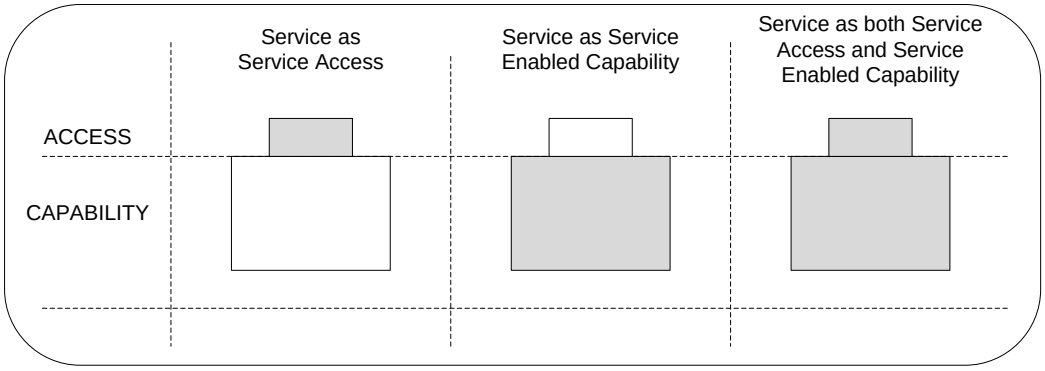


Figure 5 Options for the extent of Service

From the perspective of separation of concerns, it makes sense to distinguish between the Service Access and the Service Enabled Capability. Where a Service Access taxonomy will classify the types of access and the Service-Enabled Capability Taxonomy will extend (and reuse) the Capability Taxonomy. However, as will become clear as later in the report, it is also useful to be able to refer to both of them. To avoid the use of the loaded unqualified term 'Service', we qualify this as 'Service Access' and introduce the terms 'End-to-End Service Presence' and 'Service Enabled Capability' – as shown below.

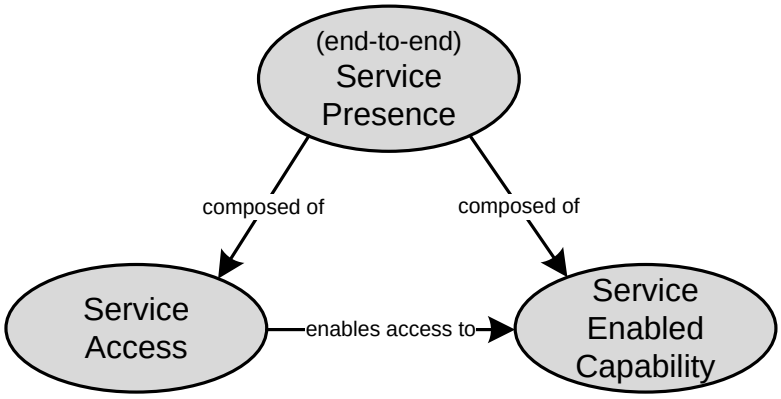


Figure 6 The whole-part (mereological) anatomy of a service

A further issue is that both the OASIS-RM and SoaML definitions also include significant constraints upon the mechanism or resource. One can see the influence of IT services here. However, it is unlikely that all (non-IT) business services (such as a taxi service) will be regimented to the extent that they have a clearly prescribed interface or a service description that specifies many, if any, constraints and policies. This suggests that this is a description of an ideal situation even though it is phrased as a necessary condition.

A Service's Provider and Consumers

The standards take a broadly similar view of participants providing or consuming services or both – rather than reifying the notion of provider and consumer. The notion of a participant is broad – as SoaML notes; "In the business domain a participant may be a person, organization, or system. In the systems domain a participant may be a system, application, or component."

OASIS RM	*3.1 Service*	*A service is provided by an entity – the service provider – for use by others, but the eventual consumers of the service may not be known to the service provider and may demonstrate uses of the service beyond the scope originally conceived by the provider. …* *A service is opaque in that its implementation is typically hidden from the service consumer except for (1) the information and behavior models exposed through the service interface and (2) the information required by service consumers to determine whether a given service is appropriate for their needs. …*
SoaML	*7.3.7 Participant*	*A participant is the type of a provider and/or consumer of services. In the business domain a participant may be a person, organization, or system. In the systems domain a participant may be a system, application, or component.* *Description* *A Participant represents some (possibly concrete) party or component that provides and/or consumes services - participants may represent people, organizations, or systems that provide and/or use services. A Participant is a service provider if it offers a service. A Participant is a service consumer if it uses a service - a participant may provide or consume any number of services. Service consumer and provider are roles Participants play: the role of providers in some services and consumers in others, depending on the capabilities they provide and the needs they have to carry out their capabilities. Since most consumers and providers have both services and requests, Participant is used to model both.*
TOG SSB	*The Building Blocks of SOA*	*A service has a provider, can have one or more consumers, and produces effects that are of value to its consumers.* *Providers and consumers see services from different points of view. To a consumer, a service is a black box. Two services are the same to a consumer if, given the same inputs, they produce the same effects. To a provider, a service is a means of exposing capabilities. Two services are different to a provider if they have different mechanisms for doing this, even though they produce the same effects. Architects talk to providers and to consumers, and must be able to see services from both points of view.*

OASIS RAF	*3.3.1 Service Providers and Consumers*	*Service Provider* *A service provider is a participant that offers a service that enables some capability to be used by other participants.* *Note that several kinds of stakeholders may be involved in provisioning a service. These include but are not limited to the provider of the capability, an enabler that exposes it as a service, a mediator that translates and/or manages the relationship between service consumers and the service, a host that offers support for the service, a government that permits the service and/or collects taxes based on service interactions.* *Service Consumer* *A service consumer is a participant that interacts with a service in order to realize the real world effect produced by a capability to address a consumer need.* *It is a common understanding that service consumers typically initiate service interactions. Again, this is not necessarily true in all situations (for example, in publish-and-subscribe scenarios, a service consumer may initiate an initial subscription, but thereafter, the interactions are initiated by publishers). As with service providers, several stakeholders may be involved in a service interaction supporting the consumer.* *Service providers and service consumers do not represent truly symmetric roles: each participant has different objectives and often has different capabilities. However, the objectives and the conditions under which those objectives align are critical for a successful interaction to proceed.*

The service provider-consumer relationship is not symmetric – as TOG-SSB says; "A service has a provider, can have one or more consumers". This supplies part of what individuates a Service – as shown in the figure below, a Service has only one provider (though that provider may change over time).

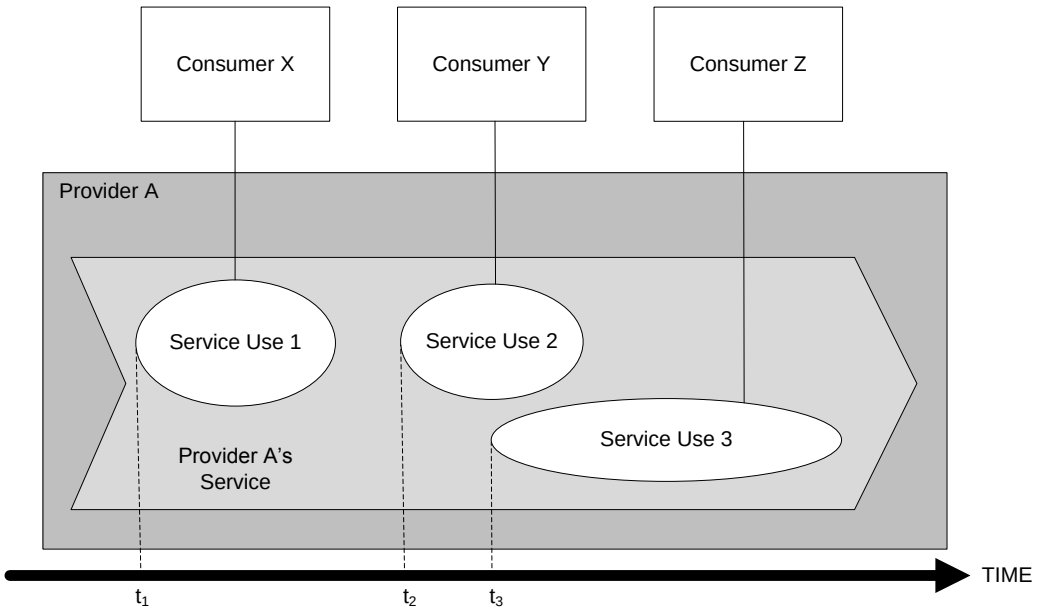


Figure 7 A service has one provider, can have one or more consumers

So, for example, the Acme Taxi Company provides a taxi service to a number of its customers. The Zenith Taxi Company also provides a taxi service to its customers. These are different individual services, but the same type of service – a taxi service.

What is a Service Use?

Taking the taxi service example from before, one can look at it from the taxi company's point of view, and say the company provides a taxi service (that is, there is one taxi service) – this is the sense of service used in the previous section. When someone uses the taxi service, they can say "the company provided me with a service" (that is, they provided me with one taxi service). If I use the taxi service three times, they have provided me with the service three times. We will call this sense *Service Use*.

However, from the perspective of the OASIS-RM and SoaML definitions the term 'Service Use' has a different meaning. As the mechanism for accessing the capability is the service then using this access mechanism is using the service – what they term the 'Service Interaction'. What this report calls 'Service Use' they would call 'Capability Use'.

OASIS RAF	*4.3 Interacting with Services Model*	Interaction is the activity involved in using a service to access capability in order to achieve a particular desired real world effect, where real world effect is the actual result of using a service.

The whole point of a service is that it can be used. It is only in the latest standard (OASIS-RAF) that there is an attempt to model this, and then indirectly. In the earlier standards (SoaML and OASIS-RM) there is some discussion about providing and invoking a service, but there seems to be no model element that corresponds to a use of a service.

SoaML	*7.1.3 Key Concepts of Basic Services*	A service is provided by a participant acting as the provider of the service - for use by others.
OASIS RM	*2.1.1 A worked Service Oriented Architecture example*	… a consumer accesses electricity generated (the output of invoking the service)

OASIS-RAF introduces the notion of an Action and invoking actions against a service.

OASIS RAF	*3.1.2.1 Action and Actors*	Action An action is the application of intent to achieve an effect (within the SOA ecosystem). This concept is simultaneously one of the fulcrums of the Service Oriented Architecture and a touch point for many other aspects of the architecture: such as policies, service descriptions, management, security and so on. The aspect of action that distinguishes it from mere force or accident is that someone or something intended the action to occur.
OASIS RAF	*4.1.2 Use Of Service Description*	
OASIS RAF	*4.1.2.1.1 Description and Invoking Actions Against a Service*	The action model identifies the multiple actions a user can perform against a service and the user would perform these in the context of the process model as specified or referenced under the Service Interface portion of Service Description.

It provides a model for this in its Figure 33, reproduced below.

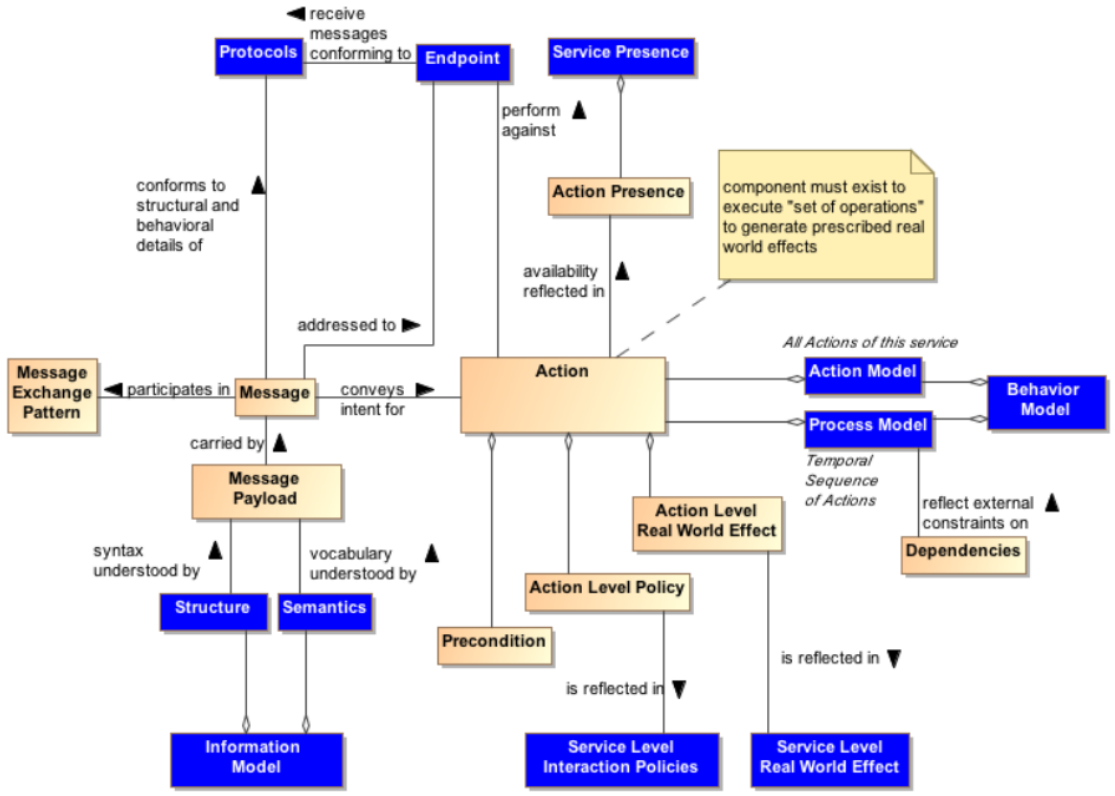


Figure 33 Relationship Between Action and Service Description Components

This gives us an indirect model for Service Use. It involves an action that has its availability reflected in an Action Presence that is performed against a Service Presence. There are various comments that help to clarify what the authors intended – copied below.

Presence for an action means someone can initiate it and is independent of whether the preconditions are satisfied.
From the above discussion of model elements of description we may conclude … presence of service is some aggregation of presence of its actions.
Protocol - A protocol is a structured means by which service interaction is regulated.
Presence - Presence is the measurement of reachability of a service at a particular point in time.
A protocol defines a structured method of communication with a service. Presence is determined by interaction through a communication protocol.
Presence of a service is an aggregation of the presence of the service's actions, and the service level may aggregate to some degraded or restricted presence if some action presence is not confirmed.

This provides us with a technical characterisation of Service Presence. When we examine Service Contracts, we will be able to give this a business characterisation.

In the OASIS-RAF model, there is a clear distinction between the message and the action (in Figure 33, they are two different boxes). This parallels the OASIS-RM and SoaML distinction, noted earlier, between the mechanism or resource that enabled access and the capability it enabled access to.

The OASIS-RAF model illustrates another challenge. From the model, one cannot work out whether a Service Use can correspond to a single or many Actions. None of the standards gives guidance on this, making clear the boundaries of Service Use. An example may help make the challenge clear. Take the OASIS-RM electric utility worked example (Appendix F). If a consumer accesses electricity generated via two

different wall outlets for two different devices at different times – is this two uses or one? It would seem to make sense to see these two 'accesses' as part of one continuing use under a single contract – as (one assumes) the contract does not divide the 'accesses' into different uses. Hence it is possible that a series of actions (which might not qualify as an action) would be a single service use. We can clarify this when we have introduced the notion of Service Contract.

What happens when a service is consumed

It is generally recognised that what underlies services from a business and conceptual level is delegation, and that delegation has been an important part of the ways humans manage, probably since they started managing.

OASIS RM	2.2 How is Service Oriented Architecture different?	SOA provides a more viable basis for large scale systems because it is a better fit to the way human activity itself is managed – by delegation.
OASIS RAF	3.1.1 Actors, Delegates and Participants	Delegate - A delegate is an actor that is acting on behalf of a participant.
OASIS RAF	3.3.4 Ownership	One who owns a resource may delegate rights and responsibilities to others, but typically retains some responsibility to see that the delegated responsibilities are met.

See also Appendix F's SoaML - Example Participant Services Architecture.

Delegation, by its nature, creates characteristic mereological (whole-part) structures that are a tell-tale sign that there is a service. When an agent (an entity capable of action) is given responsibility for a task, it can decide to delegate some part of the task to another agent. The first agent has overall responsibility[9] for the overall task. However, it only has direct responsibility for the parts of the task that are not delegated. The second agent acquires direct responsibility for the sub-task that is delegated to it.

Where an agent has direct responsibility for a task, it undertakes the whole task – the overall process – so there is no distinction between what it owns and what it is directly responsible for and what it undertakes – as shown below.

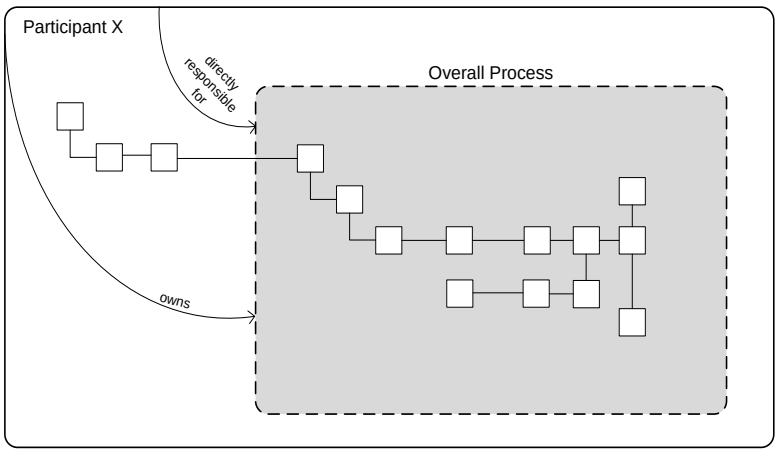


Figure 8 Ownership and direct responsibility

[9] For simplicity this has been phrased in terms of responsibilities, but it involves both rights and responsibilities.

However, when an agent delegates a task, it creates a distinction between what it owns and what it is directly responsible for (which is what it undertakes). There is a part of what it owns – the overall process – that is now part of another agent – as shown below. (Appendix F's 'SoaML - Example Participant Services Architecture' has another example of delegation – but this does not focus on the whole-part (mereological) structure.)

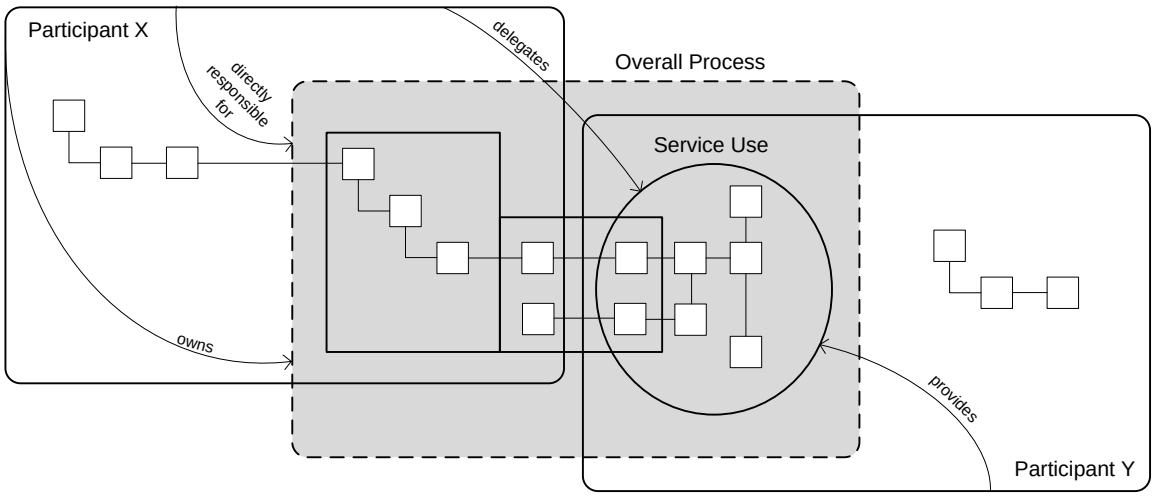


Figure 9 ownership and indirect responsibility

In business services, identifying the participant correctly can be key, as they retain responsibility. There is a telling example from early English history. The Monarch was the Commander-in-Chief of the armed forces and responsible for declaring war. There was a concern that if the Monarch was killed as the person responsible for declaring war was dead, the war was technically over. So the Monarch was regarded as a legal entity[10], and upon the death of the person occupying the Monarch position, his or her successor became the Monarch. It was the Monarch not the person that was responsible for declaring war, so the declaration was unaffected.

In the case of agents that by themselves cannot take responsibility directly (such as computer systems and their parts) it might appear that responsibilities are not relevant. However, in this case the responsibility is delegated by others and held indirectly. Ensuring that there are clear lines of responsibility is one of the key tasks of SOA Governance.

This analysis enables us to expand the conceptual model, showing how the consuming agent interacts with the providing agent through a Service Use. Note that the whole-part (mereological) access-enabled capability distinction is inherited from Service Presence to Service Use.

[10] Technically a Corporation Sole called the Crown.

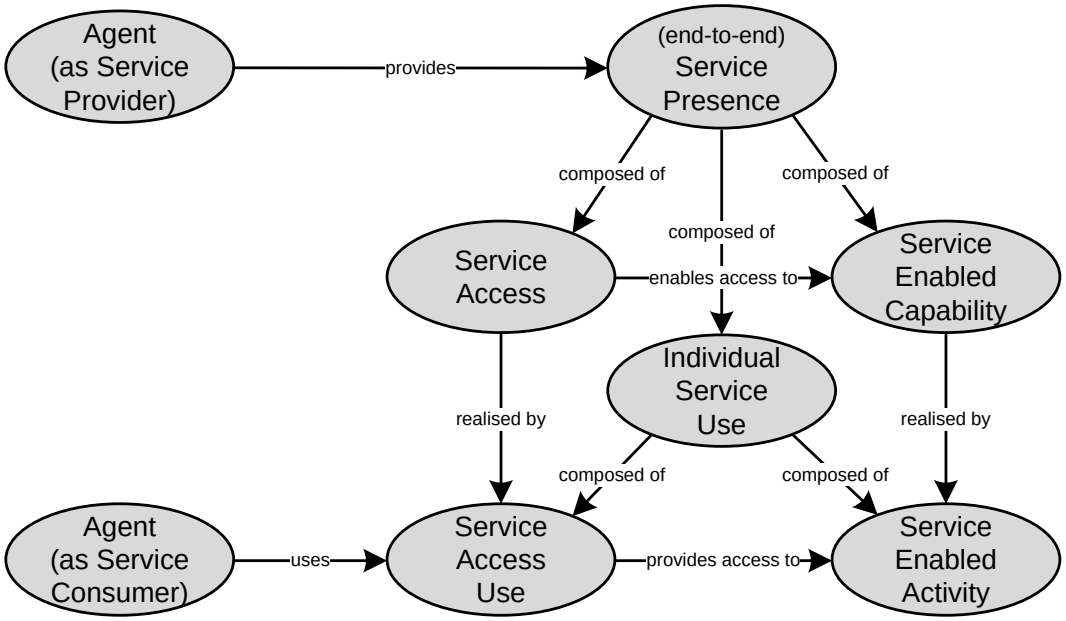


Figure 10 How the consuming agent interacts with the providing agent.

Service Contract

An essential component of a service is where a participant makes an agreement with another participant to provide a service. When a taxi service is called and a booking made, the taxi service is agreeing with customer that it will provide a taxi – and the customer is agreeing that he/she will use it. In the three standards this pattern is capture using Service Contract - definitions copied below.

OASIS RM	3.3.2.2 Service Contract	*Whereas a policy is associated with the point of view of individual participants, a contract represents an agreement between two or more participants. … a service contract is a measurable assertion that governs the requirements and expectations of two or more parties. … Since a contract is inherently the result of agreement by the parties involved, there is a process associated with the agreement action. Even in the case of an implicitly agreed upon contract, there is logically an agreement action associated with the contract, even if there is no overt action of agreement.*
SoaML	7.3.13 ServiceContract	*A ServiceContract is the formalization of a binding exchange of information, goods, or obligations between parties defining a service.* *Description* *A ServiceContract is the specification of the agreement between providers and consumers of a service as to what information, products, assets, value, and obligations will flow between the providers and consumers of that service. It specifies the service without regard for realization, capabilities, or implementation. A ServiceContract does not require the specification of who, how, or why any party will fulfill their obligations under that ServiceContract, thus providing for the loose coupling of the SOA paradigm.*

TOG SSB	The Building Blocks of SOA	*A contract is an agreement between two or more actors - the parties to the contract. The term is most commonly used for a written agreement, or one that is enforceable at law, but it can be applied more widely.* *A service contract is a contract between the provider of a service and one or more of its consumers.* *A service contract may be an implicit agreement that the service will conform to its description, or it may be a more formal agreement, which could be recorded in a signed internal enterprise document, or be a legal contract executed between enterprises.* *A service contract covers functionality (what effects the service produces), and often also covers service quality.*

In OASIS-RAF, "A contract is a constraint that has the agreement of the constrained participants" ('constraint' is not defined).

In ordinary language the term 'contract' has two different senses, one referring to the act of agreeing a contract and the other to the document that is used to record what has been agreed. The OASIS-RM and TOG-SSB broadly go with the first sense. SoaML goes with a variant of the second sense. The OASIS-RM and TOG-SSB definitions treat the agreement between the parties for the service as the contract (though we need to clarify the details). They also recognise that the contract needs to 'govern' or 'cover' the requirements and expectations of the parties.

The SoaML definition regards the contract as a specification (representation) of what the Service will be contracted to do – independent of the parties and any agreements they might have. (In IT – and perhaps business – terms, one can regard this as a pro-forma contract document that can be used when parties are agreeing the service.) From a practical point of view, this would seem to exclude any contract variations – any change to the contract and it becomes a contract for a different service. One can guess that OMG were motivated by the thought that standardising in advance on a particular form of contract can bring economic benefits; this does not make it a necessary feature of a service – a requirement for a definition.

In addition, the SoaML definition would also seem to allow for a contract that spans several providers as well as consumers (and in this respect has similarities with the way computer languages define the interfaces used in building IT services). This can be read as implying that an individual Service can span several providers. As noted earlier, the provider-consumer relationship is not symmetric in this way.

We can resolve these differences with this finer characterisation. Before the Service's capability is used, there is an agreement between the provider of the service and its consumer. The process of agreement will include agreeing what the service will provide to the consumer. In a traditional business situation the process of negotiation will produce a document that describes (represents) what has been agreed. At the end of the negotiation, if it is successful, there will be agreement. This gives us:

- The Service Agreement Process.
- The Service Agreement – a key point in (and so part of) a successful agreement process.
- The Agreed Service Description – a description or representation of what has been agreed.

These are shown diagrammatically below.

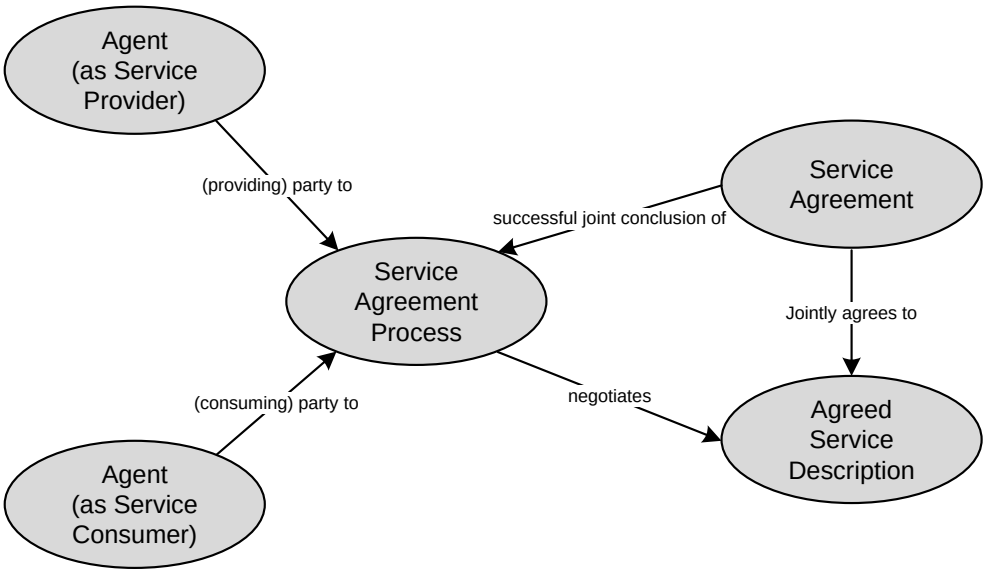


Figure 11 Service Agreement structure

In addition, one can develop pro-forma documents, copies of which can be used as Agreed Service Descriptions in the Service Agreement Process.

Agreed Service Description

There are a variety of requirements for descriptions at different levels. From an IT perspective there is a need to sufficiently formally describe the IT processes for them to be automatable. From a business perspective, there is a need to adequately describe the agreed service to provide sufficient assurance to the consumer and provider. From the business consumer perspective, there is a requirement that the Agreed Service Description adequately describes the needed real world effect. From the business provider's perspective there is a requirement to adequately describe the workings of the service so that it will produce the agreed results – the real world effect – and any other interactions that form part of the agreement. These can overlap and interlock in quite complex ways.

From a business perspective, Service Access Use process divides into two parts with quite different business concerns;

- The Service Agreement Process and
- The Service Execution Interaction Process.

However, from a technical perspective this can (and is) looked at and defined as a single process of interaction between provider and client (there is often no formal IT feature to differentiate agreeing from execution interaction).

From a business perspective, the Service Agreement Process should involve, either directly or indirectly, the joint acceptance of the Agreed Service Description. In so far as it involves it directly, the details of the description are agreed during the process – and so part of the process. In so far as it involves it indirectly, the details of the description are referred to by the process and so not part of the process.

The Agreed Service Description typically crystallises at the time of the Service Agreement. So, for example, one can negotiate a reduced rate for a taxi service immediately before the agreement is accepted. In an automated IT environment one may wish to avoid this level of flexibility and have a policy of setting the Agreed Service Description in advance.

Often a service will have a clearly defined goal – in OASIS RM terms, a "real world effect" – and this is typically described in the Agreed Service Description.

OASIS RM	3.2.3 Real World Effect	There is always a particular purpose associated with interacting with a service. … At first sight, such a goal can often be expressed as "trying to get the service to do something". This is sometimes known as the "real world effect" of using a service. For example, an airline reservation service can be used to learn about available flights, seating and ultimately to book travel – the desired real world effect being information and a seat on the right flight.

As the OASIS example above shows, in a pure IT service, the effect is typically the Service Provider giving the Service Consumer some information – and this transaction is part of the Service Execution Interaction. In this case, Agreed Service Description will include a description of this part of the Service Execution Interaction – a description of a description.

So while from the IT perspective it may seem that the Service Access and Service Access Use are both homogeneous processes, from a business perspective they involve different concerns and levels of representation.

Service, Service Contract and Levels of Service Presence

Services and Service Contracts are intimately tied together. Service Contracts provide the business explanation of Service Presence. Earlier we noted the OASIS-RAF notion of presence of a service as an aggregation of the presence of the service's actions, where presence for an action means someone can initiate it – and presence is determined by interaction through a communication protocol. This can be seen as a technical characterisation of presence. Service Contracts provide a business level characterisation.

There is a Service Presence when the provider of a service has started the Service Agreement Process. This can, and typically will, be before the consumer is involved. There is a range of ways for the provider to do this, offering different levels of presence. Here are couple of common business patterns for agreement processes that illustrate this. One party can, prior to the other party being involved:

- Make an 'offer'
- Make an 'invitation to treat' (or 'inviting an offer')

The difference between these is that if the offer is made, once the other party accepts it there is a contract – whereas if there has only been an invitation to treat, the other party has to make an 'offer' which the first party can accept or reject. (Where these differ from the situation in the OMG definition is that this described drawing up a contract without any parties being involves – and this has no contractual force.)

Both the service provider and the (potential) service consumer can start the process by making an offer or invitation to treat. However, there is (as noted earlier) an asymmetry here. If the offer or invitation is to *provide* a service, then making the offer or invitation signals the presence of the service (and withdrawing signals the end of the presence). However, if the offer or invitation is to *consume* a service, the presence only commences when a provider responds – as until then there is no service.

This business level context to Service Presence provides contractual nuances to the notion of 'can' (in "someone *can* initiate it"), for example, differentiating between 'can accept an offer' and 'can respond to an invitation to offer'.

This suggests the following conceptual model.

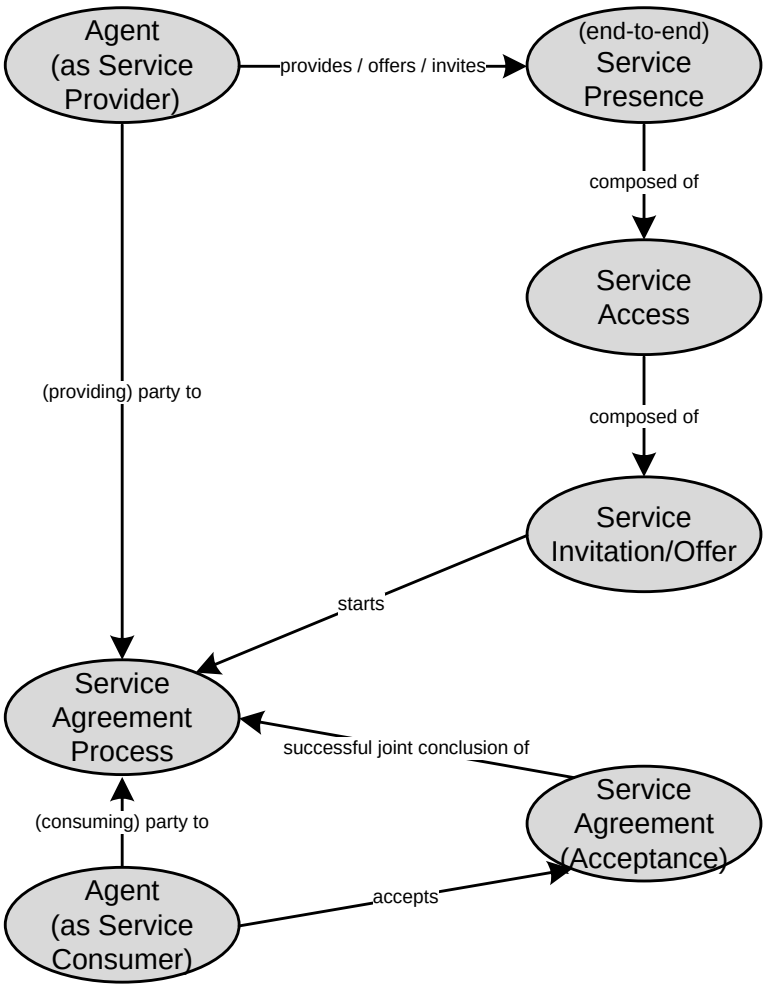


Figure 12 Service Invitation - Offer - Acceptance

The legal terminology is intended to be suggestive rather than prescriptive here. This is because the offer-acceptance formula (and its derivative invitation formula) can be regarded as a simplifying legal fiction. It was developed in the nineteenth century by legal academics and may not always reflect the exact needs of the service. However, it provides a good example of how things can work.

The OASIS-RAF comment that “Presence is determined by interaction through a communication protocol” reflects the technical requirements. Without this interaction, presence – technically – cannot be determined.

Service Interaction or Interfaces

From an IT perspective, the most basic requirement for a service is an interface across which the consumer and provider can interact. For anything but the simplest service, there will need to be some interaction (a thick interaction rather than thin interface).

The standards descriptions of service interactions are below. It is worth recalling that for OASIS RM and SoaML this interaction is the structure of the service, which is the mechanism or resource that provides access to the capability (in our terms, a Service Use Access).

OASIS RM	SoaML	OASIS RAF
3.2.2 Interacting with services	**7.1.4 Service Interfaces**	**4.3 Interacting with Services Model**
Interacting with a service involves performing actions against the service. In many cases, this is accomplished by sending and receiving messages, but there are other modes possible that do not involve explicit message transmission. For example, a service interaction may be effected by modifying the state of a shared resource. However, for simplicity, we often refer to message exchange as the primary mode of interaction with a service.	*Like a UML interface, a ServiceInterface can be the type of a service point. The service interface has the additional feature that it can specify a bi-directional service – where both the provider and consumer have responsibilities to send and receive messages and events.*	*Interaction is the activity involved in using a service to access capability in order to achieve a particular desired real world effect, where real world effect is the actual result of using a service. An interaction can be characterized by a sequence of actions. Consequently, interacting with a service, i.e. performing actions against the service—usually mediated by a series of message exchanges—involves actions performed by the service. Different modes of interaction are possible such as modifying the shared state of a resource. Note that a participant (or agent acting on behalf of the participant) can be the sender of a message, the receiver of a message, or both.*

From a business perspective, what does this interaction consist of? The provider and consumer participants have to interact to complete the Service Agreement Process[11] – so part of the interaction is the Service Agreement Process. They typically also have to interact for the consumer to access (use) the capability.

Service Interfaces and Messages

In a pure IT service, the interaction is likely to be an exchange of information which can be regarded as an exchange of messages. This is the line pursued by the standards.

OASIS RAF
4.1.1.3.1 Service Interface
As noted in the Reference Model, the service interface is the means for interacting with a service. For this Reference Architecture and as shown in Section 4.3 the service interface will support an exchange of messages, where *the message conforms to a referenceable message exchange pattern (MEP),* *the message payload conforms to the structure and semantics of the indicated information model,* *the messages are used to denote events or actions against the service, where the actions are specified in the action model and any required sequencing of actions is specified in the process model.*

However, non-IT business services involve the interaction of things other than information. Using a haircutting service involves the interaction of the barber's scissors and the consumer's hair. In the OASIS electric utility example, using the service involves electricity travelling along wiring. Hence, at the general level service interactions involve interactions that can involve all sorts of entities including information. Restricting the interaction to just information is likely to exclude a number of business services.

From an AF perspective, this means that the general description of the structure of the Service Use should not assume that it only involves the exchange of messages.

[11] Though there is a notion of a unilateral contract, where there is no requirement for interaction to create a contract.

Service Categorisation - Taxonimisation - Classification

The service analysis above has some implications for how services should be classified, which we describe below.

A common approach

A common approach to creating a taxonomy of services is to base the classification upon the capabilities provided by the service. Here is an example from NATO OA v3.1.

Attributes	
Name	NSOV-1 Communication services
Hierarchy	
Communication services	Customer Facing services
Objects	
Service type	Audio Conference
Service type	IP data service
Service type	L2 point-to-point service
Service type	Service Desk
Service type	TeleFax or facsimile
Service type	Telephony Service
Service type	Video Tele Conference
Service type	WAN Access for Coloured Cloud

Table 1 Example of a typical service taxonomy – NATO OA v3.1

A 'Telephony Service' is a 'Telephony Capability' that is delivered as a service. For each of the services there is necessarily a corresponding capability (in the wider OASIS sense) and, for this taxonomy, the type of the capability defines the type of service.

Delivering a capability

However, a service is only one way a capability can be delivered – and it can make sense to deliver a capability in a number of ways. Consider again the example of a taxi service. The underlying capability is a 'taxi' ride between two points. In the case of a family taxi firm, the capability could be used both to provide a taxi ride service to customers and (out of hours) to directly provide taxi rides to members of the family. In this case it makes practical sense to deliver the capability sometimes directly to family members and sometimes as a service to customers.

Recognising that services are not the only way to deliver capability opens the door to innovative solutions. For example, Whipcar (www.whipcar.com) in London and RelayRides (www.relayrides.com) in Boston, Massachusetts allow car owners (people with a car ride capability) to offer their spare capacity as a service. In this case, the car ride capability is delivered both directly to the owners and as a service to the renters. As these cases show, it is possible for a capability to be sometimes delivered as a service and sometimes directly.

A parallel capability taxonomy

If a capability were always delivered as a service, then one could solely work with a service taxonomy. However, in the enterprise, not every capability will always be delivered as a service, so one needs to be able to refer to the capability as well as the service. This implies that there is need for a parallel capability

taxonomy, which will, necessarily, be isomorphic to the service taxonomy typically using the same names with the 'service' suffix replaced by 'capability'.

In this situation, there is a requirement to synchronously create and maintain these two isomorphic structures. However, these kinds of duplicating structure are often an indication that the underlying framework needs to be revised[12]. And in this case a closer look at the underlying framework shows a suitable revision.

A derived capability based service classification

As we have established, a service is just one way in which a capability can be delivered, and the usual approach to classification of services is based upon the type of the associated capability. In other words, the services are being classified not by their intrinsic properties, but by the properties of something they are related to which has an independent existence. This makes it a kind of Cambridge taxonimisation[13]. The "Cambridgeness" of the taxonimisation is particularly clear if one subscribes to the OASIS view of a service (what we have termed Service Access here) where the related capability is not even part of the service.

What this suggests, from a practical point of view, is that in the core architecture one only needs to provide a taxonomy of capabilities and a clear picture of the relationship between capabilities and services – and then the appropriate Cambridge service taxonomy can be derived as and when required in the architectures. This not only avoids the nugatory work of synchronously creating and maintaining two isomorphic structures but also locates the classification close to what is really being classified - the properties of the capability.

This also provides a useful tool for SOA Governance. The framework quite clearly and unequivocally provides a scope for a SOA – the capabilities. One can map the extent of service-isation of the enterprise by looking at how many nodes on the capability taxonomy have a corresponding service – the services footprint in the capability taxonomy (we look at more sophisticated analyses below). One can then look at the fragmentation of SOA Governance, by reviewing the governance of these services. These analyses would be a tool for revealing (and so managing) the extent of *both* business and IT services and their governance. In particular, they would reveal whether there is a focus on IT services (See Appendix C – Inputs - NEC Services Strategies - MoD – NEC and US DoD – NCW).

A finer grained service classification

One of the goals of NEC/NCW is the provision of network-enabled services. In this context, a useful classification of services would be in terms of how close they are to this goal. A framework for this would provide a classification for this in terms of the steps towards the goal. For illustration, consider a simple set of steps; manual, computerised and networked.

The typical analysis above treated services as a simple whole, whereas the earlier services analysis provided a picture of a service as a composite with a number of components. As one looks at these it becomes clear that components can be at different levels of network-enablement and, indeed, that not all components need to be network-enabled for the service to be appropriately network-enabled and different levels of network-enablement can usefully be provided at the same time. These insights suggest that a finer grained

[12] The standard classical example would be the major epicycle in Ptolemaic astronomy, which was 'revised' by Copernicus.

[13] This is a term introduced by P. T. Geach to refer to a classification that not based upon a 'real' intrinsic property of what is being classified – but upon a 'real' intrinsic property of something it is related to. The standard example is widowhood, where becoming a widow does not involve a 'real' change in the widow, but reflect a 'real' change in her husband – see http://plato.stanford.edu/entries/intrinsic-extrinsic/ for more where details.

framework based upon the classification of the components should be used in drawing up roadmaps towards the NEC/NCW goal.

This can be illustrated using the example of a parking control service. In this case, the Service Enabled Capability is the ability to park in the parking space. A couple of decades ago, Service Access was usually provided by a parking attendant or a mechanical parking meter. The Service Access Use involved the interaction of the person parking with the parking attendant or parking meter. From the service consumer's perspective, his or her involvement in the Service Access is manual. From the service provider's perspective, the involvement in the case of the parking attendant is manual and in the case of the meter is automatic. Parking meter technology has improved in the last few decades. Computerised parking meters have been introduced then linked to a central network and more recently replaced by phone text services, supplemented by a phone help desk. However, through all this improvement, the consumer's involvement has remained manual, and indeed in the latest development the provider offers the consumer a manual channel. This is an example of different components of the service usefully being at different levels of automation and networking – and the same component designed to be at different levels at the same time (a network-enabled phone text –– and manual phone help line).

Developments in restaurant services provide a similar picture. Service Access may be provided by a number of channels including a menu displayed outside the door (manual) to internet advertising (so network-enabled) while Service Access use may only be a manual phone call or include an option of computerised (networked) booking. Current practice would seem to indicate that it makes sense for the provider to offer a range of types of access.

For the purposes of the NEC/NCW a more sophisticated classification of the steps towards being network-enabled is probably needed. But as the preceding examples show, these need to be applied to the components of a Service Presence rather than just to the service (Service Presence) as a whole.

A service transaction or relationship – joint actions, plural subjects

There is an aspiration in Defence to move towards a SOA where most if not all the business functions will be delivered as services. Given that the essence of a service involves the use of capabilities across ownership boundaries[14], this is likely to lead to the creation of new ownership boundaries and also issues about how Services are managed across these boundaries.

One of the benefits of operating within an ownership boundary is that it is more natural to share a common goal. One of the purposes of service agreements is to align goals of the parties on either side of the boundary. However there are limits to what service agreements can achieve and a reliance on the easily specified transaction-oriented dimensions, such as availability and timeliness, can erode the relationship dimensions that include trust, integrity, commitment and working towards a common goal.

In Defence operations it is likely that the relationship dimensions will be key and anything that erodes them detrimental, hence it makes sense to classify services along both transaction and relationship dimensions.

The ontological analysis of services has suggested that there may be a key element in the relationship dimension that has not been adequately considered – joint action and plural subjects. Joint action[15] is

[14] OASIS RM - 2.1 - What is Service Oriented Architecture? "Service Oriented Architecture (SOA) is a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains." 2.2 - How is Service Oriented Architecture different? "First, SOA reflects the reality that ownership boundaries are a motivating consideration in the architecture and design of systems. This recognition is evident in the core concepts of visibility, interaction and effect."

[15] OASIS RAF - Joint Action "A joint action is a coordinated set of actions involving the efforts of two or more actors to achieve an effect. ... By definition, joint actions are actions that cannot be performed by single participants. Sometimes this is because no

mentioned in OASIS RAF, where it is used to explain the joint communication required to reach agreement (the Service Access Use). However, there seems to be opportunity to extend this to the Individual Service Use – i.e. to include the Service Enabled Activity.

One way of explaining and understanding joint action is in terms of plural subjects[16], where an agreement constructs a plural subject with responsibilities. There is a simple test for plural subjects. These two examples will illustrate the issues. In a simple contract, both parties accept an obligation to do something. In the sale of a car, John agrees to sell his car to Jill and Jill agrees to pay him a sum of money for it. If either party does not do what they promised, then the other party is under no obligation to do what they promised. If John keeps his car, Jill is under no obligation to pay him the sum of money.

However there are common situations where agreements do not work in this way, where parties accept obligation in the context of a joint goal. If John and Jill want to clean the house before a friend's visit (their joint goal), and they agree that John will do the washing up and Jill the vacuuming. If John then hurts himself and so cannot finish the washing up, then Jill's reaction is unlikely to be stopping the vacuuming. Indeed she is likely, given their joint goal, to do both the vacuuming and the washing up.

From an ontological perspective, as OASIS RAF suggests, the Service Access Use can be regarded as a plural subject with their joint goal being to reach an agreement. What distinguishes the car sale and house cleaning examples is that only in the second case is the Individual Service Use a plural subject – with a joint goal.

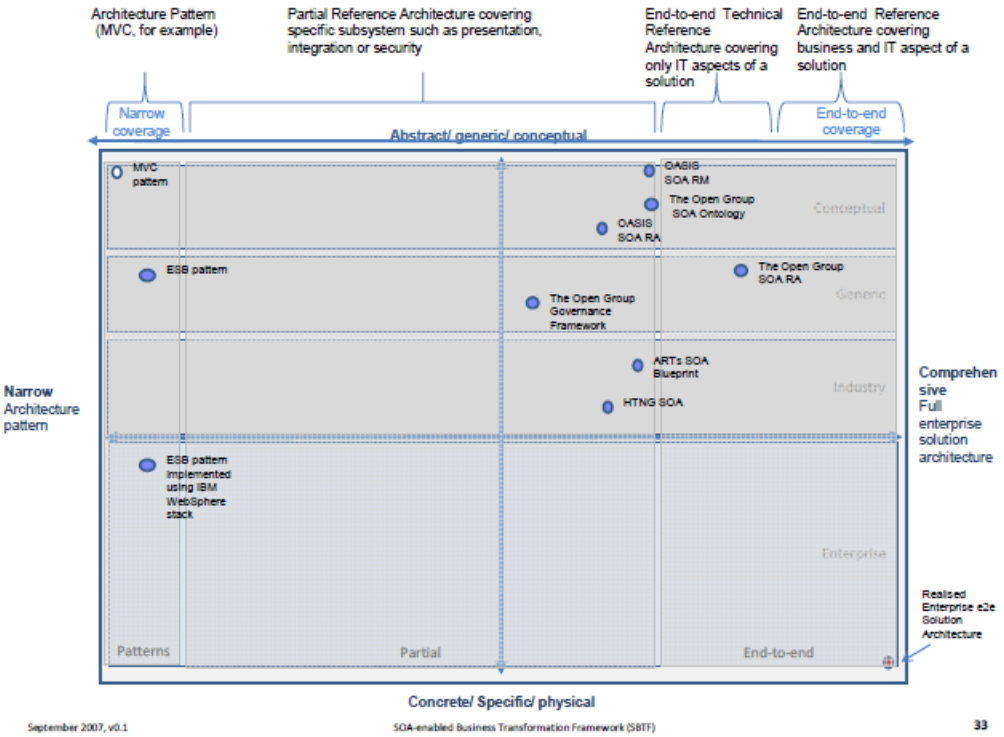
From a classification perspective, it would make sense to include classifications that demarcate where the relationship dimensions are sufficiently important that a joint goal needs to be set and a plural subject established. However, mere classification does not ensure that these conditions are met, and it is important to note that the mechanisms for establishing a plural subject need more research.

single participant has the ability to perform the action on his own; or, in the case of the speaker and listener, the 'joint-ness' of joint actions is inherent."

[16] For more details see C. Partridge, What is a customer? The beginnings of a reference ontology for customer - and M. Gilbert, On social facts. Princeton University Press, 1992.

Appendix A – End-to-end coverage

This analysis of the state of the art for SOA Reference Architectures was originally presented at 22nd Enterprise Architecture Practitioners Conference and has been reported widely. As is immediately visible there are no products in the 'End-to-end Reference Architecture covering business and IT aspect of a solution' space. This report is aimed at the Conceptual level of this space.



Enterprise Reference Architecture, A. Fattah; paper presented at 22nd Enterprise Architecture Practitioners Conference, London, UK, April 2009: www.opengroup.org/london2009-apc/fattah.htm

Also Via Nova Architectura Journal, May 2009: www.via-nova-architectura.org/magazine/magazine/enterprise-reference-architecture.html

Also Navigating the SOA Open Standards Landscape Around Architecture - http://www.oasis-open.org/committees/download.php/32911/wp_soa_harmonize_d1.pdf

Also in - OASIS SOA RAF - Reference Architecture Foundation for Service Oriented Architecture - Version 1.0 - Committee Draft - 14 October 2009.

Appendix B – Gartner's SOA without Governance

Gartner regard governance as an essential component of a successful SOA Architecture. They see SOA without governance as Degenerating SOA.

They have developed a classification of common ways in which SOA with Governance has developed.

SOA Without Governance (aka Degenerating SOA)		
"Wild West" SOA • The most common case of a degenerated SOA. • Services proliferate wildly because no formal service definition process is in place. • Frequently fueled by widespread enthusiasm about the ease-of-use of Web services. • No central registry; nobody knows how many services are in place, where they are or what they do. • Extremely difficult situation to fix and gain control of.	**Shelfware SOA** • A working SOA is implemented, but few applications actually use the public services. • Most applications remain as they are. • There's little buy-in from several business units, no agreed-on application architecture companywide and reuse is an unkept promise. • The intentions are good, but SOA is a waste of resources and won't deliver benefits.	**Duplicated SOA** • Slightly more disciplined and more devious version of a WildWest SOA. • Simply too large; may contain more than 1,000 services. • Although "things work well," many services have significant unplanned duplication • Rewarding mechanisms for creating reusable services and reusing established services are vague. • Little reuse and maintenance costs multiply. • Companies are often reasonably happy with this SOA, even though their savings would multiply if they reduced the level of duplication.

Appendix C – Inputs

IDEAS Nations Meta-Models

- MoDAF – M3 v1.3 (Assume that NAF is broadly similar to MODAF)
- DODAF – DM2 v2.1

Other Reference Models

- ZF - Zachman Framework
- OASIS SOA RM - Reference Model for Service Oriented Architecture 1.0 OASIS Standard, 12 October 2006 - http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf (and associated: Reference Model for Service Oriented Architecture 1.0 - Committee Specification 1, 2 August 2006 - http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf)
- OMG SoaML - Service Oriented Architecture Modeling Language (SoaML) - April 2009 - http://www.omg.org/spec/SoaML/1.0/Beta1/PDF
- OASIS SOA RAF - Reference Architecture Foundation for Service Oriented Architecture - Version 1.0 - Committee Draft - 14 October 2009
- The Open Group – SOAO - Service-Oriented Architecture Ontology - http://www.opengroup.org/projects/soa-ontology/uploads/40/16940/soa-ontology-200-draft.pdf
- The Open Group - SOA Source Book - http://192.153.166.92/projects/soa-book/

Reference models briefly reviewed

- W3C –WSMO - Web Service Modeling Ontology (WSMO) - W3C Member Submission 3 June 2005 - http://www.w3.org/Submission/WSMO/
- OWL for Services (OWL-S) - 2006-03: OWL-S 1.2 Release - http://www.ai.sri.com/daml/services/owl-s/
- W3C – WSDL - Web Services Description Language (WSDL) 1.1 - W3C Note 15 March 2001 - http://www.w3.org/TR/wsdl .

Existing taxonomies

- NATO OA 3.1 – Overarching Architecture
- Open Group – PART III Foundation Architecture: Technical Reference Model (TRM) - Platform Service Taxonomy Index - http://www.opengroup.org/public/arch/contents3_trm_tx.htm
- Microsoft Architect Journal - Ontology and Taxonomy of Services in a Service-Oriented Architecture

NEC Services Strategies

The US and UK documents describing Services strategies focus on the IT rather than the business aspects.

MoD - NEC

- NEC – JSP 777 - http://www.mod.uk/NR/rdonlyres/E1403E7F-96FA-4550-AE14-4C7FF610FE3E/0/nec_jsp777.pdf
- Understanding the NEC - http://www.mod.uk/NR/rdonlyres/F40663B6-F2D2-4058-A1EB-B843559BCCB5/0/1926_NEC.pdf
- MOD Information Strategy 2009 – MODIS - http://www.mod.uk/NR/rdonlyres/530403AF-FC89-41E0-852F-4D2934C15001/0/2009_MODIS.pdf

US DoD – NCW

- Department of Defense - Net-Centric Services Strategy - Strategy for a Net-Centric, Service Oriented DoD Enterprise

Appendix D – Relationship between the SOA Models

Extract from the Joint OASIS, OMG and TOG Paper - Navigating the SOA Open Standards Landscape Around Architecture (November 2009)

Influence of Technical Products

Figure 2 shows the influences of the various SOA open standard technical products (i.e., specifications, standards, etc.) on each other. Since the OASIS Reference Architecture for SOA Foundation [6], The Open Group SOA Ontology [14], and OMG SOA Modeling Language (OMG SoaML) [9] were all based on the OASIS Reference Model for SOA [5] with refinements and extensions, there is some natural affinity between these works. It should be noted that The Open Group SOA Reference Architecture [17] has not been based on or influenced by the OASIS Reference Model for SOA directly. The SOA harmonization discussions have resulted in mutual influences of the content of these reference architecture and governance specifications.

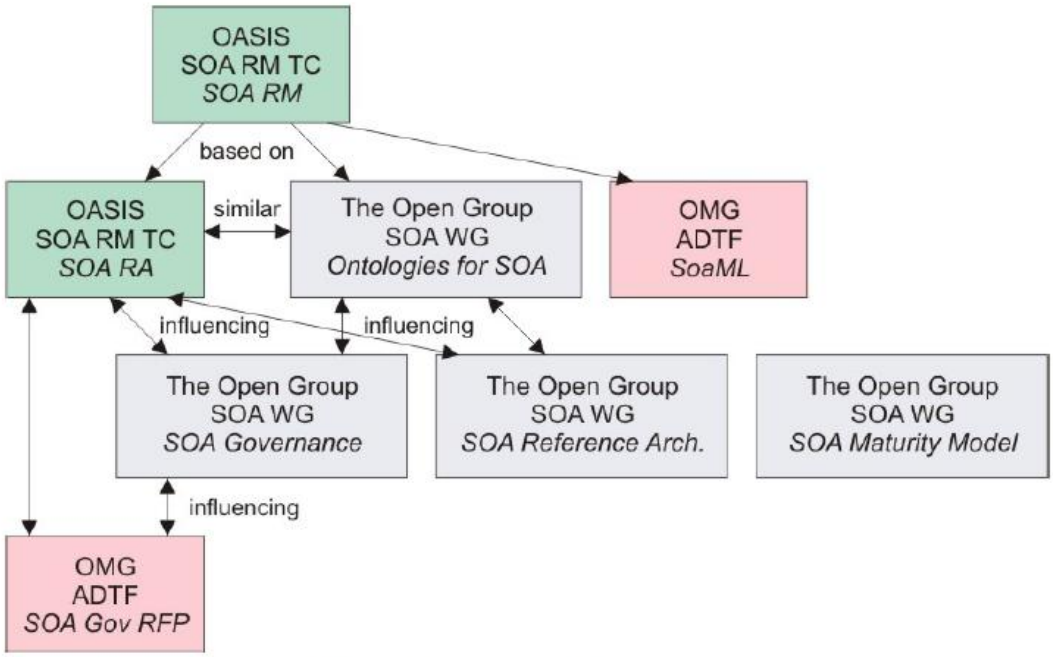


Figure 2: Relationship between Relevant SOA Open Technical Products

Appendix E – OMG SoaML – Annex B

Relationship to OASIS Services Reference Model

This specification attempts to leverage existing work by OASIS and others to ensure compatibility with existing reference models and Web Services platforms. The initial OASIS Reference model for Service Oriented Architecture (version 1.0, Oct. 2006) has been followed up by a broader OASIS Reference Architecture for SOA in April 2008.

Recently also the Open Group has published a draft SOA Ontology. (July 2008).

In the following we compare the definition of main concepts of SoaML with the definition of the similar concepts in the other reference models.

	SoaML	**SOA-RM**	**SOA-RA**	**SOA Ontology**
Org	OMG	OASIS	OASIS	The Open Group
Version	1.8 – Revised Submission	1.0	1.0 – Public Review Draft 1	*Not identified*
Date	Aug 25, 2008	Oct 12,2006	April 23, 2008	Jul 14, 2008
Status	Draft Standard	Completed Standard	Draft Specification	Draft Standard
Concept	**Definition**	**Definition**	**Definition**	**Definition**
Agent	An Agent is a classification of autonomous entities that can adapt to and interact with their environment. It describes a set of agent instances that have features, constraints and semantics in common.	*Not explicitly defined.*	Any entity that is capable of acting on behalf of a person or organisation.	*Not explicitly defined.*
Collaboration	Collaboration from UML is extended to describe ServiceContracts and ServicesArchitecturesServicesArchitectures.	*Interaction: The activity involved in making using of a capability offered, usually across an ownership boundary, in order to achieve a particular desired real-world effect.*	Adopts SOA-RM definition	
CollaborationUse	CollaborationUse shows how a Collaboration (ServicesContracts and ServicesArchitectures) is fulfilled.			

Milestone	A Milestone is a means for depicting progress in behaviours in order to analize liveness. Milestones are particularly useful for 33equiremen that are long lasting or even infinite.	*Not explicitly defined*	*Not explicitly defined*	*Not explicitly defined*
Participant	The type of a provider and/or consumer of services. In the business domain a participant may be a person, organisation or system. In the system domain a participant may be a system, or a component	*Not explicitly defined.*	A stakeholder that has the capability to act in the context of a SOA-based system See also Service Provider and Service Consumer below.	
Real World Effect	Defined as "service operation post condition".	The actual result of using a service, rather than merely the 33equiremen offered by a service provider	*Adopts SOA-RM definition*	Defined as Effect. Conprises the outcome of the service, and is how it delivers value to its consumers.
Request Point (port stereotype)	A request point defines the port through which a Participant makes requests and uses or consumes services.			
Service Point (port stereotype)	The service point stereotype of a port defines the connection point the point of interaction on a Participant where a service is actually provided or consumed			
Service (general)	*Service* is defined as a resource that enables access to one or more capabilities. Here, the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the service description. This access is provided using a prescribed interface and is exercised consistent with all constraints and policies as specified by the service description. A service is provided by a entity – called the *provider* – for use by others.	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	*Adopts SOA-RM definition*	A logical representation of a repeatable business activity that has a specified outcome (e.g., check customer credit; provide weather data, consolidate drilling reports). It is self-contained, may be composed of other services,

	The eventual *consumers* of the service may not be known to the service provider and may demonstrate uses of the service beyond the scope originally conceived by the provider. Identifies or specifies a cohesive set of functions or capabilities that a service provides.	and policies as specified by the service description.		and is a "black box" to its consumers.
Capability	The ability to act and produce an outcome that achieves a result. As such, capability involves the capacity, power, or fitness for some specified action or operation. This implies that the entity must have physical, mental, or legal power to generate an outcome that achieves a real world effect. (synonomous with capability)specifies a			
Capability	A Capability models the capability for providing, or provided by, a service specified by a ServiceContract or ServiceInterface			
Service Contract	A ServiceContract is the formalization of a binding exchange of information, goods, or or obligations between parties defining a service. A ServiceContract is the specification of the agreement between providers and consumers of a service as to what information, products, assets, value and obligations will flow between the providers and consumers of that service – it specifies the service without regard for realization or implementation	A contract, represents an agreement by two or more parties. A service contract is a measurable assertion that governs the requirement and expectations of two or more parties.	*Adopts SOA-RM definition*	*Adopts SOA-RM definition*

Service Interface	Defines the interface to a Service or Request. A ServiceInterface defines the interface and responsibilities of a participant to provide or consume a service. It is used as the type of a Service or Request port. A ServiceInterface is the means for specifying how to interact with a Service	*Service Description* The information needed in order to use, or consider using, a service.	*Adopts SOA-RM definition*	*Description.* An information item that is represented in words, possibly accompanied by supporting material such as graphics. The Description class corresponds to the concept of a description as a particular kind of information item that applies to something in particular – the thing that it describes. It is not just a set of words that could apply to many things.
Service Channel	A communication path between Requests and services. A ServiceChannel provides a communication path between consumer Requests (ports) and provider services (ports).			
Service Oriented Architecture,	An architectural paradigm for defining how people, organizations and systems provide and use services to achieve results.	A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains.	*Adopts SOA-RM definition*	An architectural style that supports service orientation. An architectural style is the combination of distinctive features in which architecture is performed or expressed.

Services Architecture	Services Architecture The high-level view of a Service Oriented Architecture that defines how a set of participants works together for some purpose by providing and using services. A Services Architecture (an SOA) describes how participants work together for a purpose by providing and using services expressed as service			

Appendix F – Some Business Examples

OASIS- RM - 2.1.1 A worked Service Oriented Architecture example

"An electric utility has the capacity to generate and distribute electricity (the underlying capability). The wiring from the electric company's distribution grid (the service) provides the means to supply electricity to support typical usage for a residential consumer's house (service functionality), and a consumer accesses electricity generated (the output of invoking the service) via a wall outlet (service interface). In order to use the electricity, a consumer needs to understand what type of plug to use, what is the voltage of the supply, and possible limits to the load; the utility presumes that the customer will only connect devices that are compatible with the voltage provided and load supported; and the consumer in turn assumes that compatible consumer devices can be connected without damage or harm (service technical assumptions).

A residential or business user will need to open an account with the utility in order to use the supply (service constraint) and the utility will meter usage and expects the consumer to pay for use at the rate prescribed (service policy). When the consumer and utility agree on constraints and polices (service contract), the consumer can receive electricity using the service as long as the electricity distribution grid and house connection remain intact (e.g. a storm knocking down power lines would disrupt distribution) and the consumer can have payment sent (e.g. a check by mail or electronic funds transfer) to the utility (reachability). Another person (for example, a visitor to someone else's house) may use a contracted supply without any relationship with the utility or any requirement to also satisfy the initial service constraint (i.e. reachability only requires intact electricity distribution) but would nonetheless be expected to be compatible with the service interface. In certain situations (for example, excessive demand), a utility may limit supply or institute rolling blackouts (service policy). A consumer might lodge a formal complaint if this occurred frequently (consumer's implied policy). If the utility required every device to be hardwired to its equipment, the underlying capability would still be there but this would be a very different service and have a very different service interface."

SoaML - Example Participant Services Architecture

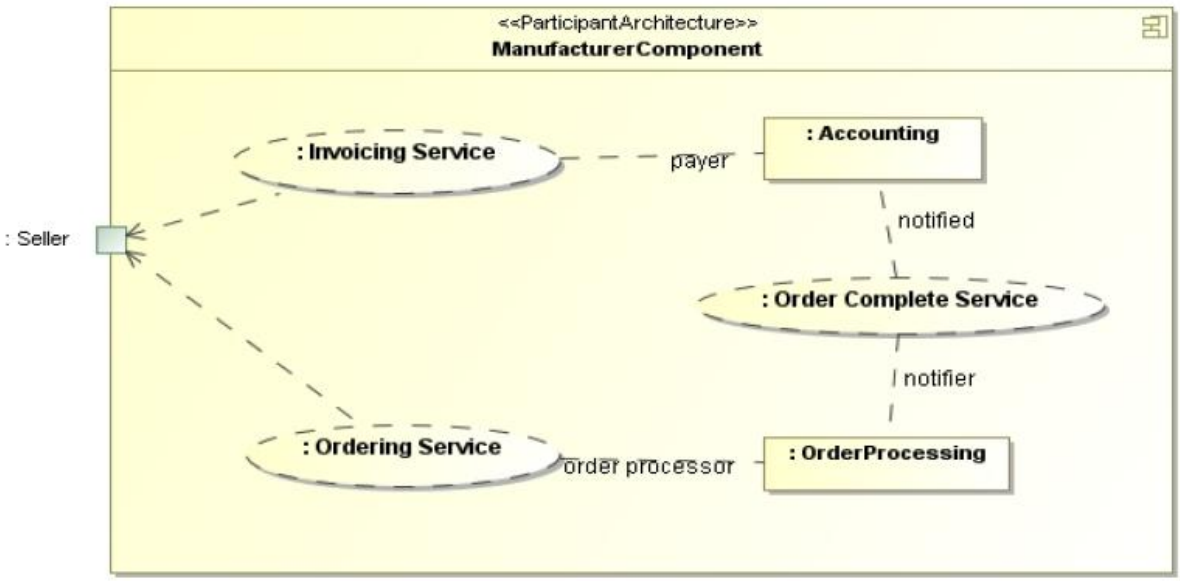


Figure 7.30 - Participant Services Architecture

Figure 7.30 shows a participant's services architecture. The "Manufacturer component" is composed of "Accounting" and "Order Processing." The "seller" service port on the Manufacturer component shows the external responsibility of the manufacturer, which is then delegated to the accounting and order processing parts. In participant architecture there are frequently services connected between internal roles or between internal roles and external ports. The "OrderCompleteService" shows a service that is internal to the Manufacturer while both the "InvoicingService" and "OrderingService" are delegated from the Manufacturer component to the internal participants, accounting, and OrderProcessing, respectively.